K-Sketch: A Kinetic Sketch Pad for Novice Animators

Richard C. Davis

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2008-171
http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-171.html

December 18, 2008

K-Sketch: A Kinetic Sketch Pad for Novice Animators

by

Richard Christopher Davis

B.S. (Massachusetts Institute of Technology) 1995

M.Eng. (Massachusetts Institute of Technology) 1995

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor James A. Landay, Co-chair
Professor John Canny, Co-chair
Professor Carlo H. Séquin
Professor Stephen Palmer

Fall 2008

The dissertation of Richard Christopher Davis is approved:

Co-chair Date

Co-chair Date

Date

Date

University of California, Berkeley

Fall 2008

K-Sketch: A Kinetic Sketch Pad for Novice Animators

by

Richard Christopher Davis

ABSTRACT

K-Sketch: A Kinetic Sketch Pad for Novice Animators

by

Richard Christopher Davis

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor James A. Landay, Co-chair

Professor John Canny, Co-chair

Animation is a powerful communication and visualization medium that is accessible to few, because current tools for creating animation are extremely complex. Simple animation tools exist, but they severely restrict the types of motion that can be expressed. To help novices create a wide range of animations quickly, I have developed a general-purpose, informal, 2D animation sketching system called K-Sketch, a *kinetic* sketch pad. My investigation began with field studies that explored the many uses of short, rough animations both by expert animators and would-be animators. The most significant results of these studies were evidence of the need for K-Sketch and a library of 72 usage scenarios for such a tool. These scenarios show how rough animation can be useful both to experts creating prototypes of animations and to novices creating animations for entertainment, visualization, or communication of dynamic concepts in a wide variety of educational or business settings.

While analyzing my library of scenarios, I identified 18 primitive animation operations that cover the most natural ways of expressing the motions and transitions in the animations I collected. To make K-Sketch simultaneously fast, simple, and expressive, I developed a novel interface optimization analysis method. I used this to visualize the tradeoffs of supporting various combinations of animation operations and choose a small but powerful set of capabilities for K-Sketch. This method and the tools I developed can be applied to many other domains.

The final K-Sketch system uses pen input for sketching objects, intuitive demonstration of motion, and a suggestive interface for resolving ambiguous operations. In one laboratory experiment that compared K-Sketch to a more formal novice animation tool (PowerPoint), participants worked three times faster, needed half the learning time, and reported significantly lower cognitive load with K-Sketch. Another laboratory comparison with a less formal novice tool (The TAB Lite) showed that K-Sketch allows novices to express a wide range of animations quickly and intuitively. K-Sketch has been released to the world and is being used by over a thousand people to create rough animations.

Professor James A. Landay
Dissertation Committee Co-chair

Professor John Canny
Dissertation Committee Co-chair

For my twin brother

William Rhett Davis

in gratitude for years of creative energy

Table of Contents

Acknowledgements

I could write an entire book about the drama surrounding this dissertation, but time and space allow only a listing of the major characters. Apart from myself, the chief protagonist and antagonist would be James Landay, often the sole supporter of this work, and sometimes appearing to be my greatest enemy. Such tension is not uncommon among the world's most fruitful partnerships. There is no doubt that I owe him an unfathomable debt of gratitude. My other committee members, John Canny, Carlo H. Séquin, and Stephen Palmer, also provided help at critical moments, and I am grateful to them all.

It is also extremely important for me to recognize others who contributed their time and talent to the K-Sketch project. Chief among these contributors is Brien Colwell, whose talent and energy were indispensible in the year leading up to K-Sketch's release. Cy Khormaee provide a very helpful installation program. Matthew Ryan Davis set up the initial K-Sketch web site and help files. Arpi Shaverdian helped with several early user tests.

Several others provided essential help with planning and analyzing user studies. Statistics goddess Elizabeth Sanders gave me a semester's worth of statistics help at the last minute and asked for nothing in return. Jacob Wobbrock also taught me some helpful analysis techniques. Mary Czerwinski gave helpful feedback on my study designs. Tom Williams and Donald E. Parker pointed me to some important pieces of related work.

I'm also indebted to the people who kept me functioning as a graduate student. Lauren Bailey and LaShana Porlaris did the leg work that allowed me to work on my dissertation while away from Berkeley for five years, while Mary Byrnes and Ruth Gjerde gave me strategies and moral support. Similar help was provided at the University of Washington by Kay Beck-Benton, Shannon Gilmore, Carol Frescoln, and Lindsay Michimoto.

I'd also like to acknowledge the many colleagues who gave me encouragement and support on this long journey. My officemates Jimmy Lin, Jason Hong, Mark Newman, Ken Yasuhara, Kurt Partridge, Imran Rashid, and Kate Deibel calmed me down on many occasions. Several students helped make a friendly work environment for me in other ways, including Kate Everitt, T. Scott Saponas, Jon Froehlich, Ana Ramirez Chang, Michael Toomim, Susumu Harada, Kayur Patel, and Yang Li. Many professors and mentors helped me along the way, as well: Richard Anderson, Steve Tanimoto, Gregory Abowd, Bill Winn, Jen Mankoff, Anind Dey, Jacob Wobbrock, Scott Klemmer, Bill Schilit, Scott Hudson, Mike Clancy, and Michael Ouellette.

I'd now like to thank my family and friends for keeping my spirit alive while finishing this endurance run: my mother Alix D. Hay, my brother William Rhett

Davis, my sister Caroline Hay, my fathers Bill Hay and Frank Davis, Paulo and Leah Pereira, Alex Edezhath, Greg Wolfe, Sal Francomacaro, Brian and LeAnn van Essen, Dave Crean, Mark and Jan Shea, Mike and Alicia McLeod, Angela Kim, Robin Enders, Amanda Henck, Pat Bucy, Tim Kowalewski, Dan and Melissa Odell, Br. Corwin Low, Br. Christopher Fadok, Fr. Garry Cappleman, Fr. Gus Hartman, Fr. Daniel Syverstad, Fr. Bernhard Blankenhorn, and Fr. Ed White.

Finally, I'd like to acknowledge the invaluable support of NSF Grants 0080562, 0205644, and 0742877 that funded this research.

1 Introduction

Animation is ubiquitous in western culture. A large segment of the entertainment industry is oriented toward the production of animated movies and television shows for children and adults. Commercials use animated logos or characters to make products enticing. Businessmen and researchers use animated presentations to communicate subtle points or to engage their audiences. Educators show their students animated videos to illustrate dynamic concepts. With so many animated images surrounding us, it is not surprising that many people develop the desire to express their own ideas through animation, but this poses a challenge. The necessary tools are not easy for a novice to master, and one must be very dedicated to produce even the simplest animation.

Consider the following situations in which novice animators are inspired to create simple animations but the barriers prevent them from doing so:

1. A teacher explores an online community for sharing animations of chemical reactions. He sees many interesting animations and wants to create similar ones that match diagrams in the textbook he is using. He has used PowerPoint [106] to make animations before, but it takes him so long to prepare lessons and grade

homework that he cannot make time to create the animations he envisions. He'd prefer to devote a minute or two of class time to creating each animation, talking through it as he goes. This teacher needs an animation tool that will allow him to work faster than he can work with PowerPoint.

2. A child receives an animated greeting card and wants to create her own for Mother's Day. An early step in learning to communicate in any medium is to experiment in that medium, trying out ideas to see what works. If this child must start, rather, by learning to use a complex tool or by taking a class in animation techniques, her inspiration may die before she has learned how to create what she imagined. This child needs a simple animation tool so that she can grasp the basic concepts quickly and explore the medium without being distracted by the tool's idiosyncrasies.

3. At a conference, a researcher sees a tool for creating simple animations of wave patterns that would help him make quick illustrations of concepts for his colleagues. The tool looks fairly easy to use, but it does not produce all the patterns he needs to create. He considers requesting a new feature from the software vendor or looking for a similar tool with the features he needs, but he gives up hope and decides to work without animation at all. This researcher needs a tool that allows him to express a wider variety of animations.

In my research, I have collected 72 usage scenarios similar to these. My analysis of these scenarios has led me to believe that novices need a tool that is simultaneously *fast*, *simple*, and *expressive*. This tool must allow them to create animations in a short time, with little learning and minimal concentration, and it must allow them to express most animations they can imagine. Unless a tool has all three of these qualities, some novices will give up on creating an animation before they have seen how valuable it can be.

Note that a rough animation would suffice in all of the examples presented here. These users are not professionals, and they do not need their animations to have the same level of polish as professionals do. A key insight of this work is that an **informal** tool would suffice for such users. Informal tools defer the specification of small details until they are necessary. The artifacts created by informal tools often have a rough feel, like sketches on paper, but they can be created more quickly and easily than is possible with conventional tools.

1.1 Contributions

The central thesis of this dissertation is that **an informal, sketch-based animation system designed through analysis of usage scenarios will allow novices to create a wide variety of animations quickly and easily and to use animation in new ways.** The dissertation offers contributions in three areas:

1. **Concepts and techniques** for designing user interfaces in general and for animation in particular, including the following:
    a. An analysis of 72 usage scenarios for a rough 2D animation tool that reveals the categories of tool users, their goals, and the abstract operations they use to create animations.
    b. A novel interface optimization method, appropriate for user interfaces in any domain, that allowed me to design an interface that balances the need to maximize the expressive power while also minimizing user task time and interface complexity.
2. **Artifacts** that provide immediate benefit to novice animators and guide future research and development of animation tools:
    a. A design for a novice animation system called **K-Sketch:** a *kinetic* sketch pad that uses demonstrated motion, a simplified timeline, and a suggestive interface for resolving ambiguous intentions.
    b. An implementation of K-Sketch that is currently available for download by the general public.
3. **Study results** that show the benefits of using an informal animation tool, including the following:
    a. K-Sketch allows tasks to be performed three times faster than with a more formal tool, Microsoft PowerPoint.

b. K-Sketch helps users work quickly by avoiding unnecessary and distracting details.

c. K-Sketch allows users to quickly express a wide variety of animations.

d. K-Sketch enables novices to use animation in new ways, for example, to make quick prototypes of user interface designs and to help children learn science.

The following paragraphs give an overview of these contributions.

1.1.1 Concepts and Techniques

I began my exploration by conducting field studies to determine how an informal animation tool might be used and whether or not it could be made general-purpose. Since many novice animators seek to perform the same tasks performed by experienced animators, I first interviewed experienced animators to see how an informal tool would fit in their work process. There were eight participants in all, and most thought that an informal tool would be a valuable way to make prototypes of finished works, either for themselves or for clients during meetings.

I also conducted eleven interviews with non-animators, people who were interested in animation and could describe specific animations they wanted but did not know where to begin. Many of these participants had heard of Flash and knew that PowerPoint could be used to create animation, but they were intimidated by the complexity of these tools or could not make time to learn how to use them. These non-animators came from a variety of disciplines including teachers, engineers, and businessmen, and their animations covered a range of subjects from science and engineering to dance and jumping rope.

From these two sets of interviews I compiled a library of 72 usage scenarios for an informal animation system, each including a specific animation (see Appendix B for a partial list). My analysis of this library showed that users fell into five categories (amateurs, artists, teachers, students, and professionals), and their goals also fell into five categories (entertain, explain, think, learn, and prototype). I also defined eighteen abstract **animation operations** that users might use to express the action in these animations. The operations include movement with direct manipulation and more complex operations, such as controlling the speed of movement, defining hierarchies and limbs, or generating motions from a set of input parameters.

To reach my goal of creating a fast, simple, and expressive animation tool for novices, I knew that I needed to carefully choose the animation operations my tool would provide. With too few operations, some scenarios would be difficult or impossible to create. With too many, my tool could become as complex as any existing tool. I noted, however, that not all operations were used by every scenario, and most scenarios could be accomplished multiple ways with different sets of operations. To identify the best set of animation operations for my tool, I developed a new analysis method called **interface optimization**.

The first step in interface optimization is for a designer to break up each scenario's animation into **features**: independent parts of the animation that users must represent. In the case of my animation tool, features include objects, their motions, and higher level aspects of the animation (e.g., the existence of multiple scenes). The designer then lists all the **approaches** to representing these features and notes the operations required by each approach. In the second step of interface optimization, a computer program processes this data to produce a table that relates the number of operations a tool could provide to the number of scenarios it would **support** (i.e., allow users to perform efficiently). Using this method, I was able to identify a set of nine animation operations that would allow my tool to support 79 percent of the scenarios in my library.

1.1.2 Artifacts

I used the recommendations of my interface optimization analysis to design K-Sketch, a *kinetic* sketch pad for novice animators (see Figure 1-1). K-Sketch is a pen-based interface that takes advantage of users' intuitive sense of space and time, allowing them to quickly sketch and animate objects.

K-Sketch models animation as a sequence of editing steps over time (see Figure 1-2)By default, any edit operation that the user performs happens instantaneously at the current time index and is visible from that time forward. To

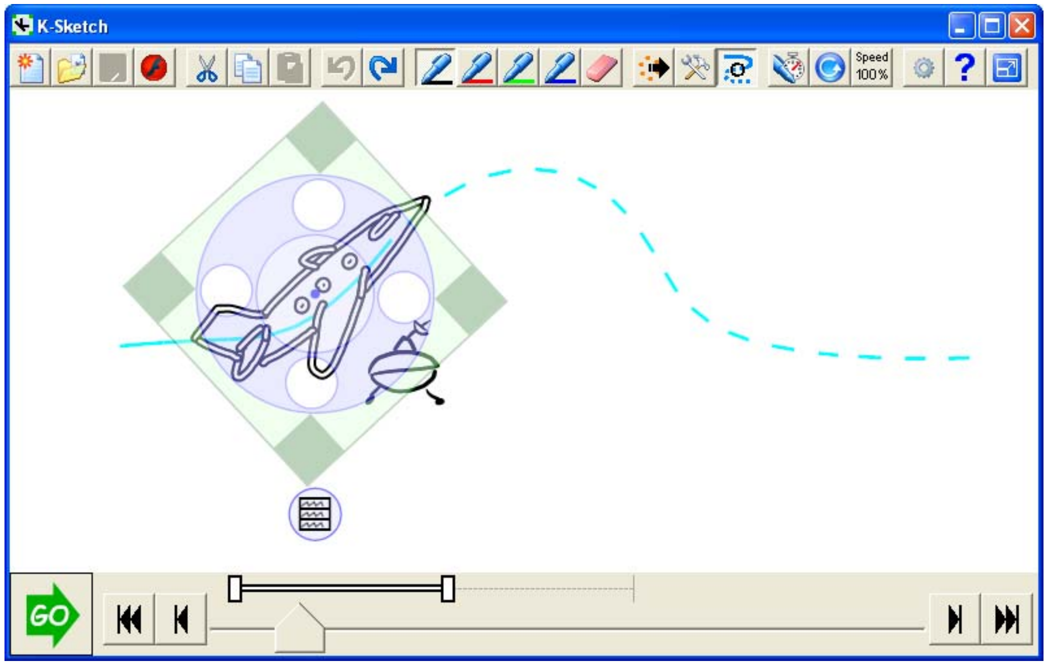


Figure 1-1: The K-Sketch User Interface. Users sketch and move (i.e., edit) objects in the center canvas. The slider bar at the bottom indicates the current moment in time. Users create animations with a series of edit operations that play back instantaneously or in real time just as they were performed. The dashed line shows the motion path of the airplane.

record an animation, users move objects with the same editing interface and demonstrate motions in real time.

In contrast, conventional animation tools have complex timelines that show all the transformations applied to each object over time. In contrast, K-Sketch has a simplified timeline that shows only the time at which important events occur, highlighting events related to the currently selected object. In addition, a motion path appears when users record a motion. This motion path serves as a reminder of how an object moves and helps users manage and coordinate objects' motions. This path is a

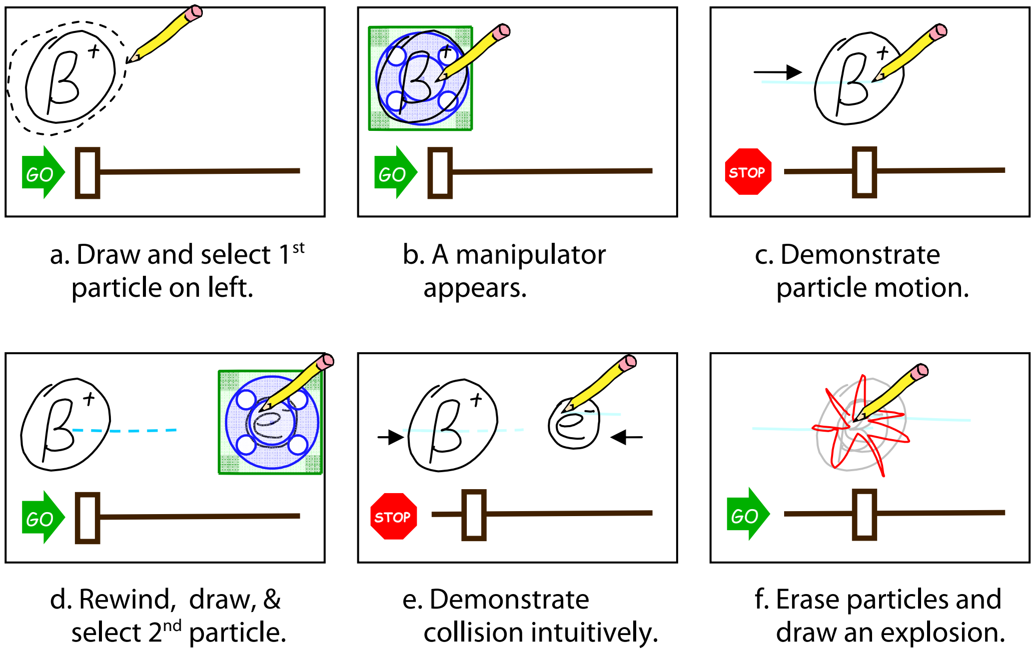


Figure 1-2: Creating a particle collision animation with K-Sketch. The animation is built up with a series of editing steps. Some edits are recorded in real time.

selectable entity that can be moved to change the trajectory of an object or copied and applied to other objects.

Many animations contain motions that are hard to express without breaking them into a set of simpler motions, each with its own reference frame. Instead of providing an interface for defining and navigating through reference frames, K-Sketch uses heuristics to choose a frame for every new motion. When K-Sketch inserts a new motion into the incorrect reference frame, users can fix the operation by choosing the correct motion from a list that animates the motion in all possible frames. By intuitively picking the correct motion from a list, users can completely avoid understanding the complex concept of a reference frame.

K-Sketch has been released to the public and can be downloaded online from www.k-sketch.org. It was downloaded by over 1000 people in the first two months of its release.

1.1.3 Study Results

In addition to the formative user tests I conducted during iterative design cycles, I conducted a summative evaluation of K-Sketch that included two sets of studies. In laboratory experiments I tested my hypothesis that K-Sketch would allow novices to create a wide variety of animations quickly and easily. In real-world evaluations, I tested my more general hypothesis that K-Sketch would allow novices to use animation in new ways.

The first laboratory experiment specifically evaluated how quickly and easily novices could create animations with K-Sketch. I asked sixteen people with little or no animation experience to create two animations from my library using K-Sketch and a formal tool for novice animators: Microsoft PowerPoint [106]. Participants were able to complete tasks three times faster on average with K-Sketch, and they were able to complete their K-Sketch tutorials twice as fast as their PowerPoint tutorials. K-Sketch also felt faster and easier on subjective scales, and participants exhibited half the cognitive load when using K-Sketch according to the NASA-TLX cognitive load self-assessment. All these results were statistically significant.

Surprisingly, participants were no less comfortable *showing* their informal animations to others than their formal animations, even to large numbers of people in formal gatherings. This may have been due to participants' inability to produce pleasing formal animations while under time pressure. Conversely, participants were significantly more comfortable *creating* animations in front of others using K-Sketch. This may indicate that K-Sketch could one day make animation a viable medium for spontaneous collaboration.

The second laboratory experiment evaluated the breadth of animations that users could express with K-Sketch, and relates more directly to my interface optimization analysis. I asked seven novice animators to create nine animations chosen randomly from my library. To provide a basis for comparison, I had participants complete tasks both with K-Sketch and The TAB Lite [48], another novice animation tool that could be easily configured to use a pen (thus factoring out the effects of formality). In six of the nine tasks, participants were able to work two to three times faster with K-Sketch. Also in contrast with The TAB Lite, participants seldom needed to sketch notes or make mathematical calculations while using K-Sketch, making it easier to work with. Thus, K-Sketch is able to preserve speed and simplicity while still allowing novices to express a broad range of animations.

My real-world evaluations explored how K-Sketch is used by real users doing real work. I analyzed logs of 262 K-Sketch sessions from 54 users as well as 228 animation files from 163 users. The results show that the benefits of K-Sketch are not confined to the laboratory. Some of this data came from two groups of novices attempting to use animation in new ways. One was a group of researchers designing a user interface for people with motor impairments. This team was able to quickly evaluate 25 alternative forms of visual feedback by sketching and discussing animations. The second was a group of 106 high school students who created animations with K-Sketch as part of an experimental learning program to improve their understanding of scientific concepts. Students' animations revealed their misconceptions in an immediate way that made it easy for their teacher to address them. K-Sketch enabled both of these groups to make a quantum leap in productivity, generating higher quality results faster than ever before.

1.2 Dissertation Outline

In the pages that follow, I will give the context for this project; present my field work, analysis, and the resulting design for K-Sketch; and describe how I evaluated K-Sketch. Chapter 2 gives an overview of current animation tools and related work in informal interfaces, tools that applies sketching and demonstration to animation, and animation as a medium for learning and communication. In Chapter 3, I present my interviews of animators and non-animators as well as the user categories, goal categories, and animation operations I gathered from them. I describe my interface optimization method and the results of my analysis in Chapter 4. Chapter 5 describes the K-Sketch system design and implementation. In Chapter 6, I describe the formative evaluations I conducted during this project. This is followed by two chapters on the summative evaluations I conducted: laboratory evaluations in Chapter 7 and real-word evaluations in Chapter 8. Finally, Chapter 9 explores future work and Chapter 10 presents concluding remarks.

Appendix A lists all the data I collected in my field studies, except for the library of animation scenarios I collected, part of which appears in Appendix B. Appendix C contains the unabridged data produced by my interface optimization analysis. The remaining appendices list documents and data from my laboratory studies. Appendix D contains documents and data from my formative laboratory evaluation. Appendix E contains documents from my first summative laboratory experiment, and Appendix F lists the data I collected from that experiment. Similarly, Appendix G contains documents from my second summative laboratory experiment, and Appendix H lists the data I collected from that experiment.

2 Related Work

Before describing this research in depth, I will show how K-Sketch grows out of related work in psychology, education, and computer science. I begin in Section 2.1 with a discussion of novice animation culture and argue that there is a need for a new novice animation tool. I point to signs of novice animation in entertainment, and then explore the need for novice animation in education. I also describe current animation tools for experts, which many novices feel forced to use, and present some animation tools designed specifically for novices.

In Section 2.2, I argue that novice animators will be best served with a pen-based tool and describe previous work that teaches us how and why to build such tools. This section examines how sketching is used by expert designers, as well as looking at what creativity support research says about sketching and about supporting novices. I also show how K-Sketch fits in the tradition of pen-based interfaces. I then look at previous attempts to use sketching and gesturing in animation. I close with a brief look at important tools and techniques for pen-based interfaces.

An epilogue to this related work is Section 2.3, which discusses motor and perceptual issues in animation sketching. Since demonstration is a technique that is rarely used, it is natural to ask whether or not people will be able to understand demonstrated animations. This includes a discussion of how people perceive information from animations, and a demonstration of how motor control research predicts that K-Sketch animation can be interpreted correctly.

2.1 Animation Culture

I have already discussed the ubiquity of animation in our culture. Though the tools for animation are cumbersome, non-experts are beginning to participate in this culture. We see this primarily in two areas, entertainment and education, which I address separately in the following sections. I then review the tools that are currently available to both experts and novices.

2.1.1 Amateur Animation for Entertainment

There is evidence that increasing numbers of amateurs want to entertain themselves and their friends by creating and sharing short animations. This animation subculture is seldom studied by scholars, but it is hard to deny its existence given the number of amateur animations that appear on YouTube [161] and similar web sites for sharing video content. Sites such as aniBoom [12] and Lilipip [100] seek specifically to profit off the work of animators by aggregating short animations. These sites encourage amateurs by offering them tools, community support, and an easy way to market their work. The skills to make animation, however, are hard to come by, and judging by the growing number of animation schools [13] animation education is starting to become a big business.

The popularity of amateur animation coupled with the obstacles to creating it make K-Sketch particularly attractive to novices. Few amateurs can invest the time and money necessary to develop skills with expert tools and may never discover the joy of creating in this medium. K-Sketch provides novice animators an easy way to get started with animation. Some may find that rough animation provides them all the expressive power that they need, and indeed many popular animations are left in rough form. Others novices may need to create rough animations for a while before they will see their own potential as animators and find the energy to tackle expert tools.

2.1.2 Novice Animation for Education

Animation in education is a well studied field. Educators have been experimenting with animated visuals since the technology first became available. In computer science, many educators have used animation to teach computer algorithms [86]. An early example is the legendary *Sorting out Sorting* produced by Ronald Baecker [18]. Soon, systems emerged for generating animations of algorithms and various system processes, such as TANGO [140] and BALSA [27]. Physics educators were also early adopters of animation. *The Mechanical Universe* was a popular educational television show produced by David Goodstein at The California Institute of

Technology [60]. Demand for animation in educational settings is so great that teachers are working hard to convince each other to produce and share animations. The Physlets community, for example, encourages teachers to create Java applets that illustrate physics principles with animation and simulation [32, 39]. Chemistry teachers also share animations of chemical processes and encourage and teach each other how to produce them [28]. The effort required to produce such animations is still significant, but K-Sketch offers teachers an easier way to create such animations.

For decades, education researchers have been attempting to determine the educational benefits of animated visual aids. The result of this effort has been largely inconclusive, with some studies showing benefits [72, 123, 129, 160], others showing no benefits or negative results [66, 150], and some producing mixed results in the same study [120]. Researchers have pressed forward, however, determined to understand the conditions under which animation produces positive learning effects. Rieber incorporated simple animations of moving bodies into computer-based physics instruction [129]. He concluded that animation was helpful in teaching concepts that involved motion and trajectory and laid out details concerning how animations should be interleaved with practice questions. Hundhausen and colleagues reviewed animation research and determined that differing forms of practice could account for conflicting results [72]. Tversky and colleagues argued effectively that learners need control of how an animation plays in order not to miss important events [150].

Current research carefully examines the mental processes that take place during learning. Some of this work focuses on the process by which learners construct mental models [90, 103, 123]. Kriz and Hegarty, for example, observed that the domain knowledge of the learner influences how they direct their attention at an animated display, which can affect learning benefits [90]. Lowe pointed out that animation runs the risk of overwhelming learners with so much information that they cannot direct their attention effectively or underwhelming them by failing to engage their mental processing [103]. Other work focuses on how conversation in collaborative learning settings reflects mental processes. Bétrancourt, Dillenbourg, Rebetez, and Sangin have found that animation affects what learners talk about in such settings [132]. They have also found that dividing animated material into segments improves conversation between learners but that showing static snapshots of an animation is detrimental to conversation [133].

My synthesis of research into the value of animated visuals is that the use of animation can have significant educational benefits but that educators are not yet clear on the conditions under which these benefits appear. In such a situation, educators need to constantly refine the animations they produce as they seek the optimal conditions for learning. Since K-Sketch allows novices to create animation quickly, it is perfect for this community.

As an alternative to animated visuals, some educators are beginning to experiment with student produced animations as a learning exercise. Sometimes, these animations mimic setting up an experiment, as in the 4M:CHEM system, which allows students to define the parameters of a chemistry experiment and watch the resulting reaction at four different levels of detail [131]. In other cases, educators ask students to produce animations of processes [144] using simple frame-by-frame animation tools (e.g., Sketchy [59]). K-Sketch is perfect for this situation, because student animations can often be left in a rough state. Also, most educators would prefer to minimize the time their students spend learning to use animation tools and maximize the time they spend thinking about concepts. Education research even hints that the demonstration technique K-Sketch uses could have educational benefits. Hegarty and colleagues found that the premotor mental representations and spatial working memory resources involved in hand gestures can help students create mental animations when solving mechanical reasoning problems [67].

In summary, K-Sketch has clear benefits for educators using animation. The simplicity of the tool and the speed with which animations can be produced allow teachers and their students to take advantage of this medium while focusing more of their attention on the subject matter and less on the tool's interface. The roughness of the resulting animations is often acceptable in educational environments. Student produced animation is a particularly promising new teaching technique, and I report the results of one teacher using K-Sketch for this purpose in Chapter 8.

2.1.3 Animation Tools for Experts

The growing demand for animation has brought about a growing number of tools for animation. The most well known tools are designed for well-trained experts. Some are general-purpose tools for animators while others are special-purpose tools for other experts. To give a sense of how K-Sketch relates to the commercial world, I will list a sample of two-dimensional (**2D**) animation tools in each of these categories. I will also use this opportunity to define key terms (highlighted in **bold**) related to 2D animation. All key terms in this section have a more complete definition in Goulekas's glossary [61].

Most tools for expert animators evolved to serve a particular market. AfterEffects [5] was developed for professionals producing television shows and short films with digital video. The tool allows animators to import photographs or scanned drawings and animate them by defining **key frames**. A key frame defines the properties of an object (e.g., position, orientation, size, and transparency) at important points in time so that the animation tool may generate the **in-between frames** that complete the animation. The generation of in-between frames is called **in betweening**. Animators can also draw a **motion path** along which an objects moves between key frames.

These techniques speed up the process of animation somewhat, but it is still time-consuming.

Flash [6] was developed for web site designers. It encourages designers to create content within the tool so that it may be kept in a vector format for efficient downloading. Flash's animation interface also uses key frames, and its vector format allows it to **morph** one complex drawing into another. This is a compelling effect, but it is hard to produce. An automatic morph may produce distractingly ugly in-between states, and the animator may need to iteratively add numerous constraints until the morph is pleasing.

Another class of tools not only handles in betweening and morphing, but also caters to animators who prefer the traditional **cel animation** process pioneered by Walt Disney. A good example of this class is Toon Boom [149]. The name cel animation comes from the sheets of cellulose acetate on which animators would draw parts of a character or scene. The illusion of movement was created both by moving the cels through space and by switching to new cels. For example, to make a character walk across the screen, an animator can flip through a cycle of cels that show the character in various walking positions while moving the cels across the screen. This process can produce beautiful animation, but it is extremely laborious, even with tools like Toon Boom.

The tools just presented are designed primarily for highly trained professional artists, but other professionals have occasional needs for animation, and tools have evolved for each group. Scientists often use tools like Mathematica [159] to create animations that help them visualize mathematical functions. To create an animation that accurately reflects the motion of a physical system in the real world, engineers can build the system in Working Model [46] and run a **physical simulation** to generate an animation. In computer science, some have used an animation description language called Slithy [164] to program animated presentations. All of these tools can produce animations of a particular type faster than they could be produced with a professional animator's tool, but they still require considerable training and domain knowledge.

2.1.4 Animation Tools for Novices

To help children and adult novices animators get started without the need for extensive training, some tools provide a shorter list of features or constrain animation in ways that make it more accessible. Sketchy [59] is an extremely simple tool that allows children to draw animations frame by frame on a mobile pen-based device. Getting started with Sketchy is easy, but producing animation frame-by-frame is painfully slow. Animation-ish [148] also encourages children to sketch with a pen, but it gradually introduces them to more powerful concepts like key frames and motion paths. The TAB Lite [48] targets adult novices and speeds up the animation process through morphing. These tools are also much easier to learn than expert tools, but because they use the same techniques as current tools for expert animators, they will never be significantly faster.

Some novice animation tools provide both simplicity and speed by using a small number of novel interaction techniques, but they have significantly less expressive power. The Pivot Stickfigure Animator [25] is an example. It allows novices to make movies of stick figures blazingly fast with little learning, but it can do little else. PowerPoint [106] suffers from the same problem to a lesser degree. PowerPoint is designed for office professionals making presentations, but it assumes that its customers are novices when it comes to animation. Through PowerPoint's **custom animation** interface, users can add **effects** that animate objects in pre-defined ways to spice up a presentation. This can produce eye-catching presentations quickly and easily, but any motion outside the range of pre-defined effects is difficult to produce.

One other sub-category of children's tools bears mentioning. Research in programming by demonstration and computer science education has produced many commercial and freeware tools that simplify programming for children—for example AgentSheets [7, 124, 125], Stagecast Creator [139] (formerly KidSim [37]), Scratch [107], and Alice [31, 34]. Many children use these systems to produce animations that tell stories [126]. Alice has even added special extensions for storytelling [85]. Such tools are easier to use than expert programming languages, but that is a low bar for simplicity. Their speed and expressivity for telling stories compares to that of expert animators' tools. As the creators of these tools would point out, their primary goal is encouraging kids to program computers.

With so many tools for creating animation, we might be tempted to think that there is nothing lacking. The fact remains, however, that many people dream of creating animations and have no idea where to begin. Most of these people would be comfortable to remain novices, and they want an easy way to get started that does not require them to invest the time, money, or concentration needed to become proficient with expert tools. Many of these people want to work quickly, because they are just kicking around ideas, or because creating an animation is part of some larger task and not an end in itself. Finally, their tool must allow them to express a variety of animations, because they do not want to go searching for an appropriate tool every time they have a new idea, nor do they want to maintain a large collection of tools.

Figure 2-1 organizes the animation tools presented in this chapter by these three important dimensions: *speed, simplicity,* and *expressivity.* Each tool seeks to optimize one or two dimensions at most, but the users described here need a tool that optimizes all three. Developing such a tool requires me to draw together the best ideas from several fields. The remainder of this chapter will present the key ideas that led to K-Sketch.

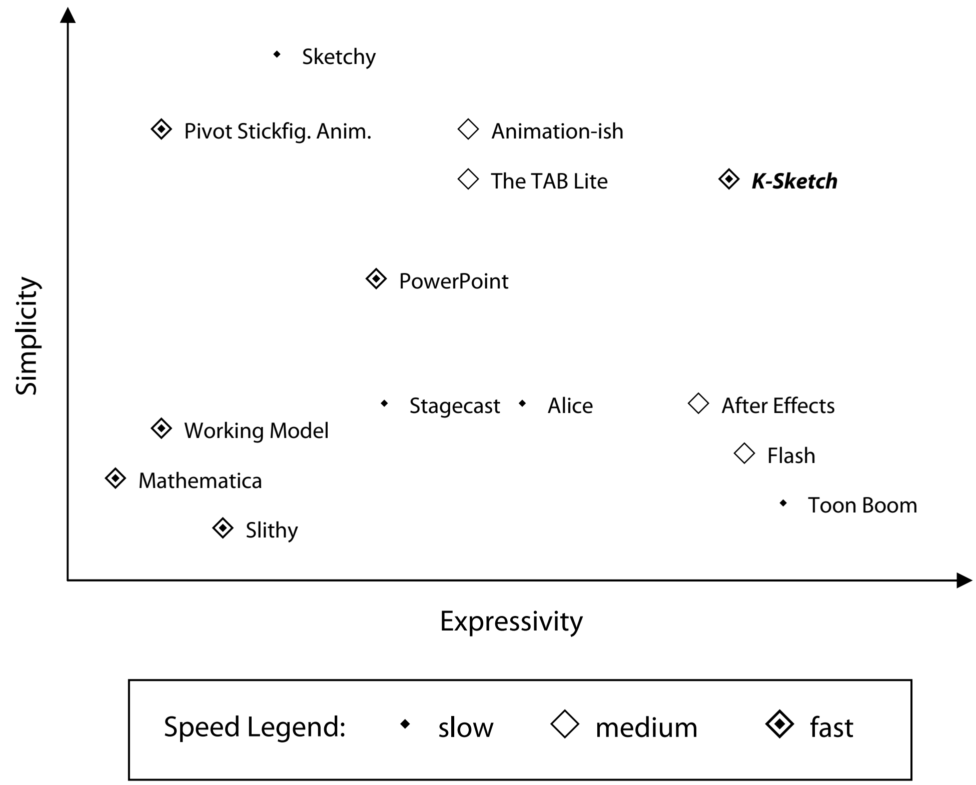


Figure 2-1: A qualitative graph showing the speed, simplicity, and expressivity of the animation tools described in this chapter. Most tools optimize for one or two of these variables, but K-Sketch optimizes all three.

2.2 Pen-based Interfaces: Why and How

Sketching is a powerful tool when we want our ideas to flow easily and quickly, and it has been the starting point for my design of K-Sketch. Sketching was also employed heavily by two animation tools described previously, Sketchy [59], which was the simplest tool I found, and Animation-ish [148], which attempted to strike a balance between speed, simplicity, and expressivity. In this section, I begin by describing design research and creativity support research that illuminates the importance of sketching. I then briefly review the history of pen-based system, describe previous attempts to apply sketching to animation, and close with tools and techniques that influenced K-Sketch's design and implementation.

2.2.1 Sketching in Design

When novices create animation, they are often engaging in a design process of some kind. When creating an animation tool for novices, then, it is helpful to consider how expert designers go about their work. Novices and experts are most alike in the early stages of a design process, because both face the problem of converting abstract ideas into concrete visuals. Experts differentiate themselves from novices more by how they formalize and disseminate their designs. This means that the early stages of the design process are most relevant to a novice animation tool, and these early stages nearly always involve sketching [29, 35, 53, 58].

Nigel Cross has collected knowledge about design processes from many fields. He describes sketching as an "intelligence amplifier" that helps designers "explore and resolve their thoughts" [35]. He has identified four reasons why designers sketch:

1. To handle different levels of abstraction simultaneously
2. To enable identification and recall of relevant knowledge
3. To assist problem structuring through solution attempts
4. To promote the recognition of emergent features and properties of the solution concept

This sheds some light on why sketching is important. It helps designers collect their thoughts and identify patterns to complete their task. The last item in the preceding list has also been studied by Goel [58]. He found that sketches are important because their roughness facilitates "lateral transformation" or the changing of one idea into another. Sketching not only helps designers collect their existing thoughts, then. It also helps them to discover new thoughts. If sketching is such an important part of the design process, then there is reason to believe that making sketching a part of the animation process could help novices create animations more quickly.

Eugene Ferguson is a design engineer who has collected the experiences of many in his profession [53]. His words reflect a pragmatic attitude toward sketching: "An engineering drawing starts with a series of freehand sketches.... The designer uses sketches to try out new ideas, to compare alternatives, and (this is important) to capture fleeting ideas on paper" [53, p 96]. He goes on to describe three categories of sketches used by design engineers:

1. *Thinking sketches:* These are used to clarify mental images and focus thoughts. Cross's four reasons for sketching fall into this category.
2. *Prescriptive sketches:* These serve as a reference for others who formalize the design in some way.

3. *Talking sketches:* Engineers create these constantly as they interact with one another. They create a common reference point for ideas under discussion.

During field studies I carried out as part of this research, presented in Chapter 3, I found scenarios in which *sketched animations* could play the same role as static sketches in all three of these categories. In some cases, the sketched animation is a step toward a more polished animation, but in others it is a step in the design of educational content, an engineering artifact, or even a dance routine. The possibility of an animation that plays the role of a talking sketch is particularly exciting. Is it possible that a tool could make animation so fast, simple, and expressive that non-experts could use it as a medium for spontaneous collaboration? This is one of the goals of the K-Sketch project.

2.2.2 Creativity Support Research

Since creating an animation is a creative process, it makes sense to ask what creativity support research has to say about sketching and supporting novices. If we were to use Schneiderman's genex framework [138] as a guide, we would say that K-Sketch fits in the *create* phase, when a user needs to explore solutions by connecting ideas, composing artifacts, and reviewing past work. It also makes sense to situate it in the *relate* phase, when users discuss ideas with peers and mentors. Sketching is a helpful practice in both of these phases.

Csikszentmihalyi has studied creativity as it relates *flow* or *optimal experience* [36]. He notes that there are limits to how much information the human mind can process, and people seeking an optimal experience must focus their attention and avoid distractions. "Whenever information disrupts consciousness by threatening its goals we have a condition of inner disorder, or *psychic entropy,* a disorganization of the self that impairs its effectiveness. Prolonged experiences of this kind can weaken the self to the point that it is no longer able to invest attention and pursue its goals" [36, p 37]. According to Csikszentmihalyi, then, focusing attention on the *right* things is so vital that failure to do so could prevent someone from even attempting a creative task. It is especially important to help novices at this stage, because they may still be learning where and how to direct their focus.

Csikszentmihalyi's prescription for achieving creative flow is learning to control attention, but other researchers have attempted to support creative flow by removing all "unnecessary" information from current design tools. These tools are often called *informal* tools (see Section 2.2.3). They frequently use sketching as the basic interaction but allow sketches to be transformed and linked in various ways. Further research has shown, however, that care must be exercised when interfering with the sketching process. Dekel and Herbsleb [45] studied the UML diagrams made by software engineers as they went through the early stages of their process. They observed that designers frequently make pen strokes that are not part of the diagram itself but are vitally important for solving problems or communicating with team members. Goel made similar observations for other types of designers [58].

The concept of flow and the variety of pen strokes in sketches both have important implications for the use of handwriting recognition technology in a sketching tool for designers. Any notification given to the user or any automatic transformation of ink runs the risk of giving the user too much information and disrupting flow. This means that handwriting recognition should be employed sparingly and carefully, if at all. (The designers of the DENIM informal web site design tool [101, 117], learned this lesson from recognition difficulties in SILK [95].) Further, sketching tools interfere with the creative process when they inhibit designers from making the pen strokes they would normally make. A tool that employs recognition can interfere both when it misrecognizes a stroke and when it causes a designer to hesitate for fear of a misrecognized stroke. For all of these reasons, K-Sketch avoids the use of handwriting recognition technology whenever possible and leaves all pen strokes in their original state.

Creativity support research also helps us understand how to support novices in particular. Davis and Moar [44] studied amateurs rather than novices, but their research is applicable to K-Sketch. They define “amateurs” as people who are content to remain unskilled and “novices” as people who are in the process of developing expertise. Their definition of “amateur” is not the most common definition: The word is more often used as the opposite of "professional," and some non-professionals develop the desire to improve their skill. Also, all amateurs are novices when they take their first steps. Davis and Moar's purpose in making this distinction is to point out the need for tools that are significantly simpler than current tools for experts, since some people will never be motivated to develop expertise. Since this is also a major theme of the K-Sketch project, Davis and Moar's directives for supporting "amateurs" are also appropriate for supporting K-Sketch's "novices."

Davis and Moar present insights into how creative tools can support amateurs. They argue that tools for amateurs can be designed by deriving them from tools designed for experts [44]. The process I employed in designing K-Sketch can be framed in their terms, because I looked at how animation is expressed in existing tools as a starting point for my own process. They list five transformations that can be applied to existing tools to produce amateur tools:

1. *Foregrounding:* An amateur needs the tool's most important functions to be readily accessible. Davis and Moar point out that choosing the most important functions requires the tool's designer to know what users will and will not want to do. Choosing the wrong functions results in a tool that is difficult to use. I developed interface optimization to address this problem (see Chapter 4).

2. *Backgrounding:* Amateurs need less important functions to be hidden or removed altogether. This is also the goal of interface optimization, to identify functionality that can be safely omitted from a tool without severely limiting its utility.
3. *Automation:* Amateurs need tedious tasks to be automated. K-Sketch automates tedious tasks in many ways. The copy motion operation, for example, automates the process of defining motion for many objects moving in similar ways.
4. *Constraining:* Amateurs need to be protected from making mistakes. One way K-Sketch protects amateurs from making mistakes is by constraining the reference frames that are available for new motions (see Section 5.1.4 on page 120). K-Sketch uses heuristics to choose a reference frame, and allows the user to correct this choice by picking the correct motion from a list. This avoids the need to set an explicit reference frame, which would create many opportunities for errors.
5. *Integration:* Amateurs need weaker functions to be integrated into more powerful functions. K-Sketch does this, for example, by integrating the definition of many key framed motions into a single demonstrated motion (see Section 5.1.3 on page 113). Integration can also be seen in the object

manipulator (see Section 5.1.3 on page 113) and simplified timeline (see Section 5.1.9 on page 141).

These design guidelines are as appropriate for a novice tool as they are for an amateur tool. Davis and Moar's design process is very similar to the one I have employed with K-Sketch, giving further evidence that it is a good process for supporting novices.

From the research presented so far in this chapter, we see that there is good cause to believe that there are many novices looking for a faster, easier way to create animation. Framing animation as an exercise in sketching gives them support at the earliest stage of the process, which is the most delicate, and it allows them to work quickly. The task that remains is to design a tool that integrates the best of existing tools and techniques.

2.2.3 A Brief History of Pen-based Interfaces

Before delving into the finer aspects of research in pen-based animation tools and animation techniques, it is useful to situate that research within the history of pen-based systems. I begin with a few examples of the earliest systems, and then describe various classes of systems that came into being. I close with a review of informal interface research, which is most relevant to this work.

The earliest work in pen-based interfaces was done in the 1960's at MIT's Lincoln Laboratory on a computer called the TX-2. Though the computer was the size of a room and had a primitive interface by today's standards, many of the ideas in today's pen interfaces were first developed on the TX-2. The first and most famous system was Ivan Sutherland's SketchPad [141, 142], which many call the origin of the modern direct manipulation interface. A lesser-known system is Ronald Baecker's GENESYS system [15, 16], which allowed an artist to sketch figures and animate them with hand gestures that were recorded in real time. GENESYS also had support for simple cel animation. The system was used to produce dramatic demonstrations, but as it evolved into SHAZAM [17], sketching and demonstration were discarded for techniques that would produce more polished end results. Now that pen-based hardware is widely accessible to novices who do not require polish, it makes sense to re-evaluate these older techniques.

More recent research in pen-based interfaces has tried to clarify the value of pen-based systems over the more successful mouse-and-keyboard systems. Many have enhanced the traditional processes of taking handwritten notes or annotating documents. Wang's Freestyle [55], the Digital Desk [118, 152, 153], and XLibris [122] all gave office workers methods for collecting handwritten notes and integrating them with traditional documents. Other systems have explored hand written note taking or annotation in classes or meetings (e.g., NotePals [42], LiveNotes [81], Classroom Presenter [10, 11], and eClass [1, 2, 26]) and found significant benefits in placing users' notes in context with their collaborators'. Some systems, such as Dynomite [155], Filochat [154], and Tivoli [108, 110], recorded audio as users took handwritten notes; the notes could then be used to access the audio that was playing at the time they were written. K-Sketch's animation interface could theoretically be integrated with all of these systems. Since my field studies (Chapter 3) uncovered situations in which sketched animations could be used to clarify ideas or serve as reference points for discussion, it makes sense to treat them like any other notes or annotations.

Other systems help people integrate disparate scraps of handwritten information and discover their structure. Tivoli was a whiteboard system for manipulating lists of information [109]. It eventually evolved *domain objects*, which were formal or semi-formal objects that integrated specific types of data provided special interfaces for manipulating themselves [111, 112]. A more general form of this idea could be found in Translucent Patches [89] and Flatland [73, 116]. Both of these systems provided environments for managing multiple regions of information, each with its own interpretation and behavior (e.g., math calculation, flow chart, or road map). K-Sketch's interface could be viewed as an animation behavior that could be inserted into such a system.

The pen-based systems most relevant to K-Sketch are those that support designers. The most successful of these systems have **informal** interfaces that support designers in the early stages when sketching is most prevalent (see Section 2.2.1).

These systems grew out of the observation that much of the complexity of conventional design tools comes from their focus on precise details. When these details can be ignored or deferred, design tool interfaces can be much simpler and can be operated more quickly. Consequently, these systems attempt to support designers with an array of traditional design tool features, but they do so with the assumption that thoughts will begin in a completely unstructured form.

Informal tools vary by their domain, how they support design, and how much structure users can add to their designs. SILK [94, 95], DENIM [101, 117], DEMAIS [19, 20], and Topiary [99] all supported user interface designers by allowing them to simulate interfaces and possibly test them on real users while still in rough form. All recognized the structure of certain interface components and allowed parts of the interface to be linked so that transitions would happen automatically during a simulation. DENIM specialized in web sites and had special features for defining and navigating through site maps. DEMAIS specialized in multi-media presentations and had means for handling timelines and narration. Topiary specialized in location-enhanced applications and gave designers a rough way to specify locations at design time and during simulations. Knight [38] helped software designers collaboratively construct detailed software diagrams. The tool recognized the structure of the diagram and allowed it to be exported to Computer Aided Software Engineering (CASE) tools. Finally, the Electronic Cocktail Napkin attempted to support designers in all disciplines by giving them an end-user programming environment for processing their own drawings [62].

K-Sketch is an informal tool that supports animators by giving them a fast and easy way to construct an animation. The animation process is similar to a sequence of editing steps in a conventional graphical editor, but it has been enhanced with real-time demonstration, a fast technique that takes advantage of users' intuitive sense of space and time. Users add structure to their sketches by making edits that play back instantaneously or in real time, by copying existing parts of the animation, and by changing the reference frame for a motion (see Section 5.1.4 on page 120). Note that K-Sketch differs from all of the informal tools mentioned here, because it is intended to support novices rather than experts and must therefore provide a more carefully chosen set of operations.

All of the informal systems mentioned here recognize the structure of users' diagrams as they work. Some make attempts to mitigate the disruption in creative flow (see Section 2.2.2) caused by such interruptions. For example, The Electronic Cocktail Napkin will only recognize patterns automatically after it is trained, and Knight only recognizes pen strokes when in a special mode. K-Sketch avoids most recognition in most situation by requiring users to specify actions explicitly through an object manipulator and a timeline (see Figure 1-2 on page 9) There are two types of operations that require recognition, however. The reference frame for new edit operations is determined implicitly by the context of the currently selected strokes. Also, a paste command may paste an object, a motion, or an object and a motion depending on the contents of the clipboard and the current selection. When any of these heuristics cause K-Sketch to do something unexpected, uses can press the **Fix Last Operation** button and pick the result they expected from a list (see Section 5.1.4 on page 120).

I have now established how K-Sketch relates to the major classes of pen-based systems. In the following section I will discuss how K-Sketch relates to specific systems that use sketching and gestures to define animation.

2.2.4 Sketching and Gesturing for Animation

The popularity of animation and the difficulty in creating it have made it an active area of research. Some have simply automated the traditional cel animation process (as in Tic-Tac-Toon [52]), but many have experimented with novel interaction techniques for defining animated movements. Since my work on K-Sketch is trying to move beyond existing capabilities, this section focuses on the latter class of research.

One common approach is to generate animation from static drawings. For example, KOKA defines a visual language that allows users to animate objects by drawing structured annotations on top of them [84, 143]. This can be effective as long as the language is simple, but such languages are usually complex. Further, visual languages are hard to learn, and they require heavy use of handwriting recognition, which is problematic (see Section 2.2.2). Living Ink avoids some of these problems by requiring users to specify an explicit mode for each pen stroke [130]. However, this system has a large number of modes and requires strokes to be drawn in a specific order, which can disrupt creative flow. Another unfortunate side effect of both of these systems is that timing is difficult to control, because the timing of strokes is discarded and neither tool has a clear timeline. For all of these reasons, K-Sketch does not generate motion from static sketches but from the demonstrated actions of users.

Another branch of research in animating from static sketches focuses specifically on directing the motion of complex figures (usually human). The Coach's Playbook used annotated diagrams of football plays to generate 3D simulations [121]. Motion Doodles generated complex human walking, running, and jumping motions from side-view sketches and a sketched motion path [147]. Its approach is somewhat closer to K-Sketch's, because it uses the timing of the motion path stroke, but the figure is not manipulated directly. James Davis and colleagues took a completely different tack, generating motion from sketches of skeletal figures at key frames [40]. These are intriguing approaches, but my interface optimization found that such motions are rarely required in rough animation scenarios (see Section 4.3 on page 95).

Some have investigating the use of morphing as technique for generating animation quickly and easily. Di Fiore and Van Reeth [47] investigated morphing as an in betweening method for cel animation. Their approach has some promise, but few professional animators use it because of the difficulty in generating a pleasing morph (see Section 2.1.3). This difficulty caused Vronay and Wang to develop a sophisticated method for generating pleasing morphs for sketched objects [151]. Igarashi, Moscovich, and Hughes tackled a similar problem by developing a system for animating 2D sketches by **deforming** them [75]. Deforming is sometimes used to produce the same effects as morphing, but motion is generated by stretching or bending a single drawing rather than generating the motion between two drawings. These approaches also have merit, but my interface optimization also found both morphing and deforming to be good candidates for omission from a novice's tool.

There is other research that that generates animation for specific domains. MathPad[2] was a learning tool for children that used sketches of math formulae to generate animations that gave a visual analog [97]. This approach is too limiting to be appropriate for a general purpose animation tool. Another set of tools generates physical simulations from static sketches [8, 9, 41, 82]. This is a more widely applicable approach, but again, my interface optimization found that any realistic motion could be approximated with other motions, making physical simulation a good candidate for omission.

The systems presented so far in this section all make use of sketching, either for defining the objects to be animated or their motion or both. A final class of related systems makes use of real-time hand gestures to define the motion of objects. Some formal 2D animation systems allow users to record mouse drag operations as a short cut for defining key frames (e.g. Pavlov [157]). This is very close to what K-Sketch does, but it is far more useful in K-Sketch because users have an expectation of roughness. Some have investigated the use of real-time mouse gestures for directing the movement of 3D characters, either to make it more accessible to novices [145] or to give animators an easier way to prototype [49]. These are the goals of K-Sketch, though my domain is 2D rather than 3D.

The work that is closest to K-Sketch comes from Moscovich, Hughes, and Igarashi, who have developed systems for demonstrating the motion of 2D sketches in real time. Race Sketch [113, 114] handled only the translation of single-stroke objects, but the interaction felt very similar to that of K-Sketch. Race Sketch also included a function for fine-tuning the synchronization of events. K-Sketch omits this in favor of the less precise but more general method of speeding up or slowing down the entire animation for easier coordination (see Section 5.1.8 on page 140). Race Sketch also had tools for defining skeletal structures and demonstrating deformations in real time. This work was later refined into a system that allowed sketched objects to be animated by stretching and moving them with several fingers at once on a multi-touch display surface [75]. As mentioned previously, K-Sketch omits deformation in favor of other operations. It also avoids the use of multi-touch displays in favor of commodity hardware like Tablet PCs.

With so many novel pen-based and gesture-based animation techniques in existence, it seems natural to compare K-Sketch to these systems when conducting summative evaluation (see Chapter 7). Unfortunately, most of these tools were not available for comparison (Assist [8, 9] and Race Sketch [113, 114] are notable exceptions). Even if more were available, a comparison with these tools would not be appropriate. Their purpose is to illustrate the value of particular interaction techniques, while the purpose of the K-Sketch project is to identify the *set* of techniques that best serves novice animators. The tools presented here have not reached the level of refinement necessary to make such a claim. For example, nearly all are missing important features like *undo and redo,* and only Living Ink handles more than a quarter of the scenarios I collected in my field studies (see Table 4-2 on page 101). Therefore, the best way I can take advantage of this body of research is to take the best animation techniques and show how they stack up against other systems that make reasonable attempts to serve novices (e.g., PowerPoint [106] and The TAB Lite [48]).

2.2.5 Tools and Techniques

Now that we have examined sketching and gesturing techniques for animation systems, let us quickly review other research in tools and techniques that is related to

K-Sketch. I begin with research in techniques for selection and issuing commands. I then list several toolkits that have been developed for sketching systems. I close with a look at techniques for reviewing edit histories and suggestive interfaces.

K-Sketch uses a simple test for ink selection. If more than sixty percent of the points in a stroke lie inside a selection loop, then it is selected. There are other approaches, however. Sloppy Selection used human motor control limitations to refine decisions about what strokes to include in a selection [96]. PerSketch analyzed the structure of sketches and automatically grouped strokes as they human perception would normally group them [135]. These are both fascinating technique that I may have used if they were available to me. PerSketch does, however, run the risk of disrupting creative flow when grouping does not match a user's expectations (see Section 2.2.2). It is also worth noting that K-Sketch users must hold a physical mode switch to alternate between drawing and selecting modes. This style of mode switching is one that Li and colleagues found to be particularly effective [98].

Edit commands in K-Sketch are issued through a novel object manipulator (see Figure 1-2b on page 9), which integrates many commands in one control (as in integrated manipulation [70]). It is visually similar to a tracking menu [54], but it remains fixed on top of an object rather than following the pen as tracking menus do. Because this manipulator is a transparent overlay, it also appears similar to a toolglass

[23], but toolglasses are usually intended for bimanual input, and K-Sketch is designed for a single pen.

Other commands in K-Sketch are issued through a standard toolbar or context menu. There are many interesting alternatives to toolbars and context menus, particularly marking menus [92, 93, 163], which K-Sketch does not use because there is no good way to distinguish menu commands from selection. Scriboli [69] elegantly combines stroke selection with command selection when editing static documents, but this technique is awkward for issuing real-time animation commands.

There has also been interesting research in toolkits for sketching interfaces. SATIN provides a scene graph, useful pen-based widgets (such as pie menus), and support for managing ink recognition engines [71]. Oops provides tools for managing ink recognition errors [104]. Since K-Sketch uses no ink recognition technology, neither toolkit was particularly helpful. Instead, K-Sketch uses Microsoft's Tablet PC SDK [76] for its responsive ink interface and Piccolo.NET for its scene graph management tools [21].

Because K-Sketch animations are built with a sequence of graphical editing steps (see Section 5.1.2 on page 112), K-Sketch resembles other systems that keep track of edit history. WeMet [128] and The Visual Knowledge Builder [137] both have history slider bars that are very similar K-Sketch's time slider bar. TimeWarp [51] shows a graphical history of events similar to the event history shown on K-Sketch's timeline. The Designer's Outpost [87] also distinguishes between the main document history and the history of a particular object as K-Sketch does. K-Sketch differs from all of these systems, however, because users control when time advances and they can make edits to previous points in time without causing branches in history. In this latter respect, K-Sketch is somewhat closer to Chimera [91], which allows users to review and selectively reuse parts of the edit history.

Finally, K-Sketch's fix dialog, which allows users to correct the reference frame for a new motion (see Section 5.1.4.3 on page 129), is closely related to suggestive interfaces. These interfaces were pioneered by early systems that infer graphical constraints in drawing and graphics interfaces [57, 83] and were further popularized by Igarashi's Chateau system, which facilitated the construction of 3D models [74]. When the user highlighted parts of the model, the system would suggest possible operations through an array of small thumbnails showing the results of those operations. This technique has been imitated in many design tools. Some show alternatives whenever the system makes an automatic choice [40]. Some show alternatives only when the user is about to execute a command [146]. Since K-Sketch's fix dialog is not frequently needed, K-Sketch avoids disrupting creative flow by showing alternatives only when a user issues a fix command.

This concludes my review of research in pen-based design philosophies, tools, and techniques. We have seen that sketching plays an important role in design, and it has the potential to facilitate a novice's creative flow. Many of the ideas in K-Sketch are borrowed from a growing tradition of informal interfaces. We have also seen that K-Sketch borrows many existing interaction techniques and seeks to make a contribution by determining the best set of techniques to include in a tool that supports novice animators.

2.3 Motor and Perceptual Issues in Animation Sketching

The evaluation of K-Sketch presented in Chapters 7 and 8 shows that sketching and demonstration are effective techniques for creating animations. It is instructive, however, to see how this evaluation squares with the current understanding of human capabilities. I conclude my review of related work by examining two areas of work related to K-Sketch, research in human perception of animation and motor control research.

2.3.1 Perception of Animations

In my attempts to understand how the imprecise nature of sketched animations might interfere with a viewer's understanding, I have found extensive research that shows what people perceive from motion. Some of this research is focused more on low-level perceptual issues. Through a series of experiments, Michotte mapped out the conditions under which people perceive causation when one object (A) hits another (B) [105]. One of his findings is that people will perceive B's motion to be caused by A if B begins to move within 200 milliseconds of the collision. Also, B's trajectory must be within 25° of A's trajectory, and the distance between A and B at the time of the collision should not be more than two millimeters (if the screen is half a meter from the viewer and A moves at six centimeters per second). We will revisit this research in the following section when we consider whether human motor control can demonstrate motion within these thresholds.

A few others have studied low-level perceptual issues in animation. Bertamini and Proffitt studied human ability to recognize various motions in random-dot fields [22]. Their work shows that translation is easier to perceive than rotation or divergence, which may help to explain why translation was far more common in the animations I collected.

Other research has focused on the perception of higher-level information from motion. Heider and Simmel, for example, showed that viewers perceived similar stories when viewing a silent, two-minute animation of simple shapes moving over a white background [68]. This work shows how even simple motions can evoke a powerful response. Other researchers who study naïve physics have found that people who are unable to choose the physically accurate example from static diagrams of moving objects will choose the correct example from simple animations [78, 79]. This work shows how animations can tap into intuitive knowledge that is otherwise unreachable. This effect is lost when an animation has too many moving objects or the objects themselves are too complex [80]. This could indicate that the simplicity encouraged by rough animation has perceptual benefits.

Finally, Tversky compared the ability of viewers to perceive information from both static diagrams and animated diagrams [150]. She found that viewers can miss important events in an animation and suggests providing viewers with fine grain control over the animation, including starting, stopping, moving to arbitrary points in time, and controlling playback speed. K-Sketch provides all of these controls to animation builders and viewers alike.

2.3.2 Motor Control Issues in Animation Demonstration

K-Sketch is built on the assumption that hand gestures are a natural way for people to express their moving mental images. As mentioned in Section 2.1.2, Hegarty and colleagues also showed that hand gestures can help people understand their moving mental images. In their experiments with students solving mechanical reasoning problems, they found that students who were allowed to make had gestures performed better [67]. However, this still leaves open the question of whether or not people can make hand gestures that others can understand. One way to address this question is to look at how motor control research connects to Michotte's work on perception of causality, presented in the previous section.

In the field of Human-Computer Interaction, serious work has gone into the development of engineering models of human behavior, and these models can shed light on motor control issues related to sketching. One of the most popular engineering models is the model human processor [30]. This computational cognitive model can predict the time needed to accomplish tasks when those tasks are broken down into low-level steps. It includes long-term memory, short-term memory, sensory image stores, and various processors (cognitive, perceptual, and motor) with different "cycle" times.

Using the model human processor as a guide, we can begin to estimate a typical user's ability to demonstrate an animation of one object (A) colliding with another (B). Michotte found that a viewer will perceive causation if object B begins to move within 200 milliseconds of being hit by object A. In K-Sketch, a user would typically demonstrate A's motion first and then demonstrate B's motion, waiting for the right moment to begin moving his hand. The model human processor would break this user's action into two steps, recognizing the need to move his hand (one perceptual processor cycle), and beginning to move his hand (one motor processor cycle). This gives an estimated reaction time of 70-300 milliseconds. This indicates that some users will be able to respond in time, but others will not. K-Sketch addresses this problem by giving users the ability to slow down the animation and demonstrate motions at a more manageable speed.

The Steering Law [3] is a more recent engineering model that shows how steering accuracy decreases as hand speed increases. The model assumes that users are attempting to move their hand along a particular path and predicts the maximum distance a user's hand will stray from this path given the instantaneous speed. The steering law predicts that a hand moving at six centimeters per second can stay within 1.75 millimeters of an ideal position, which is within Michotte's two millimeter threshold for perception of causality when object A hits object B. As speed increases, hand control gets worse, but the threshold for perception of causality also increases. The steering law also predicts that keeping B's trajectory within 25° of A's trajectory will not be difficult unless B moves a very short distance in response to being hit. These results bode well for demonstrated animation.

We can infer from this research that viewers will be able to correctly interpret demonstrated animations, even difficult animations such as collisions. Demonstration does have limits, but slowing down an animation can theoretically bring any animation within those limits. It is important to keep in mind, however, that novices can most frequently accomplish their goals without the level of precision that is being pursued in this section. The perception of causality in collisions is seldom vital to tell a story. When causality is vital, viewers can be aided in this perception by other context, such as discussion that may be occurring as the animation is being made.

2.4 Related Work Summary

All the research presented in this chapter points to the need for K-Sketch. Novice animation in entertainment is a small but growing segment, and both teachers and students can benefit from becoming novice animators. Novices are most likely to become lost and give up at the beginning of the animation process, and they need a fast, simple, and expressive tool to help them get started. K-Sketch supports novices both by making careful decisions about what features to include and by framing animation as a process in sketching. Sketching is the most common way to begin any design process, and a growing tradition of sketching tools, particularly informal interfaces, shows how to support the creative process without interfering with it. As the following chapters will show K-Sketch succeeds in these goals and allows novices to use animation in ways that were not previously possible.

3 Field Studies

System design processes usually begin with collection and analysis of user tasks [119]. Collecting and analyzing tasks is fairly simple when a new system will automate an existing process, but there is no existing process for sketching animation. I approached this problem by interviewing people who were potentially interested in sketched animation and looking for patterns that applied to novices. This chapter summarizes these interviews. A more detailed presentation of the data from these interviews can be found in Appendix A.

I first interviewed people with animation experience to better understand their work processes (Section 3.1). Then I interviewed people who wanted to create animations but had no animation experience to better understand what they were trying to do and why they were unable to do it (Section 3.2). From these interviews, I created detailed usage scenarios for an animation sketching tool (Section 3.3). The analysis of this library of scenarios forms the foundation of this project, particularly the set of animation operations I define in Section 3.3.4. These are the basic building blocks for an informal animation sketching tool.

#	Occupation	Years anim. experience	Media	Techniques	Tools	Interest in informal animation
1	artist, animator	10	clay, drawings	key frames	AfterEffects, Flash	prototyping
2	animator	25	drawings	cel animation	AfterEffects	prototyping
3	animator, animation teacher	13	drawings	key frames, cel animation	AfterEffects	prototyping, learning tool
4	computer science graduate student	2	vector graphics	programming	Slithy	none (does not prototype)
5	computer science graduate student	1	vector graphics	programming	Slithy	prototyping
6	producer, animator	11	vector graphics, images	key frames	AfterEffects	prototyping
7	web designer	12	vector graphics, images	key frames	Flash	none (prototypes in Flash)
8	producer, animator	3	drawings	cel animation, deformation	n/a	medium for finished works

Table 3-1: Summary demographics and results from interviews with animators.

3.1 Interviews with Animators

Since many novice animators wish to do what experienced animators already do, I started by interviewing experienced 2D animators to see how an informal tool would fit into their work process. A summary of the results from these interviews appears in Table 3-1. The animators fell into three rough categories. Animators 1–3 were *traditional animators:* They made short films and commercials using traditional techniques such as key frames and cel animation. (Animator 3 also taught classes in animation to amateur adults and children of all ages.) Animators 4 and 5 were computer science graduate students who made highly polished animations for conference presentations by programming them with the Slithy animation description language [164]. I refer to them as *scientific animators.* Finally, the *new media animators* (animators 6–8) produced short animations while working on broader multi-media projects such as interactive DVDs (animator 6) or web sites (animators 7 and 8).

Interviews were loosely structured or not structured at all (in the case of Animators 7 and 8). Discussions with Animator 8 were conducted entirely over e-mail, while the others were primarily in person with some follow-up e-mail. All animators were told that the purpose of the interview was to guide the design of a tool for quickly sketching rough animations. During the interviews, I recorded how animators went about the various steps in their process, collecting sketches, photographs, animations, and video footage of the animators at work whenever possible. In the loosely structured interviews, I also asked the following questions:

1. How long have you been working as an animator?
2. Could you describe the steps in your work process? Give greater detail on the early parts of the process and the parts that involve sketching.
3. What hardware or software tools do you use to do your work?

As seen in Table 3-1, most of the traditional animators and new media animators were very experienced, although the new media animators had a harder time counting their experience with animation per se. Animator 6 included seven years of schooling in his count. The scientific animators were much less experienced than the others and could be classified as amateurs. Animator 4 had produced four conference presentations (see Figure 3-1), while animator 5 had produced a single presentation including two lengthy and complex algorithm animations (see Figure 3-2).

All but two animators began their process by sketching. The first sketches were often done alone just to kick around ideas or else in the company of a client to get immediate feedback on design ideas. For a scientific animator, sketching helped to clarify whether or not a visual image communicated the important aspects of a process being animated, as in Figure 3-3. For other animators, *character sketches* were among the first to be created. Static character sketches like the one shown in Figure 3-4 were made to clarify static visual details. Moving character sketches mimicked the

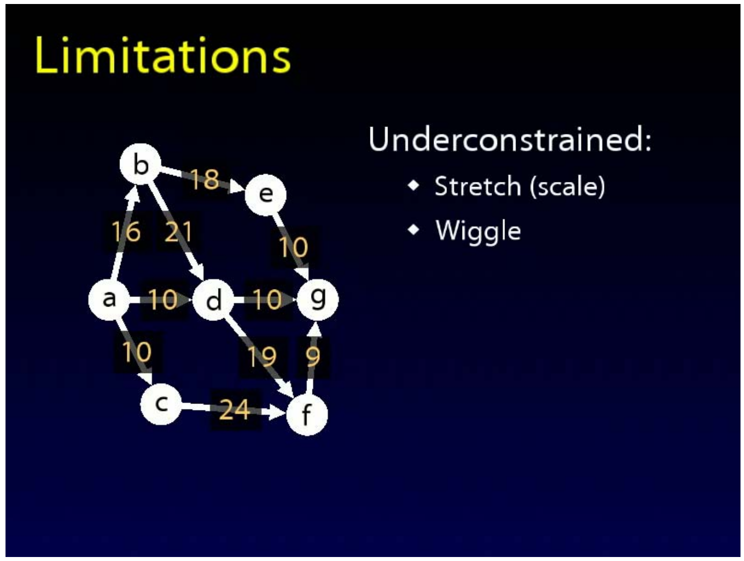


Figure 3-1: An animated presentation from animator 4. Nodes *b* and *f* stretch out from the center while all arrows remain connected. The numbers change as the arrows stretch.

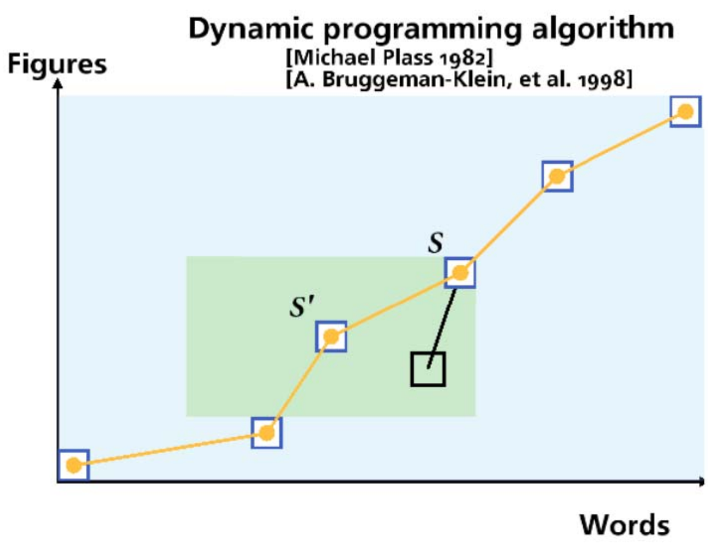


Figure 3-2: *Dynamic programming* animation (professional, prototype) from animator 5. The black box scans through the shaded region line-by-line. A black line keeps the box connected to node *S*.

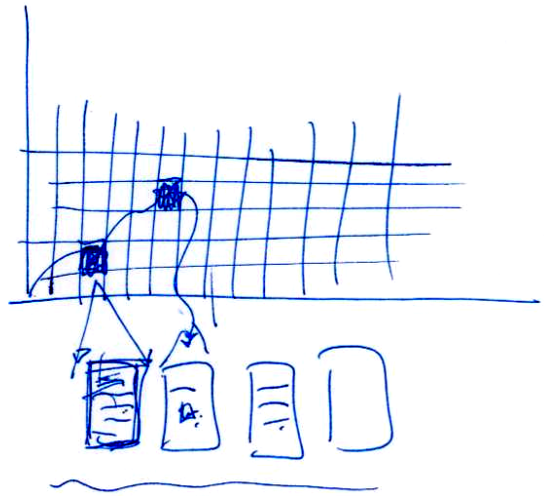


Figure 3-3: Sketch made by animator 5 that tests an alternate approach to the animation in Figure 3-2. Nodes are set in a visible grid, and details explode out of them.

cel animation process and clarified how a character would move. Animators often made these sketches on thin, overlapping sheets of paper placed on a light box, which allowed them to see all frames at once or view them in motion as in a flip book (see Figure 3-5). Sometimes these moving character sketches were created on a single page, as in Figure 3-6 and Figure 3-7.

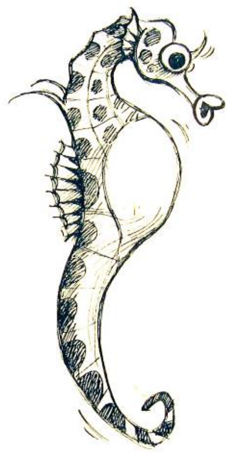


Figure 3-4: Static character sketch of the *Potbelly* character from animator 1. Static character sketches help animators work out visual details.

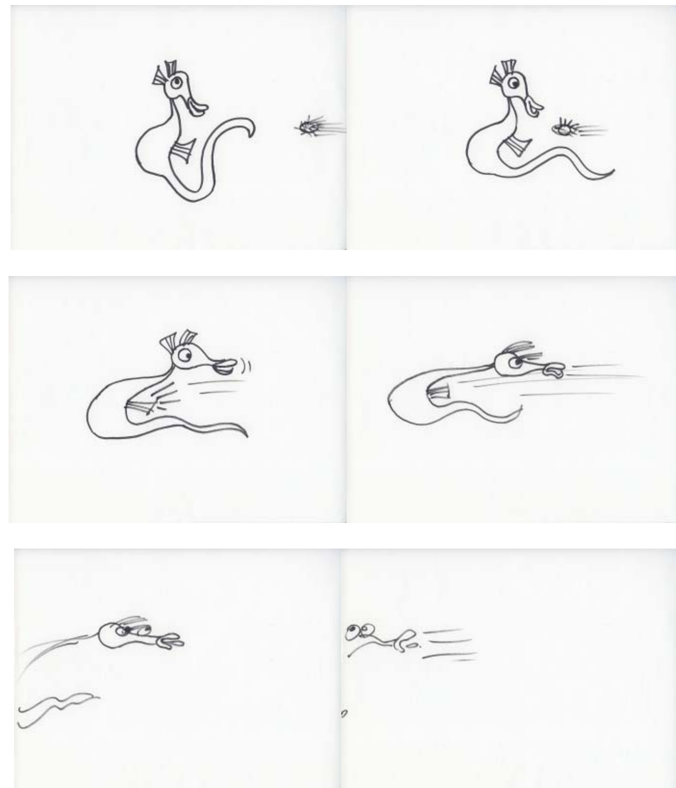


Figure 3-5: *Potbelly collision* animation (artist, prototype) from animator 1. This animated character sketch helps the animator work out the motion of the character in Figure 3-4. All sketches are on thin sheets laid over a light box so that the animator can see all frames at once or flip through them to animate.

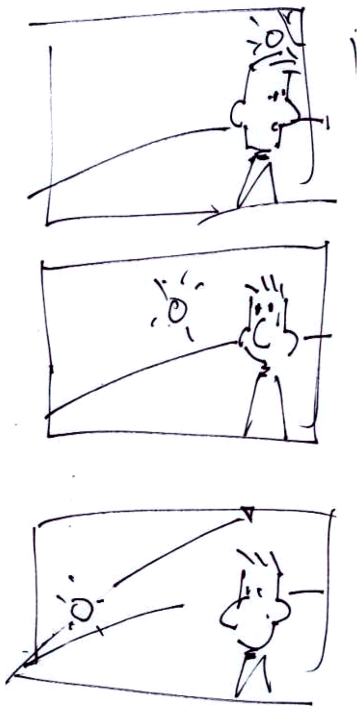


Figure 3-6: *Man watching sun* animation (artist, prototype) from animator 2. This is an animated character sketch on a single sheet.

Figure 3-7: *Car bobbing* animation (artist, prototype) from animator 2. This is an animated character sketch on a single sheet. The letters show experimental cel cycles.

For all but the shortest or simplest animations, the next step in these animators' processes was *storyboarding*. A storyboard is a sequence of images that show how a story evolves over time (see Figure 3-8 and Figure 3-9). Usually, each frame of the storyboard is accompanied by text that explains the action of the frame or the speech and sound heard while it is visible.

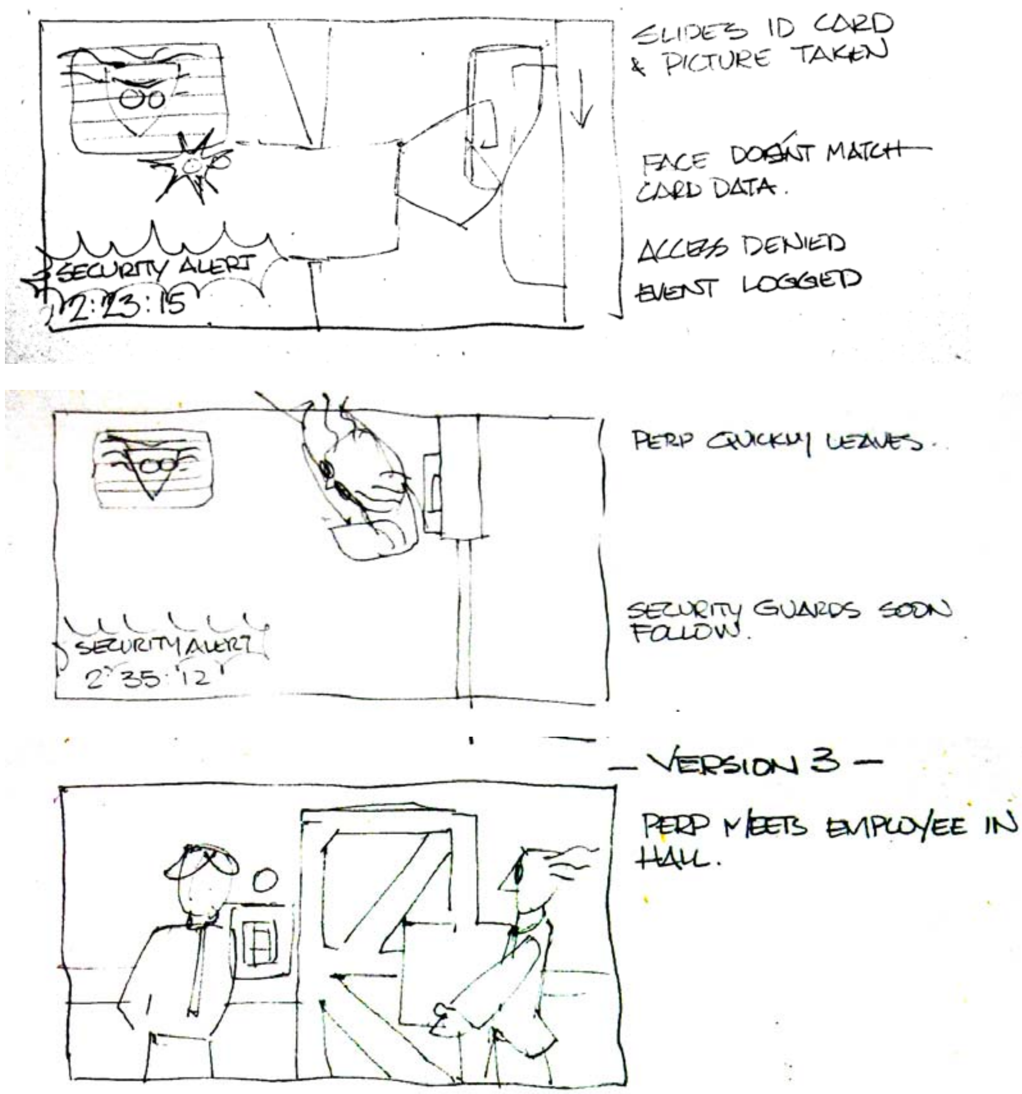


Figure 3-8: Storyboard sketches from animator 2. Top: *Security card slide* animation (artist, prototype). Middle: *Perpetrator turning* animation (artist, prototype). Bottom: *Perpetrator entrance* animation (artist, prototype).

Figure 3-9: Storyboard frame from animator 1 showing the *Simple potbelly crash* animation (artist, prototype).

Two animators took the additional step of producing *animatics*, which are rough animations that help animators work out the timing of events. Animatics also help clients to comment on animation projects. The animators I interviewed produced their animatics by scanning their drawings into a computer and then animating them in AfterEffects. Most animatics were as simple as a timed sequence of storyboard frames, usually with a rough sound track. Some animatics also implied action by shaking the frame at key points. In a few cases, clients required more detailed animatics with colored drawings and key framed animation in each panel.

When it came time for final production, each animator reached for the tools most appropriate for their target format. The scientific animators coded their animations in Slithy. The others would most often use Flash for web productions and AfterEffects for video productions. The traditional animators' processes were the most involved.

Figure 3-10: Animation stand used by animator 2. This facilitates lighting and moving of animation cels during final photography.

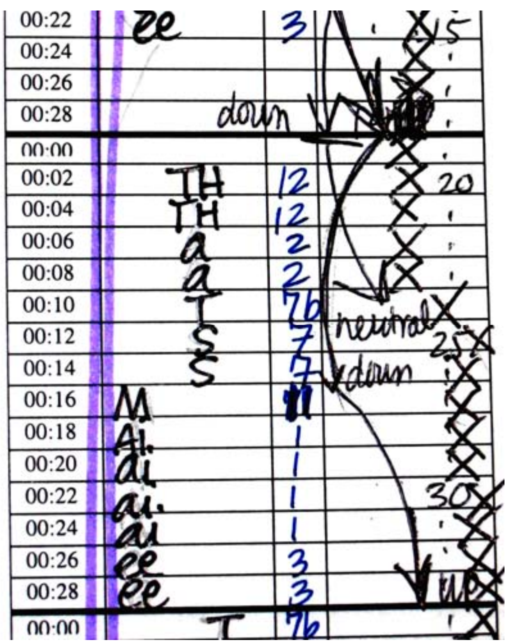


Figure 3-11: Close-up view of a timing sheet from animator 1. This sheet lists each frame of the animation for coordination of events and for tracking progress.

Animator 3 replaced parts of her animatic with more refined parts until the production was complete. One animator made film productions using cel animation techniques, which required the use of an animation stand, shown in Figure 3-10. A common tool used by traditional animators was a timing sheet (see Figure 3-11), which is a frame-by-frame map of an animation that helps animators track their progress and manage the relationships between frames.

As the figures in this section show, most animators sketch throughout the early stages of their production process. However, two animators had developed such a level of skill with their tools that they did not always need to sketch. Animator 4 worked exclusively in Slithy to make fairly simple animations that followed a limited set of patterns. He was so proficient in programming Slithy that he often skipped storyboarding and moved directly into coding his animations. Animator 7 was so experienced with Flash that she was often able to mock up animations directly in the tool after only a few sketches. When she did create storyboards, she created them in Flash using vector graphics. Both of these animators were able to avoid sketching because they had stable work processes and deep knowledge of their tools. Their experience is further evidence that sketching is particularly important for novices, because they do not have the benefit of such experience.

At some point in each interview, I described possible designs for a rough animation tool, suggested ways that the animator might be able to use such a tool in their work process, and noted their reactions. As Table 3-1 shows, most were interested in such a tool as a prototyping aid. Traditional animators saw it as a faster, easier way to make moving character sketches. Animators 1, 2, 5, and 6 also saw it as a faster, easier way to create animatics. All said that an animation sketching tool would be particularly valuable if they could prototype while meeting with clients, because it would help them to communicate their ideas and allow them to go through many design iterations in one sitting.

Two other reasons were given for wanting a rough animation tool. Animator 8 wanted a tool that would allow him to direct deformations of a string-like character using hand gestures. In this case, animation sketching is not for prototyping but rather for a finished work. Animator 3 taught animation classes for children and asked specifically for a demonstration-based animation interface, because it matched children's intuitions. Her students frequently act out the action of characters in front of the camera when recording an animation frame by frame. Some do this to test out motions, and others do it because they haven't grasped how frame-by-frame animation actually works. This is a use case that is specifically for novices, and it overlaps well with the goals of prototyping.

These interviews show that animation sketching could play an important part in an animator's work process. Sketched animation could be particularly helpful for prototyping character sketches and animatics, and it could make meetings with clients more productive. There is also further evidence that a demonstration-based animation sketching tool would help novices by matching their intuitions about animation.

3.2 Interviews with Non-Animators

In the previous section, we saw many examples of how sketched animation can help someone who is animating as part of a larger, professional production. However, many people have modest goals for their animations and would be content to remain relatively unskilled. Some want animations that they will show only once to illustrate a concept or to amuse a friend. These uses cannot justify the investment of the weeks

#	Occupation	Domain	Animation descriptions
1	education graduate student	biology	student exercise: meiosis
2	education graduate student	physics	detection of distant planets
3	mechanical engineering professor	engineering	dislocations in molecular structure, gear interference
4	computer science graduate student	dance	sequence of contra dance moves
5	chemistry professor	chemistry	particle collisions, rust and battery reactions
6	control systems researcher	engineering	construction equipment tread motion
7	college math instructor	math	cantor set construction
8	reading tutor	reading	visual rewards for correct answers
9	aeronautical engineer	engineering	robot arm moving around airplane
10	engineering manager	engineering	box sliding into casing
11	geochemical researcher	geochemistry	etalon noise

Table 3-2: Summary of demographics and results of interviews with non-animators.

or months necessary to become proficient with tools like those presented in the previous section. A fast, simple, and expressive animation tool for novices has the potential to remove this barrier.

To better understand the needs and background of such animators, I conducted interviews with people who avoided current animation tools but had specific examples of animations they wanted to create. All of these people were found by word of mouth or by chance meetings, and all were looking for fast, easy ways to create animations that they envisioned. I recorded interviews only with people who met the following criteria:

1. They must describe their animations in sufficient detail that I could create them.
2. There must be a plausible reason why they do not use existing tools to create their animations.
3. Their animations must support specific tasks.
4. There must be a plausible reason why their animations are necessary to accomplish those tasks.

Table 3-2 presents the diverse picture that emerged from these interviews with non-animators. Of the eleven people who met my criteria, six were educators. Non-animators 1, 2, and 8 were involved in the education of children, while non-animators 3, 5, and 7 were college-level educators. The remaining five worked in science and engineering disciplines. These non-animators described scenarios in which animations from a variety of domains could be created to accomplish a variety of goals. The remainder of this section will give an overview of these scenarios.

Figure 3-12: *Detection of distant planets* animation (teacher, explain) from non-animator 2, as sketched by me. Waves grow out of a star as a small planet revolves around it.

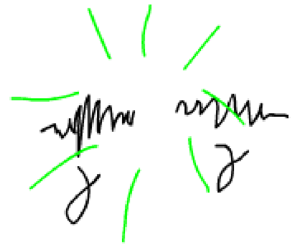


Figure 3-13: *Chemistry: particle collisions* animation (teacher, explain) from non-animator 5, as sketched by me. Particles collide and explode, producing gamma rays.

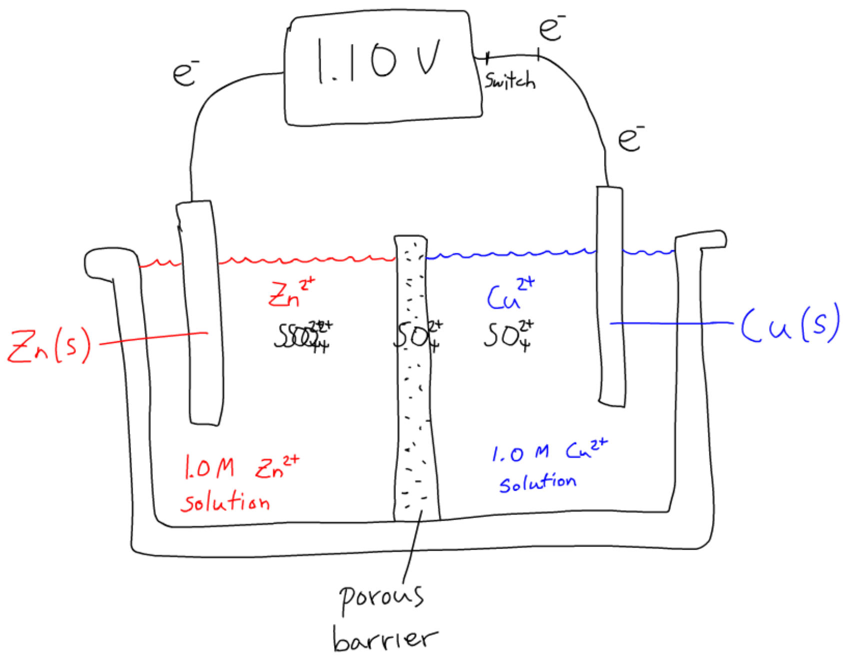


Figure 3-14: *Chemistry: battery reaction* animation (teacher, explain) from non-animator 5, as sketched by me. SO_4^{2+} ions move across the porous barrier in the center as e^- ions move along the wire at the top.

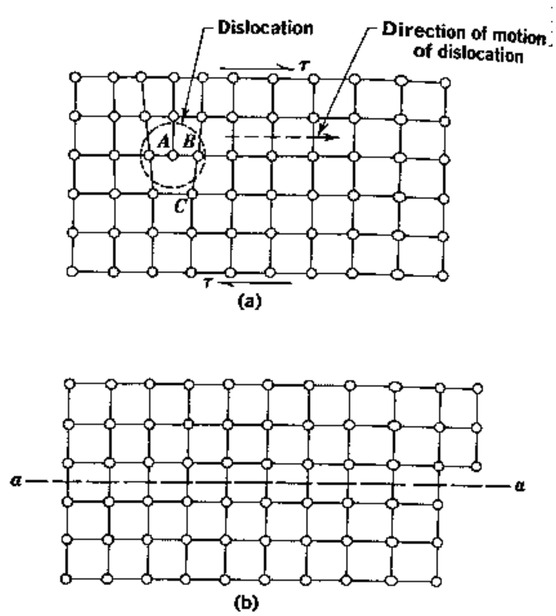


Figure 3-15: Lattice slip animation (teacher, explain) from non-animator 3. Molecule *C* breaks its connection to *B*, connecting instead to *A* and causing a chain of similar changes that propagates to the right. Non-animator 3 referred to this image to describe an animation he wanted to create. *Source:* Richards CW, *Engineering Materials Science*. San Francisco: Wadsworth, 1961, p 78.

The largest number of scenarios came from teachers or professionals seeking a better way to explain a concept to their students (e.g., Figures 3-12 through 3-16). Non-animator 8 tutored young children in reading, and wanted to create little animations as visual rewards for children when they answer questions correctly (see Figure 3-17). Non-animator 1 was not interested in animated illustrations, but did want her students to create animations of cell processes as a learning exercise.

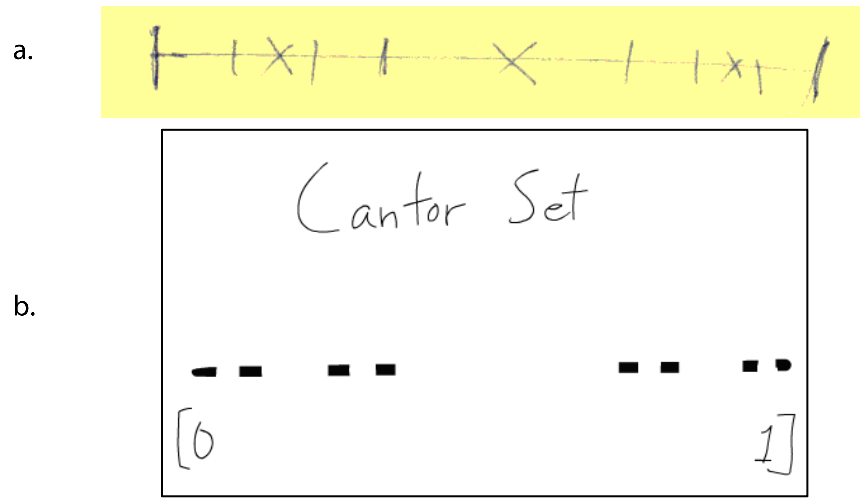


Figure 3-16: *Cantor set construction* animation (teacher, explain) from non-animator 7. Parts of the line disappear in stages as time progresses. *a:* A sketch drawn by non-animator 7. *b:* An animation I produced to verify scenario details.

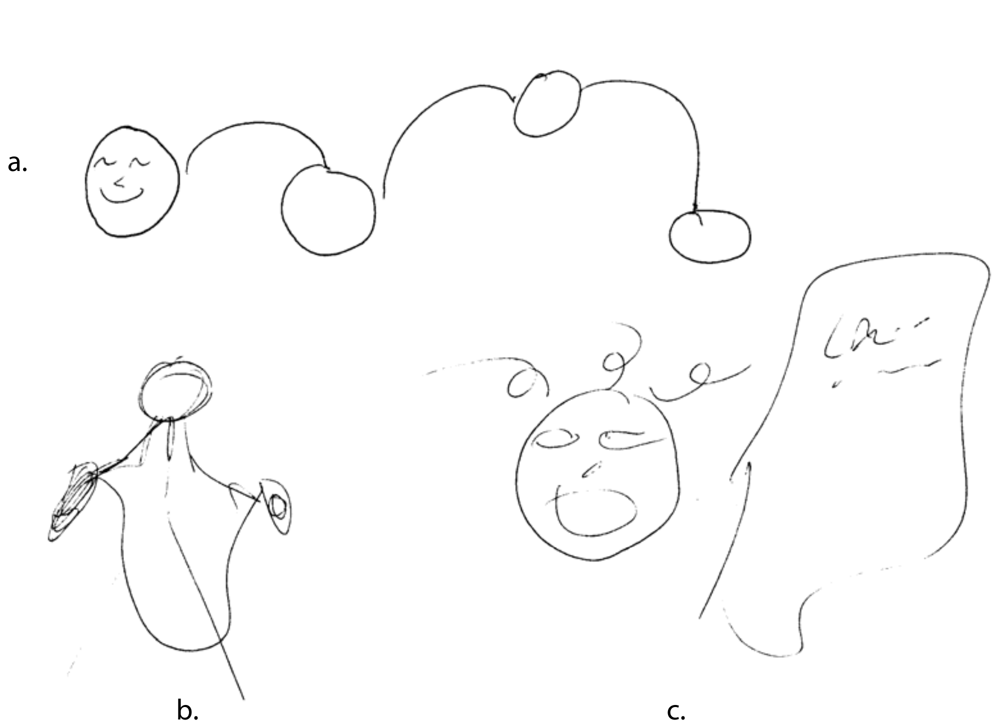


Figure 3-17: Sketches drawn by non-animator 8. *a: Bouncing ball* animation (teacher, entertain). *b: Jumping rope* animation (teacher, entertain). *c: Face singing* animation (teacher, explain).

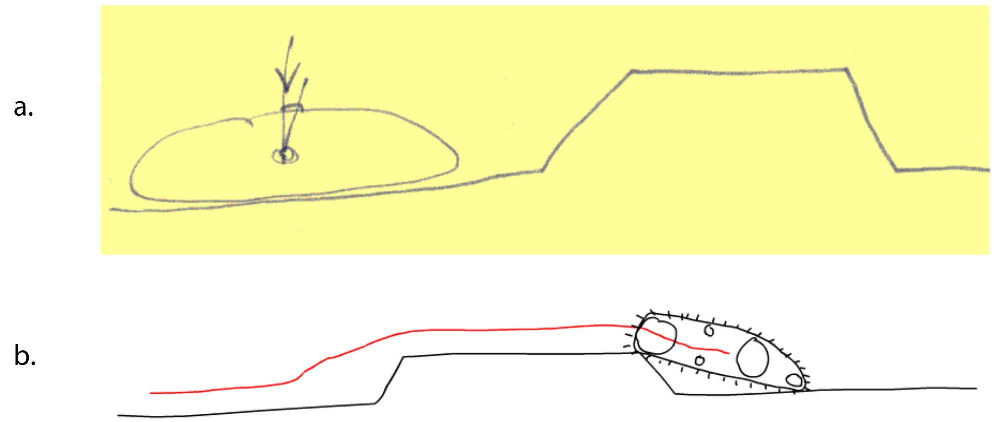


Figure 3-18: *Construction equipment tread* animation (professional, explain) from non-animator 6. The tread traces out a line as it rolls over the bump. *a:* A sketch drawn by non-animator 6. *b:* An animation I produced to verify scenario details.

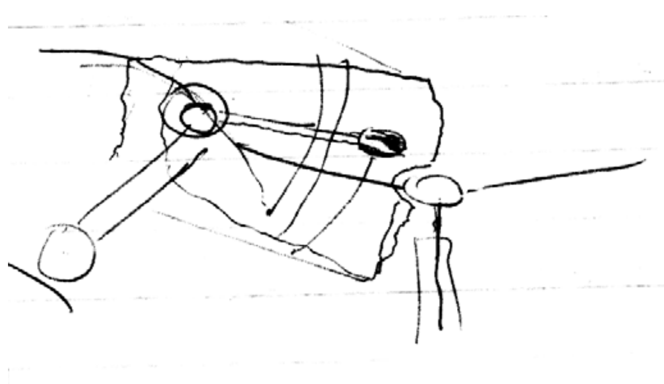


Figure 3-19: *Robot arm* animation (professional, explain) sketched by non-animator 9. The robot arm sweeps a sensor array around the wing of an airplane within the marked region.

Educators were not the only people interested in animation sketching, however. Four non-animators were professionals seeking to explain ideas in small, informal meetings (see Figures 3-18 through 3-21). Non-animator 4 was a contra dance hobbyist interested in creating an animation for his own visualization.

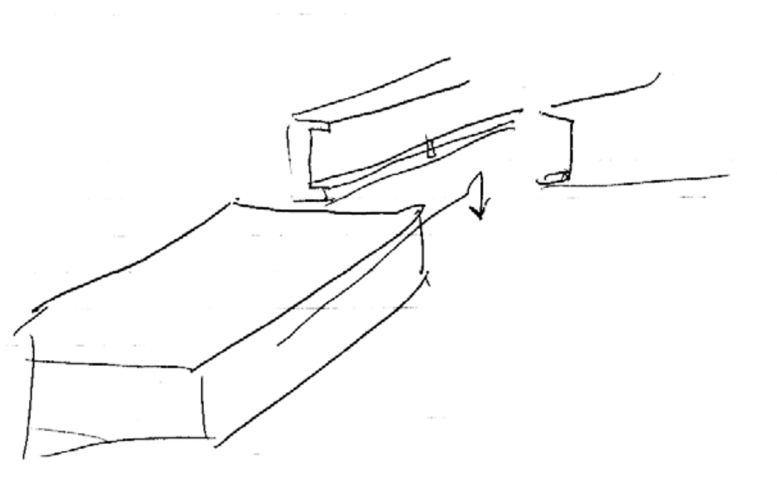


Figure 3-20: *Casing slide* animation (professional, explain) sketched by non-animator 10. An engineer explains to a colleague that the box should slide into its casing by going back and then down.

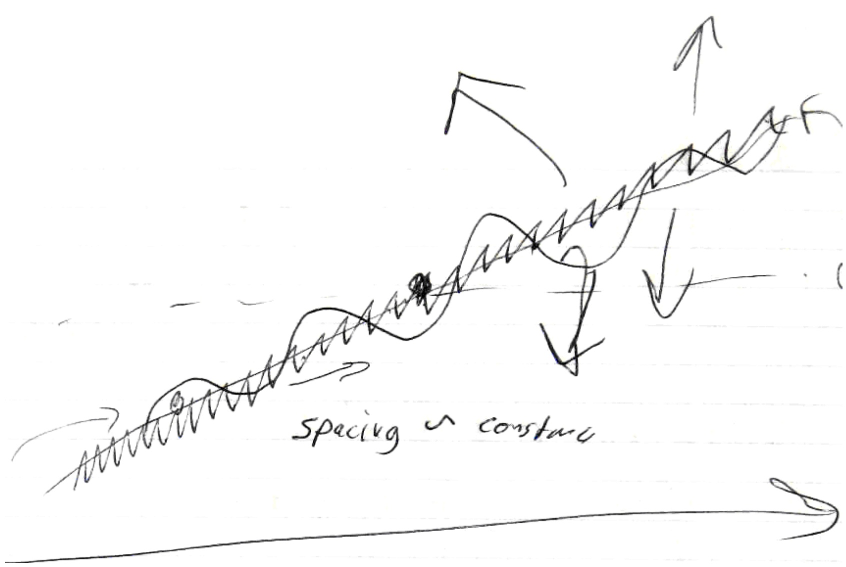


Figure 3-21: An etalon noise animation sketched by non-animator 11. Waveforms of various frequencies slide back and forth, grow and shrink, and twist in place.

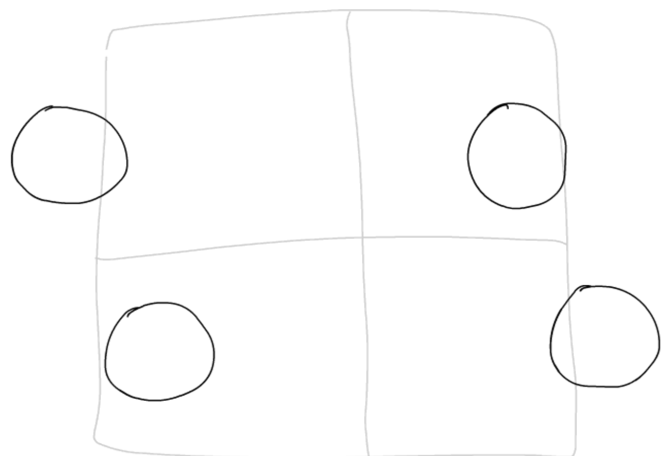


Figure 3-22: *Contra dance* animation (amateur, think) from non-animator 4, as sketched by me. The circles move around each other in complex patterns.

As a contra dance *caller*, his role was do direct a room full of dancers, and he wanted to get a sense of what his dance sequence plans would look like before he called them.

My non-animators showed that they had good reasons for creating animations, but they lacked a fast, simple, and expressive tool that would allow them to do so. Two (Non-animators 3 and 4) knew of domain-specific animation tools that partially fit their needs, but the output of these tools could not be customized. A single tool capable of expressing a variety of animations would have fit their needs better. Some non-animators had heard of Flash and some knew that PowerPoint handled animation, but they believed these tools to be prohibitively complex. Teachers had trouble devoting lesson planning time to learning or using such tools. Non-animator 1 was concerned that her students would waste studying time learning to use animation tools. These stories are further evidence of the need for a simple animation tool that novices can learn in short order. Non-animators who were professional scientists and engineers said that the need for an animation would arise suddenly in meetings. If they had a tool that allowed them to work fast enough, they could create their animations during or just before their meetings.

Though non-animators tended to express discomfort with drawing, most drew rough sketches that helped to explain the action of their animations, as in Figures 3-16a, 3-17, 3-18a, 3-19, 3-20, and 3-21. This gives further evidence that inexperienced users will be able to sketch well enough to use an animation sketching tool. Four non-animators described animations by pointing to existing figures or animations that they wished to reproduce or alter. Only non-animator 3's animations required a level of precision that could not be produced by sketching (see Figure 3-15). This shows that while precision is sometimes necessary, it is not required in the common case.

Like the interviews with animators, these interviews convinced me that there is a need for a fast, simple, and expressive tool for novice animators. However, the variety of examples I collected hinted that accomplishing all three goals in a single tool would be a major challenge. The remainder of this chapter details my early attempts to determine the scope of this problem by collecting and categorizing the animation scenarios from my interviews.

3.3 Library of Usage Scenarios

As I looked more closely at the tasks these interviewees were suggesting, I saw a staggering variety of subject matter, levels of complexity, and usage contexts. It seemed that the complexity of current general-purpose animation tools might, in fact, be necessary. In hopes that I could identify a small set of capabilities that would still support a wide range of animations, I began to gather the task scenarios into a library for deeper analysis.

3.3.1 Gathering Scenarios

Each scenario contains a description of the objects and actions in a single animation. It also includes a detailed description of the user who wanted to create the animation and the goal this user was trying to achieve. I decided not to include scenarios that were nearly identical to others in the library. Consequently, the library captures the breadth of tasks that users envision.[1]

I collected 16 scenarios during my interviews with non-animators. From my interviews with animators, I found another 27 scenarios that were described in enough detail for me to reproduce them. Animator 3 also gave me 22 animations

[1] This preference for breadth is consistent with my goal of maximizing expressivity in K-Sketch. I made no attempt to balance tasks by the number of people who want to perform them or the number of times a person has to perform them. Such a balance would be useful only if my goal were to support a particular sub-community more effectively.

produced by children in her classes. These were rough, 2D animations that show what a novice might create when exploring this medium for the first time. Finally, I supplemented the library with seven others from myself and other researchers. I discovered these supplementary scenarios outside of my interviews, but each captured an interesting situation in which a novice might be inspired to animate. One animation explained an actual automobile accident (see Figure 3-23), another explained gear reduction (see Figure 3-24), and a third entertained guests at a party (see Figure 3-25). In total, the library contained 72 scenarios.

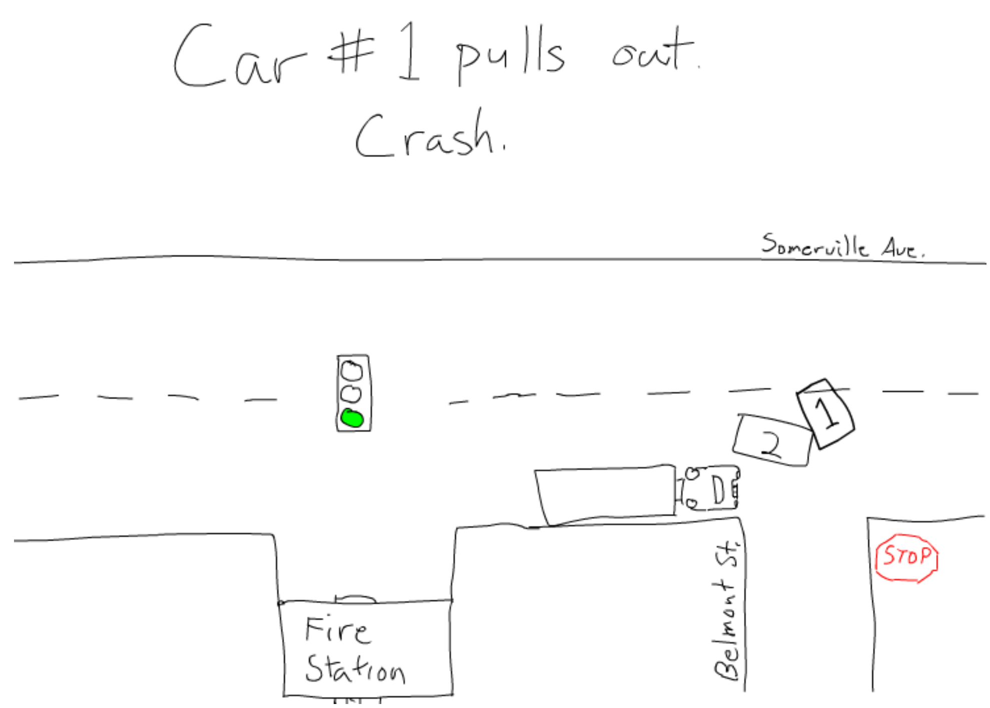


Figure 3-23: *Automobile accident* animation (amateur, explain). I was required to draw a picture of an accident scene in a police report. This animated diagram was able to communicate the circumstances of the accident more clearly.

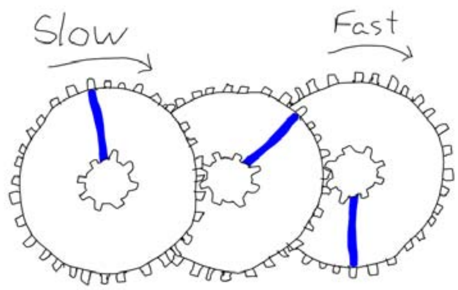


Figure 3-24: *Gear reduction* animation (amateur, explain). This diagram could help explain gear reduction to a child designing mechanical systems with Lego.

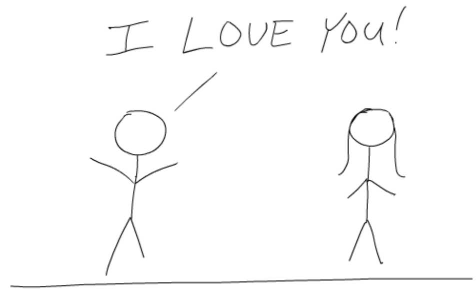


Figure 3-25: *Bachelor party* animation (amateur, entertain). This animation told the story of a friend's courtship and was shown to guests at a party one night only.

Eleven scenarios needed clarification, and I gathered more details by producing these animations in Flash. The images in Figure 3-12, Figure 3-13, Figure 3-14, Figure 3-16 b, Figure 3-18 b, Figure 3-22, Figure 3-23, Figure 3-24, and Figure 3-25 are all snapshots from these animations. Most of these animations came from non-animators, and I checked with them all to make sure that my animations accurately reflected their mental images. Producing these animations also game me my first impression of how fast novice tasks could be accomplished with current tools. Counting production time only (not tool learning time or task planning time), most animations took around an hour to produce, and some took two or even six hours to produce. It was clear that my non-animator interviewees were not merely imagining the difficulty they would encounter with current tools. Table 6-1 on page 192 lists the time I needed to build these animations in Flash alongside a comparable time for creating them in K-Sketch.

Compiling a scenario library of this kind benefits a user interface design project by making users' goals concrete. Because of the size and variety in this particular library, it was necessary to look for patterns in these scenarios. After a deeper analysis, I was able to categorize each scenario by the type of user performing it and their goals. I was also able to extract an abstract set of operations that users could perform to represent the action in each animation. The following sections explain these patterns in detail.

3.3.2 User Categories

All users fell into one of five categories. These categories are summarized in Table 3-3 and described here. Each title is followed by the number (and percentage) of scenarios with a user in this category.

- **Artists: 24 (33%).** These users are undertaking large creative tasks that involve animation. They would use an animation sketching tool to try out new ideas quickly (Figure 3-5), to prepare animatics (Figure 3-9), and to share these sketches and storyboards with clients.
- **Teachers: 12 (17%).** These users are working to impart some knowledge to their students. They want to use animation to explain some dynamic

concept (Figure 3-13), or to make course material more attractive and engaging (Figure 3-17).

- **Students: 1 (1%).** These users are taking a class from a teacher who has asked them to create an animation as a learning exercise. This is a growing class of users that I examine in the case study described in Section 8.2.2 on page 303. The sole example in my library, however, comes from non-animator 1.
- **Professionals: 5 (7%).** These are knowledge workers such as scientists or engineers who are working on a variety of complex tasks. These users may use an animation sketching tool to explain some concept to a colleague (Figure 3-18), think through a problem, or prototype a more formal presentation (Figure 3-2).
- **Amateurs: 30 (42%).** This includes any other creative amateurs undertaking small creative tasks in which animation plays a part. Amateur animators would use an animation sketching tool to tell short stories, possibly at social gatherings (Figure 3-25) or over the web. Other amateurs might use these tools to work out a problem or share an idea (Figure 3-22).

Category	Scenarios	Percentage
Users		
artists	24	33%
teachers	12	17%
students	1	1%
professionals	5	7%
amateurs	30	42%
Goals		
entertain	15	21%
explain	15	21%
think	2	3%
prototype	25	35%
doodle	15	21%

Table 3-3: User categories and goal categories extracted from the library of scenarios.

Note that, of the above categories, all but artists may have limited time to learn about animation tools. This is why simplicity is so important in an animation sketching tool.

3.3.3 Goal Categories

When undertaking the tasks in each scenario, users had one of five goals in mind. As before, these goals are summarized in Table 3-3 and described in the following paragraphs. Each title is followed by the number and percentage of scenarios with this goal.

- **Entertain: 15 (21%).** In most of these scenarios, an amateur is telling a story with animation. The four longest animations in the library tell stories—for example, the movie in Figure 3-25 is three minutes long. The other animations in this category come from teachers adding life to their materials (Figure 3-17).

- **Explain: 15 (21%).** In these scenarios, an amateur, professionals, or teacher explains some concept to another.
- **Think: 2 (3%).** In these scenarios, a novice is attempting to understand something by creating an animated visualization. The two examples of this in the library are for an amateur (Figure 3-22) and a student.
- **Prototype: 25 (35%).** This is a large segment of the library with scenarios primarily from artists, plus two from a professional (e.g., Figure 3-3). This goal can be thought of as a special case that combines *thinking* and *explaining*.
- **Doodle: 15 (21%).** This goal can be similar to *entertaining*, but the entertainment is primarily personal. This goal can also be similar to *thinking*, but in this case visualizations are more ephemeral and exploratory. The doodles in this library are similar in content to animations in the *entertain* category, but they are shorter (all 16 seconds or less) and contain only one scene.

Note that when pursuing most of the goals listed in this section (possibly excepting entertain), it is helpful to work quickly. This helps to explain why and when speed is important in an animation sketching tool.

The percentages for each user category and goal category reveal the makeup of the scenario library, but they may not be equivalent to similar numbers drawn from actual use. What is most important to observe is the breadth of users and goals that could be served by a single, well designed tool.

3.3.4 Animation Operations

The categories just presented helped me to reason about the background of these scenarios, but they say nothing about the content of each animation. My first attempt to categorize the motions in each animation produced categories that reflected the structure of each animation. For example, objects can appear and disappear. They can move by translating, rotating, scaling, or some combination of the three. They may remain static, they may move at a constant speed, or they may accelerate. They may move in one reference frame or in a hierarchy of reference frames. These categories describe the motions in each animation, but to design an animation tool for novices, I needed categories that match novices' intuitions. For a novice, figuring out how to categorize a motion is not important. The important thing is figuring out *how to express the right motion*. For this reason, my categorization evolved into a set of abstract animation operations that a novice could perform to express the motion in each animation.

These animation operations are not tied to a specific interface, but they do reflect the mental steps performed by animators using existing techniques. For example, animators can make a ball bounce by drawing each frame by hand, as in traditional cel animation, or they can speed up this process with higher-level operations like interpolation or physical simulation. Each operation has it own advantages and limitations, and a set of operations together determines how an animator can express a particular animation. Therefore, each animation interface can be characterized by the operations it provides, and each animation can be characterized by the operations that can be used to create it.

After a detailed analysis of the 72 usage scenarios in my library, I defined the following 18 animation operations:

- **Translate, rotate,** and **scale:** Move an object in one of these common ways.
- **Set timing:** Specify the timing of a motion by demonstrating it, rather than moving it at constant speed.
- **Move relative:** Add a motion on top of another, so that the new motion is relative to the old motion's reference frame.
- **Appear** and **disappear:** Make an object come into view or go out of view.
- **Trace:** Make a line appear over time, as if traced by a pen.
- **Repeat motion:** Repeat an event sequence, perhaps indefinitely.
- **Copy motion:** Move an object in a way that is the same or similar to another object's motion.

- **Define cels:** Create alternate appearances for an object, as in traditional cel animation.
- **Morph:** Turn one object into another over time.
- **Physically simulate:** Set up a simulation that moves objects as they might move in the real world.
- **Interpolate:** Define the start and end states of a change and animate the transition between the two.
- **Move forward/back:** Change the stacking order of objects, so objects that were covered up are now uncovered.
- **Deform:** Stretch an object out of its current state.
- **Move limb:** Define a skeleton for an object, and move a segment of it.
- **Orient to path:** Translate an object while pointing it in the direction it is moving.

Mapping the above list of abstract operations to operations in an actual interface may require using several operations together. For example, a frame-by-frame animation system uses *appear* and *disappear* together. A key framed translation will require the use of *translate* and *interpolate* together. A demonstrated translation will be a combination of *translate* and *set timing*.

I developed this list over three lengthy iterations of examining each scenario to extract a concise set of patterns. It was not immediately clear what operations should be included in or excluded from consideration. I chose to include any operation that I could imagine or that I had seen in an existing animation tool. Some operations are common to graphical tools in general, and I included only those that were important for representing a motion accurately. The granularity to use for operations was not obvious, either. At first, I made fine grain distinctions, and the list grew to over a hundred operations. In later iterations, I combined smaller operations into larger ones that worked together. However, I was careful to keep operations separate when I saw a possible reason to include one without the other or when I saw other tools that did so. For example, RaceSketch [113, 114] includes *appear* but not *disappear*, so I kept them separate rather than combining them as I did other operations such as *move forward/back*.

In addition to the 18 animation operations defined here, I defined five variants of *translate*, *rotate*, *scale*, and *set timing,* each with certain limitations (e.g., translation only along straight paths, rotation about an unspecified center, or timing with acceleration but not continuous demonstration). These variants enabled me to compare my designs to other animation tools, but they added nothing else to my analysis, so I do not discuss them in what follows. I also defined eight other operations that would make fairly obvious and independent improvements to most animation interfaces: *Repeat Playback*, *Add Scene*, *Play Sound*, *Occlude*, *Zoom*, *Copy Object*, *Import*, and *Define Background*. These also added little to my analysis, and I simply assume that all would be present in an animation sketching system. An abbreviated version of my analysis that includes the entire set of 31 operations appears in Appendix C.

This left me with 18 animation operations to choose from in my designs. The length of this list helps to explain why general-purpose animation tools are so complex, and this posed a significant obstacle to designing a simple tool. I suspected that I should not support all operations, but I lacked a method for choosing between them. This led me to develop a new user interface analysis and optimization technique, which I describe in the following chapter.

4 Interface Optimization

My primary goal has been for K-Sketch to be like animation paper: the first thing a novice will reach for when trying to express a moving visual image. This has produced three sub-goals; speed, simplicity, and expressivity; but there is a tension between these goals. Most tools push far in one direction to support a particular set of users or a particular set of tasks instead of balancing all needs to support a broad class of novices. Tools for professional animators optimize for speed and expressivity, but they sacrifice simplicity. Specialized animation tools and tools for children often optimize for simplicity or speed over expressivity. Pursuing all three is unusual, but it is important for a paper-like interface [158]. If K-Sketch takes too long to learn, is useful in too few situations, requires too much concentration, or does not allow tasks to be accomplished quickly, then it cannot approach the fluid interaction of pen and paper.

Finding the right balance between speed, simplicity, and expressivity for K-Sketch was a significant challenge. The library of usage scenarios described in Section 3.3 was an important first step, because it explicitly defined what tasks *should* be supported. Defining *how* to support these tasks was an important second step, and I did this by defining a set of abstract animation operations in Section 3.3.4. The next step was to choose a set of operations to support, but how was I to make that choice? Many of the animations in my scenario library could be represented multiple ways, and while all were not equivalent, many were acceptable. Also, there was no clear target for a reasonable number of operations to include in K-Sketch, nor was there a clear target for a number of scenarios to support. I needed a graph like the one shown in Figure 4-1 that showed me how simplicity decreased as expressivity increased, assuming users could work as fast as possible (or close to it). I hoped to find a point on the graph that gave high value for a relatively small number of operations. This point of diminishing returns is where an animation sketching tool would fit best.

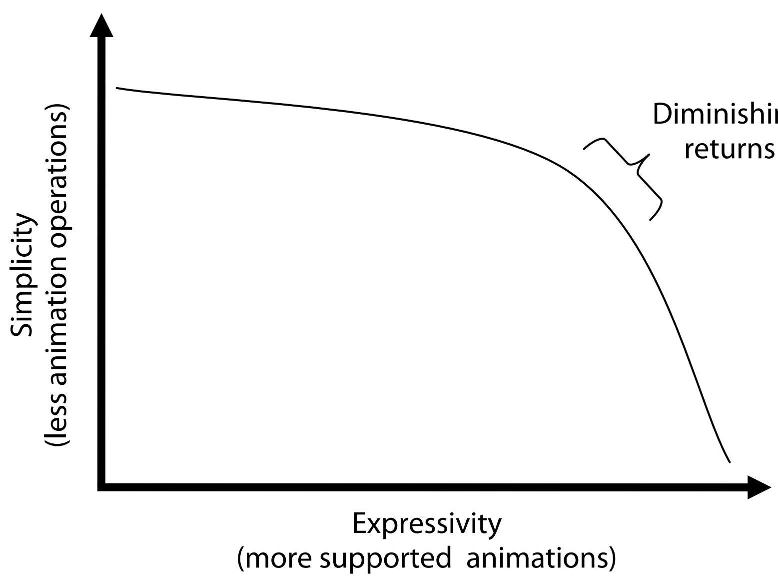


Figure 4-1: A qualitative graph showing the simplicity–expressivity tradeoff. Supporting more animations requires a tool to provide more operations, which reduces simplicity. A graph of this kind may reveal a point of diminishing returns that has the best value for operations.

I began to formulate this as an optimization problem. To make K-Sketch fast, I could minimize the number of steps a novice must perform to accomplish these tasks. To make it simple, I could minimize the number of animation operations a novice must understand to accomplish tasks. To make K-Sketch expressive, I could maximize the number of tasks a novice could accomplish. If I was to find a good balance between these variables, I needed a method for visualizing the tradeoffs associated with them. Interface optimization is the method I developed for doing this.

This chapter explores interface optimization in depth. In Section 4.1, I describe the first step in the process: coding a library of scenarios to enumerate all the possible ways of accomplishing each task. Section 4.2 describes the second step: using an optimization program to produce visualizations that help designers understand design tradeoffs. In Section 4.3, I explain how I used the output of this optimization program to design K-Sketch. I close with an overview of approaches to generalizing and improving interface optimization in Section 4.4.

4.1 Coding the Scenario Library

Before I could attempt any optimization, I needed to *code* my scenario library in a way that would reveal useful properties. I use the word "code" here as it is used in the behavioral sciences to refer to the interpretation and characterization of qualitative data [77, p 85]. Coding of spoken words, for example, can be done by *segmenting* speech into utterances and then assigning a code to each utterance corresponding to a category. This method inspired my approach to coding my library of animation scenarios. The full coded library appears in Appendix C.

My coding process began with segmenting each animation into *features* that a user would have to represent to complete the animation. These features usually corresponded to classes of objects (or parts of objects) that moved in a particular way. For example, consider the rust animation in Figure 4-2. I counted four features in this animation: the rust, the water droplet, the hole, and the text label. Note that the two clumps of rust are mirror images of each other and are counted together.

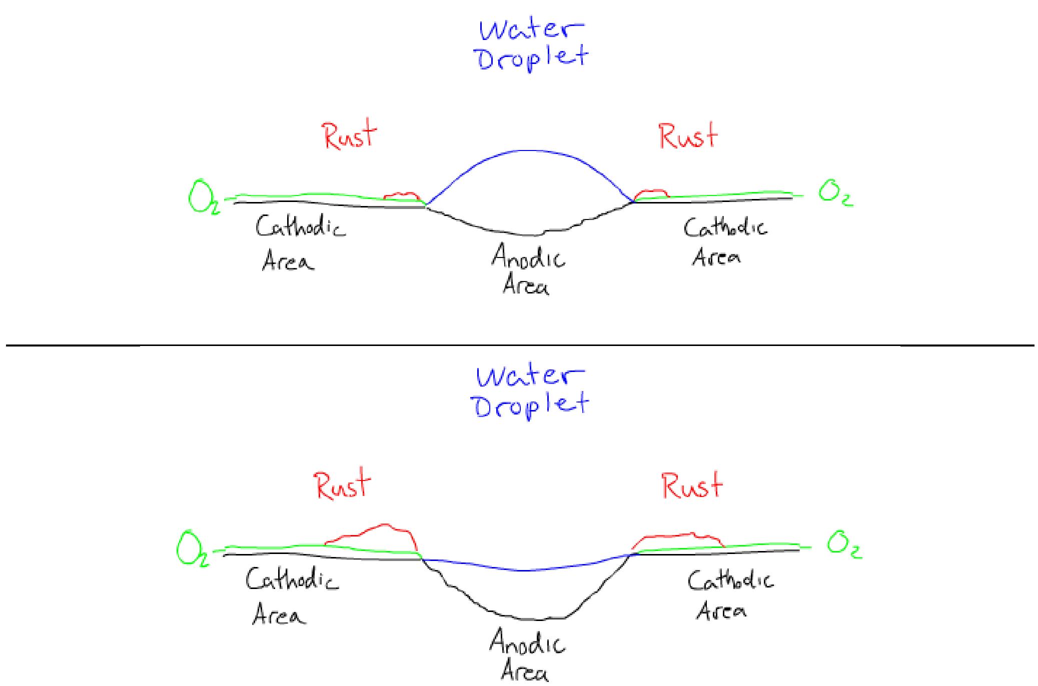


Figure 4-2: *Chemistry: rusting reaction* animation from non-animator 5, as sketched by me. This animation has four features: two growing clumps of rust, a shrinking water droplet, a growing hole (under the droplet), and the label "Anodic Area" that moves down as the hole grows.

My next step was to code each feature by listing all possible *approaches* to representing that feature with my animation operations. For each approach, I recorded the animation operations required by the approach as well as the number of conceptual steps involved. Using the rusting reaction as an example once again, I listed two approaches to animating the water droplet. One uses the *morph* animation operation and requires three steps: drawing the initial shape of the drop, drawing the final shape of the drop, and executing a morph command. In the second approach, the animator draws the initial and final states of the droplet, but then adds four intermediate frames rather than animating a smooth transition between them. This approach uses six *appear* and five *disappear* operations, a total of eleven steps, which is eight more than the first approach.

This example shows that the person coding features must exercise judgment concerning what is an acceptable approach to representing each feature. I can reasonably say that four intermediate steps would be sufficient for illustrating the in-between states of this water droplet, because the scenario specifies that the purpose of this animation is to show how water and metal are combined to produce rust. If this animation were attempting to illustrate detailed properties of the droplet's shape as it shrinks, then it might require more steps.

Still, not all approaches are equivalent. I ordered approaches by their number of conceptual steps; the one with the least steps was the *preferred* approach. (When several operations tied for the preferred approach, I gave preference to the approach that looked easiest or gave the best result.) The remaining steps I then ordered by their number of *extra steps* above the minimum given by the preferred approach. For example, the preferred approach for the water droplet described previously was the approach that uses *morph* (three steps), and the approach that uses *appear* and *disappear* (eleven steps) has eight extra steps.

The list of approaches could become quite large for some objects (the longest in my library had twelve), but there are ways to reduce this list in most cases. Many objects can be broken down into independently moving parts, and some motions have independent aspects that can be coded as separate features. The stick figure movie in Figure 3-25 on page 76, for example, had characters that both moved around the screen and changed their appearance, and I counted these as two different features. I also omitted some approaches altogether if they required more than ten extra steps.

With the coding scheme presented here, I was able to define what it means for a particular user interface to *support* a scenario (see the definition for *Supports* in Figure 4-3). I defined a user interface by the set of animation operations that it provided. An interface supported a scenario if it supported all of that scenario's features. An interface supported a feature if it supported at least one approach to representing that feature. An interface supported an approach if the approach's set of operations was a subset of the interface's set of operations. This was a formal definition that matches our natural understanding of the word "support" when we say that an interface supports a scenario.

```
Supports(I,s):
        ∀f ∈ Features(s), ∃P ∈ Approaches(f) s.t. P ⊆ I

Count-Supported(I,S):
n ← 0
for each s in S
        do    if Supports(I,s)
                    then n ← n + 1
return n
```

Figure 4-3: Pseudocode that formally defines how support for scenarios is determined. ***Supports*** is a predicate that defines how a user interface I can support a scenario s, and ***Count-Supported*** simply returns the number of supported scenarios. A user interface I is encoded as a subset of A, which is the set of all animation operations. S is the set of all scenarios.

```
Compute-Interfaces-Idealized(A,S):
for k ← 1 to |S|
        do    B ← I ∈ 𝒫(A) s.t. Count-Supported(I,S) ≥ k
              L[k] ← argmin_{I∈B} |I|
return L
```

Figure 4-4: An idealized interface optimization procedure that computes the simplest interfaces supporting all size subsets of the scenario library. A is the set of all animation operations, and S is the set of all scenarios. Upon completion, $L[k]$ contains the simplest interfaces that support k or more scenarios. Note that "$\mathcal{P}(A)$" is the power set of A, and "$\text{argmin}_{I \in B}\ |I|$" returns all interfaces I contained in B that minimize $|I|$.

4.2 Optimization Procedure

The coded library contains rich information about the speed, simplicity, and expressivity tradeoffs in different user interfaces, but I needed a tool that could reveal these tradeoffs. My goal was to produce visualizations similar to the one in Figure 4-1 that would allow me to weigh these tradeoffs. This meant that I needed to compute a range of solutions that found the simplest interfaces (i.e., the smallest sets of animation operations) that would support every size subset of my scenario library from one to 72. Figure 4-4 shows an idealized version of this optimization procedure. (To visualize interface speed, I also needed to compute solutions that allowed a varying length of time to produce each animation, and I will address this shortly.)

I considered reducing this problem to a series of better-known optimization problems or formulating it as a dynamic programming problem, but fortunately, a simple solution presented itself. The number of animation operations and the size of my coded scenario library were so small that the simplest approach of exhaustively searching through all possibilities yielded results in a reasonable time. The pseudocode in Figure 4-5 gives a detailed picture of how my optimization program works. The system iterates through all possible user interfaces testing each one against the coded library to see how many scenarios the interface supports. A table records the simplest interfaces (i.e., the ones with the least animation operations) that support each size subset of the scenario library, from one to 72. This table is updated

```
Compute-Interfaces-Actual(A,S):
1  for k ← 1 to |S|
2        do   N[k] ← |A|
3             L[k] ← {A}
4  for each I in 𝒫(A)
5        do   k ← Count-Supported(I,S)
6             while k ≥ 1 ∧ |I| ≤ N[k]
7                   do   if |I| < N[k]
8                             then N[k] ← |I|
9                                  L[k] ← ∅
10                       L[k] ← L[k] ∪ {I}
11                       k ← k - 1
12 return (N, L)
```

Figure 4-5: The actual interface optimization procedure. Lines 1–3 initialize the table (two arrays) that records the simplest interfaces supporting *k*-size subsets of the scenario library: *N*[*k*] holds the number of operations in the interfaces, and *L*[*k*] holds the interfaces themselves. Line 4 iterates through all possible interfaces. Line 5 tests each interface against the library. Lines 6–11 update the arrays *N* and *L*.

after each test that discovers an interface that is smaller than any previous interface handling the same number or fewer scenarios. Upon completion, the table holds a series of globally optimal results, just as required for visualizations like Figure 4-1.

Unfortunately, this algorithm has exponential complexity. With 18 animation operations, the system must search through 2^{18} or 262,144 interface possibilities[2]. My

---

[2] When I also included the other 13 operations mentioned in Section 3.3.4, this number became 2^{31} or over two billion interface possibilities. The system still generated results in under three hours when I ran it on a network of workstations, as explained in Appendix C.

implementation of this program was a 766 line Python script that I ran on a 2.8 GHz Intel Xeon CPU with 2 GB RAM running Fedora Linux release 7. This was able to compute solutions in about 18 minutes.

Each solution can be used to produce graphs similar to Figure 4-1 showing the tradeoff between simplicity and expressivity in possible interfaces. To explore speed tradeoffs as well, I computed multiple solutions. Each time I ran the program, I limited the approaches that would be considered when testing whether or not an interface supported a scenario. The first run allowed only the fastest solutions, and successive runs allowed increasing numbers of extra steps, which produced slower but simpler interfaces. Thus, with successive runs, it become possible to visualize a three dimensional space of speed, expressivity, and simplicity tradeoffs.

4.3 Interpreting Optimization Results

I ran the preceding optimization program five times to produce the data summarized in Figure 4-6. Each point on the graph shows the smallest number of animation operations necessary to support a certain number of scenarios, according to one run of the program. The five lines show the results produced when I allowed between zero and four extra steps. When no extra steps are allowed, this line is relatively flat, but if users are required to perform a few extra steps, then somewhat slower interfaces are allowed, and a point of diminishing returns begins to emerge. I could see that supporting more than 70% to 80% of the scenarios in my library might not be worth

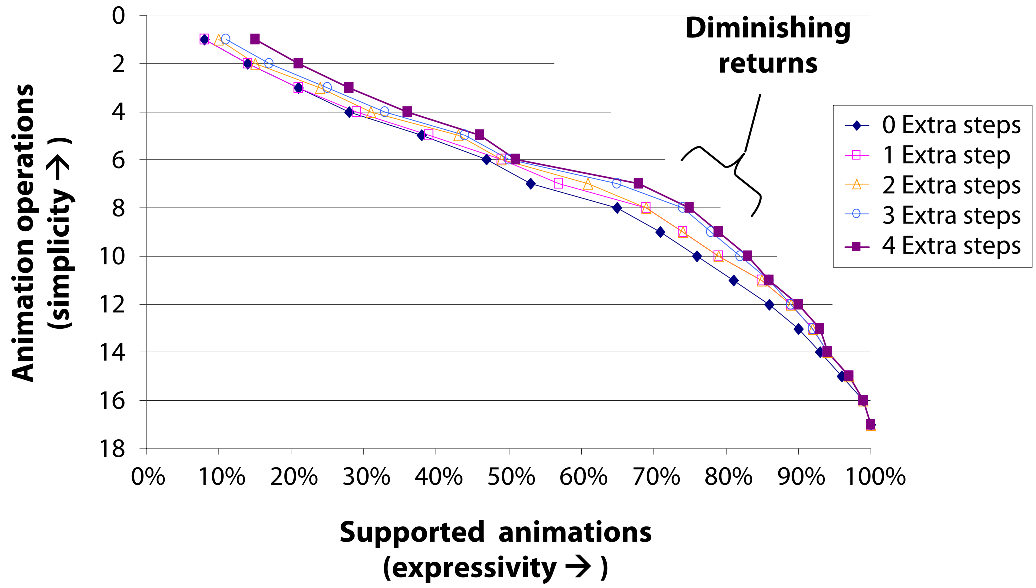


Figure 4-6: Graph showing aggregated data from five runs of the optimization program. The vertical axis represents simplicity, the horizontal axis represents expressivity, and the lines themselves represent speed. An area of diminishing returns is highlighted.

the effort, as it would require almost twice as many operations to support the last 20–30% of the scenarios. I made it my goal to support 80% of the scenarios.

After making that decision, it was necessary to consider the relative importance of each operation. Table 4-1 shows a detailed subset of the data that makes this relative importance more clear. (This data and the remainder of my analysis assume that four extra steps are allowed, because four seemed like the maximum that users would tolerate. Appendix C shows similar tables for 0–3 extra steps.) Operations in Table 4-1 are sorted from top to bottom in decreasing order of importance. There are many solutions in this table, but the visualization shows clear trends. The operations toward the top of the table are used more often than the ones toward the bottom. The table also shows that K-Sketch could support 68% of the scenarios with only seven operations (*Disappear* and above). After that, the relative importance of operations becomes harder to judge.

Many columns of Table 4-1 contain multiple interfaces as solutions. The second row of each column shows the number of animation operations in that column's interfaces, and the remaining rows reveal the operations in those interfaces. Black cells correspond to animation operations that are in *all* of a column's interfaces, and white cells correspond to animation operations that are in *none* of a column's interfaces. Grey cells are for operations that appear in some, but not all of a column's interfaces. The presence of multiple solutions is another reminder that interface optimization aids a designer's judgment but does not replace it.

Scenarios Supported	15%	21%	28%	36%	46%	51%	68%	75%	79%	83%	86%	90%	93%	94%	97%	99%	100%
Num. of Operations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Translate	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Scale		▒	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Rotate		▒		■	■	■	■	■	■	■	■	■	■	■	■	■	■
Set Timing			■		■	■	■	■	■	■	■	■	■	■	■	■	■
Move Relative				■	■		■	■	■	■	■	■	■	■	■	■	■
Appear						■	■	■	■	■	■	■	■	■	■	■	■
Disappear						■	■	■	■	■	■	■	■	■	■	■	■
Trace									▒	■	■	■	■	■	■	■	■
Repeat Motion								■	▒	▒	▒	■	■	■	■	■	■
Copy Motion									▒	▒	▒	■	■	■	■	■	■
Define Cels									▒	▒	▒	■	■	■	■	■	■
Morph										▒	▒	■	■	▒	■	■	■
Physically Simulate		▒									▒		▒	▒	▒	▒	▒
Interpolate													▒	▒	▒	▒	■
Move Forward/Back											▒			▒	■	■	■
Deform										▒	▒			▒	▒	▒	▒
Move Limb		▒												▒	▒	▒	■
Orient To Path														▒		▒	■

Table 4-1: A more detailed subset of interface optimization data from a run with four extra steps. Each column shows the simplest interfaces (smallest sets of animation operations) that would support the percentage of scenarios shown at the top. If an operation is needed in all interfaces, a black box appears in its cell. Gray boxes show those operations that are in some, but not all interfaces. The column representing 79% of the scenarios was the closest to my target.

Table 4-1 gave me the information I needed to make my design decisions. The closest column to my target showed interfaces supporting 79% of the scenario library. The seven operations with black cells (*translate, scale, rotate, set timing, move relative, appear,* and *disappear*) were clearly important, so I made these a priority in my designs for K-Sketch. I then needed to choose two of the following: *trace, repeat motion, copy motion,* and *define cels.* According to the data, I had three choices: *trace* and *copy motion, trace* and *repeat motion*, or *repeat motion* and *define cels.* I chose *trace* and *copy motion*, because they fit most naturally into the designs I created for the other seven operations.

These decisions are data-driven, but they are not directed solely by the data. Interface optimization produces results that are only as trustworthy as the data fed into it, and a designer's gut feeling may conflict with these results. For example, I decided to add the *orient to path* operation to K-Sketch, because it fit so well with my designs that it seemed to add negligible complexity. (A few study participants had trouble distinguishing the *orient to path* region from the *translate* region, however, so I may have underestimated this complexity. Still, *orient to path* delights many users, and I have not removed it from the interface.)

After adding *orient to path* to my list of animation operations, the number of scenarios that K-Sketch would support increased to 81%. As an evaluation of my progress in designing K-Sketch, I compared my choice of animation operations to the choices made by other animation tool designers. Doing so required me to extend my optimization program to perform a different computation. The primary analysis computed the smallest sets of operations that support a given number of scenarios, but the new analysis computed the reverse: the number of scenarios supported by a given set of operations. If I define an animation tool by the set of operations it provides, then this reverse analysis allows me to compare the number of scenarios supported by each tool. (Note that this analysis was only possible because I designed the animation operations to capture distinguishing features of animation tools. For example, *appear* and *disappear* were not combined into a single *appear/disappear* operation because some animation tools included one operation without the other.)

Table 4-2 shows the results of comparing K-Sketch in this way to several tools described in Section 2.1.3 (on page 20) and Section 2.1.4 (on page 22). The results of this comparative analysis are quite favorable for K-Sketch, but this is not surprising. Remember that an interface supports a scenario only if all of its features can be represented using no more than four extra operations. Many of the other tools in the table are commercial tools with different standards of success. These commercial tools exclude operations that produce sloppy results quickly (e.g., *set timing,* and *trace*) while often including others that produce polished results more slowly (e.g., *interpolate* or *morph*). The remaining tools in this table are research tools that explore a set of interaction techniques but do not seek to strike a balance between conflicting

Animation Tool	Number of Operations	Supported Scenarios
K-Sketch	10	81%
Commercial Tools		
Flash	13	63%
PowerPoint	10	47%
The TAB Lite	8	32%
Research Tools		
Living Ink	6	42%
Race Sketch	5	24%
KOKA	4	14%

Table 4-2: Speed and simplicity of various tools, as measured by number of operations and supported scenarios.

needs as K-Sketch does. In this case, the table shows how expressive each particular set of interaction techniques is on its own.

I have shown that interface optimization helps designers to identify fast, simple, and expressive interfaces and to compare existing interfaces, but there are two side benefits worth mentioning. First, interface optimization can help to explain *why* certain operations are important or not important. I provide two examples of this here:

- After conducting interface optimization for K-Sketch, I was surprised to discover that *move relative* was so much more important than *move limb.* This caused me to review those scenarios that required *move relative* and improve my interface designs to better support them.

- I used the coded library to characterize those 14 scenarios that would *not* be supported by K-Sketch. The largest number of these were slightly more formal animations that needed to cycle an object through a set of appearances as it moved, requiring the *define cel* operation. Some other scenarios needed to *deform* an object so far from its original state that redrawing it many times to simulate deformation became onerous. Finally, a few animations required a level of precision that could only be produced with *interpolate.*

The second side benefit of conducting interface optimization is that it can help a design team understand the nature of a design problem. Before conducting interface optimization on K-Sketch, I did not know how to go about designing an animation interface that was simultaneously fast, simple, and expressive. The process of coding my library of scenarios helped me to understand exactly how each operation would be used. This allowed me to design an interface that streamlined operations to fit users' needs.

Because of this interface optimization process, I had a very clear picture of who would be using K-Sketch and what they would be trying to do. I would not support all scenarios, but I could optimize for those scenarios that I would support. My coded library gave me a good sense of how my interface would support these scenarios. Knowing these things put me in a much better position to design an interface that met users' needs, and it gave me some assurance that the system could attract a broad audience of novice animators.

4.4 Generalizing and Improving Interface Optimization

My experience with interface optimization shows that the technique is worth generalizing and applying to other interfaces that seek to balance speed, simplicity, and expressivity. While techniques do exist for assessing the relative importance of operations in a user interface by comparing existing, deployed implementations [88], interface optimization reveals this information at the *beginning* of the design process. If these issues are not addressed early, a design team may not discover them until multiple implementation efforts have produced poor interfaces. Not only is this expensive, but making drastic changes to an existing interface can alienate an installed user base. Also, at the early stages of design, many team members may be motivated by unsubstantiated opinions that come into conflict. Designing a user interface will always involve guess work and require designers to rely on their intuition, but interface optimization's data-driven approach can introduce a healthy dose of objectivity into design discussions.

If interface optimization is to become widely applicable, a number of improvements may be necessary. Chief among these is an algorithm that scales to larger numbers of operations. The exhaustive search that I currently perform may scale up to about 40 operations, but only if the program is run on a large network of workstations. With other researchers however, I am investigating techniques that quickly generate approximate solutions without guaranteeing optimality in all cases. This has enabled us to run interface optimization on a different set of 84 operations in only two hours, and it should scale to even larger operation sets. I will continue to investigate such techniques in the future.

Further scrutiny of this technique may also show that it is important to formally evaluate the process of segmenting and coding a library of scenarios. Similar practices have evolved in the social sciences for assessing the quality of a coding process [77]. For example, while defining a set of operations, designers could develop a coding manual that formalizes the process of breaking a scenario into coded features. One half of the scenarios could be used to define operations and develop the manual, after which the second half could be coded. If the second half can be coded easily, then the coding scheme is replicable to new scenarios. It is also helpful to have a second person code around 25% of the scenarios and check for inter-coder reliability. These techniques can provide some measure of the quality of a coded library. While I did not perform any of the evaluations mentioned here, Section 8.1.2 on page 281 describes a final evaluation that compares the predictions of my coded scenario library against actual usage data.

If formalizing the coding process shows a high number of errors in the coding process, then it may also become important to develop tools that facilitate this process.

In most cases, the set of operations will be defined while an initial set of scenarios is being coded. At this stage, operations are being split and merged into new operations, and errors may appear in previously coded scenarios but remain undetected. An interface that manages this process of splitting and merging could flag scenarios with potential errors.

The number of errors in the coding process could also be reduced with a tool that enforces consistency between coded scenarios. For example, a tool could require all steps in an approach to be listed out explicitly, and it could provide macros for reusing sequences of steps that can be applied in many situations. This would speed up the coding process, and it may also discourage the practice of cutting corners by omitting approaches, a potential cause of coding errors. Macros would also have the important side effect of documenting the coder's assumptions about how users will go about accomplishing tasks. This documentation can facilitate reviews by other team members and help designers see the common cases around which the interface should be designed. Better documentation would also make it possible to verify whether or not design-time assumptions hold true by comparing them to actual user behavior after an interface has been developed.

Finally, further development of this technique may show that better estimates are needed for the relative complexity of operations and the relative speed of approaches. Each operation could be given an index of difficulty as a weight in the optimization procedure, and this would account for some operations adding more complexity to an interface than others. If the practice of counting conceptual steps in an approach proves to be highly inaccurate, then it may be possible to annotate each step with a time estimate. It may also be useful to normalize the approaches for a feature by the number of steps (or time) in the fastest approach. The current method of counting extra steps does not normalize in any way, and consequently some approaches may appear to be far too slow when in actuality the feature is inherently complex.

The improvements mentioned here are all incremental, however. The experience of using interface optimization with K-Sketch has shown that this is a powerful method for balancing conflicting needs in a new user interface. By extracting a detailed picture of design tradeoffs in actual usage scenarios, interface optimization can remove a lot of the guess work in the design process.

5 K-Sketch Design and Implementation

Interface optimization helps a designer build a skeleton of a user interface by defining what it will do in abstract terms. Fleshing out the interface involves making many smaller, interrelated decisions. This chapter describes these smaller decisions and explains how the K-Sketch system presents animation operations to users.

An important decision I made early in the design process was to build a sketching interface. My literature review showed that sketching is used heavily in design (see Section 2.2.1), and I observed this practice in the animators I interviewed (see Section 3.1). I also saw reason to believe that sketching would help novices preserve creative flow, which is particularly important when trying to work quickly (see Section 2.2.2). Consequently, I borrowed ideas heavily from existing pen-based systems and research.

This chapter begins by explaining how each of the animation operations shown in Table 5-1 is provided by the K-Sketch interface (Section 5.1). I then tie these details together by presenting four examples of K-Sketch being used to produce animations from my scenario library (Section 5.2). I close in Section 5.3 with a brief description

Number	Animation Operation
1	appear
2	disappear
3	translate
4	scale
5	rotate
6	set timing
7	orient to path
8	move relative
9	copy motion
10	trace

Table 5-1: The ten animation operations supported by K-Sketch. This chapter explains how each of these operations is supported by K-Sketch. Operations are listed in the order that they appear in this chapter. In most of this chapter's text, operations are written with the number shown to help track progress while reading.

of K-Sketch's implementation that emphasizes important lessons I have learned about the architecture of such systems.

5.1 The K-Sketch User Interface

K-Sketch currently supports twelve animation operations. Ten of these were identified through interface optimization: 1–*appear,* 2–*disappear,* 3–*translate,* 4–*scale,* 5–*rotate,* 6–*set timing,* 7–*orient to path,* 8–*move relative,* 9–*copy motion,* and 10–*trace.* (Nine of these were found to have high value, and 7–*orient to path* was found to add some value at little cost in complexity, as I explained in Section 4.3.) Also recall that my interface optimization assumed the presence of eight other operations that I felt any commercial tool would need. Of these, K-Sketch currently supports *repeat*

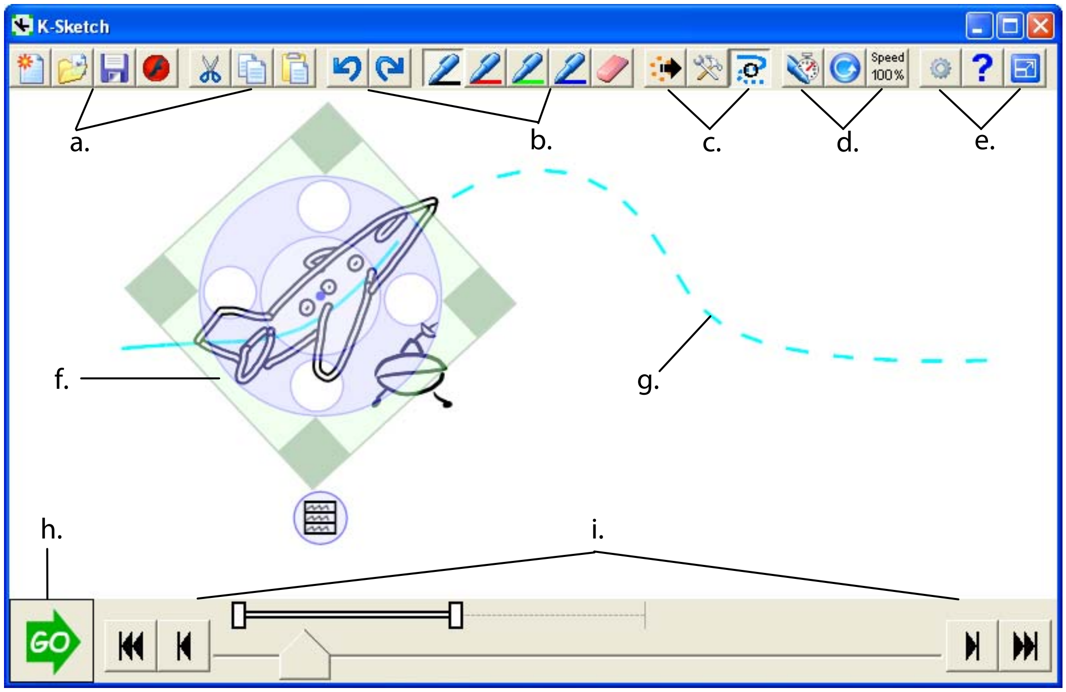


Figure 5-1: The K-Sketch user interface. *a: New, Open, Save, Export flash, Cut, Copy,* and *Paste* buttons. *b: Undo, Redo,* and draw–erase mode buttons. *c: Select next guess, Fix last operation,* and *Show motion paths* buttons. *d: Record drawing, Loop playback,* and *Adjust speed* buttons. *e: Options, Help,* and *Full screen mode* buttons. *f:* Object handle *g:* Motion path. *h: Go–Stop* button. *i:* Time slider bar.

playback and *copy object.* The others are straightforward to add and have little research value.

The K-Sketch user interface appears in Figure 5-1. It is visually divided into three parts: a tool bar on top, a drawing canvas in the center, and time controls on the bottom. This arrangement of controls is designed for notebook-sized pen-based computers (e.g., a Tablet PC), but the design could easily be converted to work with wall-sized surfaces. For example, the toolbar and time controls could float near the pen or be placed somewhere that is more accessible to users while standing.

The following subsections walk through notable features of this interface. After discussing some foundational mode switching issues, I then explain how users perform basic animation in K-Sketch through a series of editing steps. Next, I describe how objects are selected and manipulated to make rich animations quickly and easily. This is followed by an explanation of how K-Sketch users can modify existing motions and add new ones to create a hierarchy of reference frames. Then come subsections that describe how the 9–*copy motion* and 10–*trace* animation operations are supported. The final three subsections give details on K-Sketch's recording controls, speed control, and time navigation controls.

5.1.1 Mode Switching in K-Sketch

Mode switching has been the subject of a long-running debate among designers of pen-based user interfaces. Some designers prefer to change pen modes with buttons and menus in the interface (e.g., Tracking Menus [54]), while others prefer to change modes using physical buttons (e.g., Scriboli [69]). Still others prefer entirely modeless interaction, but this usually requires the intent of some pen strokes to be disambiguated after they are drawn (e.g., with a menu [134] or with additional pen gestures [162]).

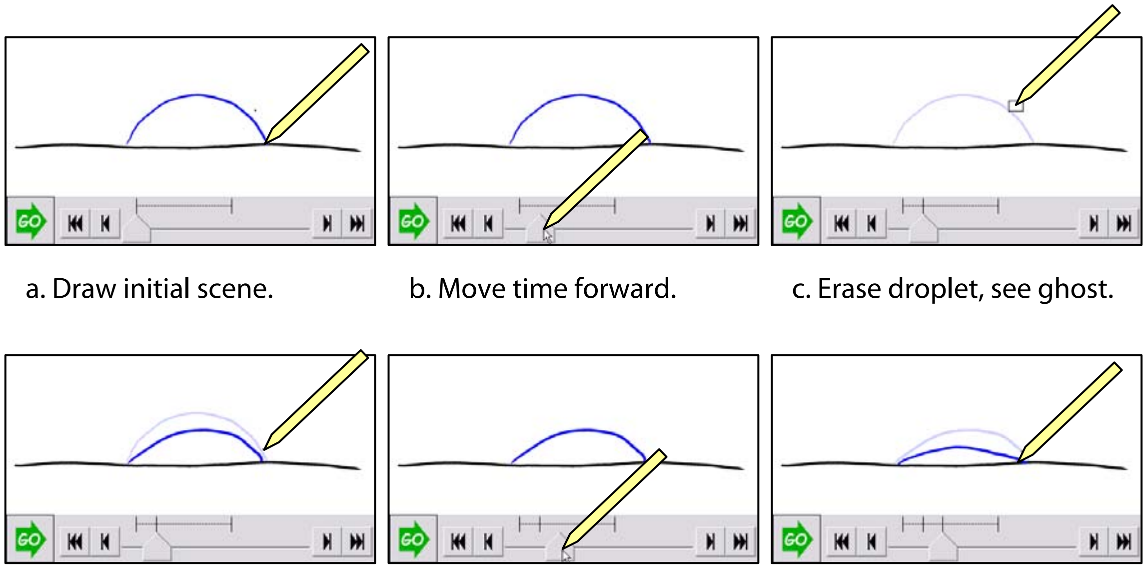
Because my approach to mode switching could affect many other design decisions, it was necessary to make some choices early in the design process. I chose to avoid modeless interaction because some interactions I envisioned required the pen mode to be unambiguous when the pen touched the screen. Also, disambiguating pen strokes after the fact can break creative flow. Mode error can also break creative flow, however, so I chose to avoid software buttons that changed modes. (There are two exceptions: The **draw–erase** mode is chosen with one of the five buttons shown in Figure 5-1b, and there is a special **Record drawings** mode as explained in Section 5.1.6.)

Most modes in K-Sketch are accessed in a way that is analogous to most direct-manipulation interfaces: Selected objects have a manipulator with multiple control zones (see Section 5.1.3). When yet another mode is needed, users access it by holding a physical mode switch with their non-dominant hand before touching the screen with the pen. Li and colleagues found this to be an effective means of switching modes in pen-based interfaces [98], and similar two-handed mechanisms exist in popular keyboard-and-mouse interfaces. This action is analogous to holding a keyboard key such as Shift, Control, or Alternate before dragging the mouse. Because of this similarity, I referred to the mode switch as the **Alternate** button in user documentation. I hoped that K-Sketch users would begin to use this Alternate button unconsciously as they used the keyboard key with the same name.

5.1.2 Basic Animation

To create an animation with K-Sketch, users perform a sequence of ordinary graphical editing steps over time. In the simplest case, this reduces to creating an animation frame by frame. Users can draw or erase pen strokes at one moment in time, move to a later time using the time slider bar (see Figure 5-1i), and draw or erase more pen strokes. Like all edit operations, draw and erase events are visible starting from the time indicated by the time slider bar and lasting until more edits occur to the same objects. This is how K-Sketch provides the 1–*appear* and 2–*disappear* animation operations.

Figure 5-2 shows the sequence of steps involved in creating part of the rust animation from Figure 4-2 on page 89. The user begins by drawing the initial appearance of the scene. To make the water droplet shrink, the user moves time forward, erases the droplet, and draws it again. To remind the user of the droplet's previous appearance, a transparent version of the droplet called a **ghost** appears. This ghost is visible only at the moment in time when an object is erased, so as not to clutter the interface. In this way, users can easily create animation frame by frame as in other tools like Sketchy [59], even though K-Sketch has no true notion of frames. Section 5.1.9 gives more detail on the time navigation controls that make this possible.



a. Draw initial scene. b. Move time forward. c. Erase droplet, see ghost.

d. Draw new droplet. e. Move forward again. f. Repeat for each frame.

Figure 5-2: Animating a shrinking water droplet frame by frame in K-Sketch. The process involves repeated steps of moving time forward, erasing, and redrawing. A transparent *ghost* appears as a reminder of the erased object's previous appearance.

5.1.3 Selecting and Manipulating Objects

All edit operations other than drawing and erasing require an object to be selected. To select pen strokes in K-Sketch, users must hold the Alternate button and draw a loop around the strokes. Normally, the pen is in the current draw–erase mode when it hovers over the canvas, but when the Alternate button is held, the pen switches to lasso mode, as shown in Figure 5-3. If 60% of a stroke lies inside the selection loop, it is selected and rendered in outline (as in Figure 5-3b). It is also possible to select individual strokes by holding the Alternate button and tapping on them. To clear the

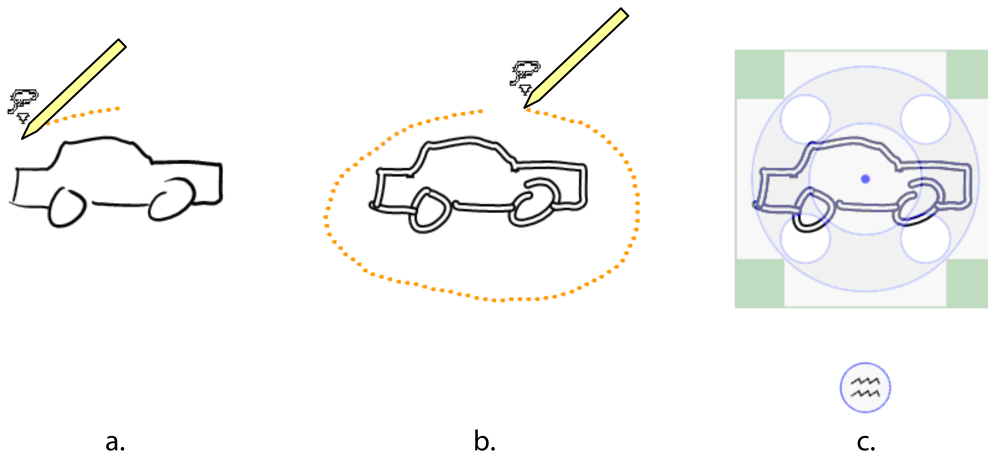


Figure 5-3: Selecting objects in K-Sketch. *a:* Hold the Alternate button and begin to draw a loop. *b:* If 60% or more of a stroke's points lie in the selection loop, it is selected and rendered in outline. *c:* When the pen is lifted, the object handle appears over strokes.

selection, users can hold Alternate and tap on empty space. Drawing or erasing also clears the selection.

When selection is finished, the **object handle** (Figure 5-3c) appears on top of the strokes. This manipulator allows a group of strokes to be moved, rotated, scaled, or stretched, depending on where the pen touches it, as shown in Figure 5-4. The handle is similar in function to the control handles in many graphical editors, but it has larger control zones for easy grabbing with the pen, and it has additional functions. Honda and colleagues proposed a similar interaction method for CAD tools called integrated manipulation [70]. Because the handle is a transparent overlay, it is similar to a tool glass [23], but the arrangement of buttons is more similar to a Tracking Menu [54]. Unlike both of these tools, however, the handle stays fixed over

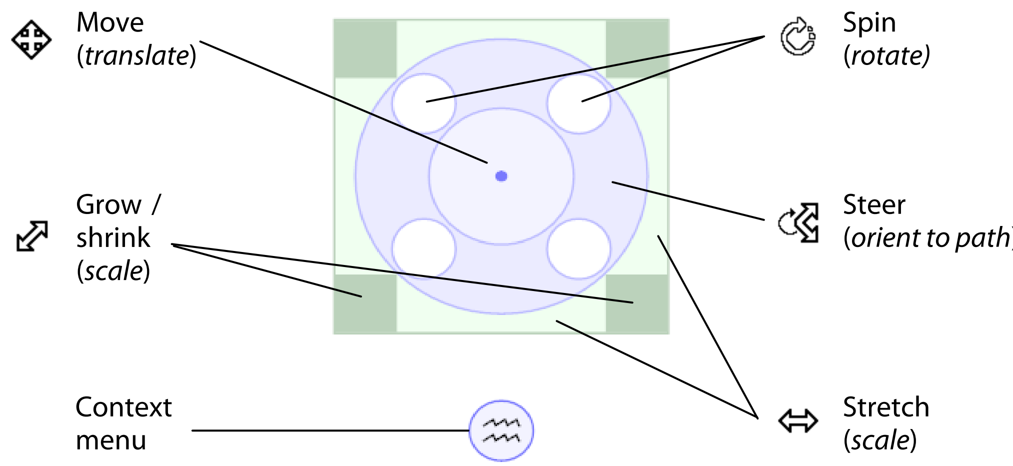


Figure 5-4: Object handle control regions. Each region is labeled with its name in the *K-Sketch User Guide* and the corresponding animation operation (in parentheses). The icons next to each label are the cursors in each region.

a group of strokes instead of being positioned by the user's non-dominant hand or floating near the pen.

Once a group of strokes has been moved with this handle, the strokes are implicitly grouped so that they can be selected more easily in the future. The *Select next guess* button is used to access these groups. If a selection loop grabs only part of a group, then clicking *Select next guess* will select the entire group. Repeatedly clicking this button will cycle through the motions associated with this group (see Section 5.1.5) and return eventually to the original selection.

By default, moving the handle inserts an instantaneous change event into the timeline as do drawing and erasing, but the handle can also be used to demonstrate motion in real time. If the user holds the Alternate button when hovering over the handle, the control regions turn red to indicate that they are *active* (see Figure 5-6c).

When the user drags an active handle region, time begins to advance from the moment the pen touches the screen until it is released. When objects are moved in this way, K-Sketch records not only their final state but all intermediate states as well. This is how K-Sketch supports the 3–*translate,* 4–*scale,* and 5–*rotate* animation operations. Because the timing of these motions is taken directly from the user's pen movement, this supports 6–*set timing* as well.

In static graphical editors, an object can be repositioned with a series of smaller edits, but a demonstrated motion must be done in one pass. For this reason, K-Sketch's handle has two important features not found in static graphical editors. First, a *steer* region combines the 3–*translate* and 5–*rotate* operations so that objects are turned to face the direction they are moving (see Figure 5-4). This is how K-Sketch supports the 7–*orient to path* animation operation (see Figure 5-6h). Second, since an object may need to be rotated about a particular point or stretched along a particular axis, the handle can be permanently re-positioned relative to an object. The handle initially appears centered on an object and axis-aligned. To move or rotate the handle relative to the object, users first press the context menu button below it. Selecting *Move handle* or *Rotate handle* in the context menu causes the handle to turn grey for a single drag during which it can be moved to reposition the pivot point or rotated to alter the stretching axes (see Figure 5-5). Moving and rotating the handle also allow users to grab it in a more convenient place, which may be important for some objects.

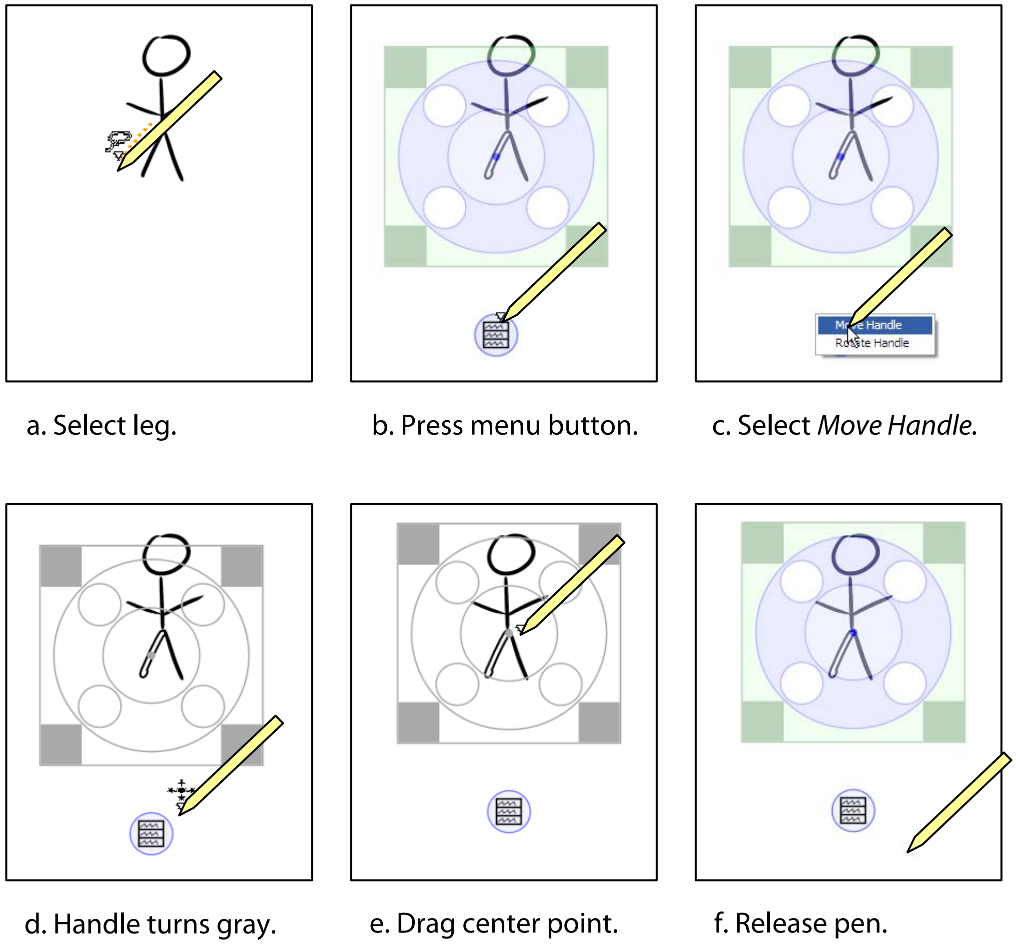
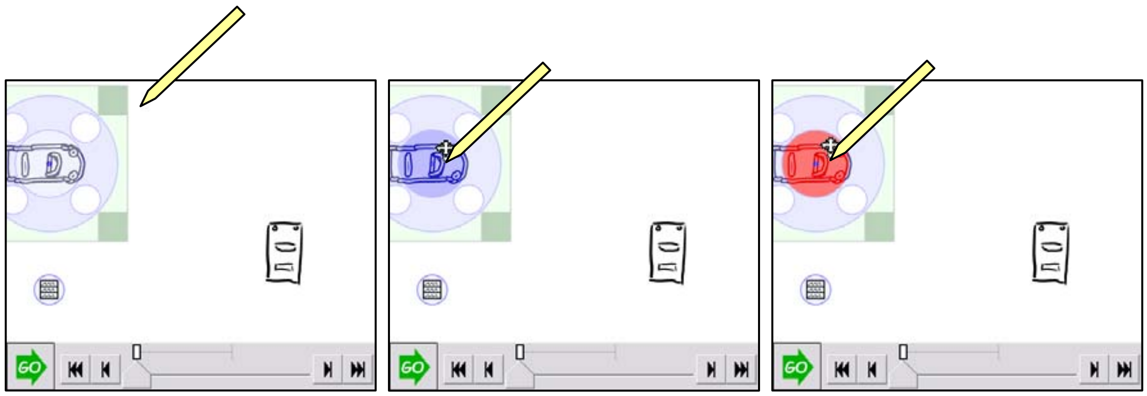


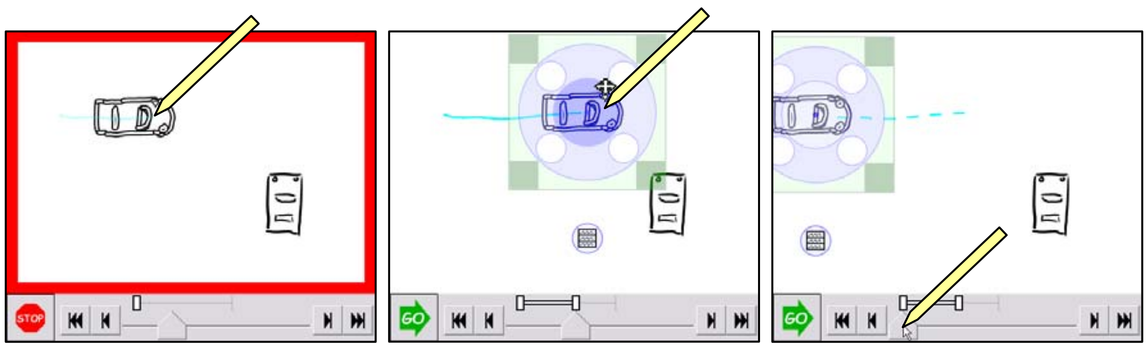
Figure 5-5: Moving a handle to set a pivot point for a character's leg. Selecting *Move Handle* in the context menu allows a single drag operation to move the handle relative to the selected object. Once it has been moved, future rotate operations will pivot about the new center.

Figure 5-6 shows how this manipulator can be used to animate part of the car crash animation from Figure 3-23 on page 75. After selecting the car (Figure 5-6a), the user positions the pen over the *translate* region (b). Holding the pen over a region darkens it and changes to cursor to remind the user of this region's function (see Figure 5-4 for the full set of cursors). Holding the Alternate button turns the region red, showing that it is active (Figure 5-6c). When the user begins to drag the car, time begins to advance (note that the *Go* button has changed to *Stop*), and the motion is recorded (d). When recording, the handle disappears, a red border is drawn around the screen, and a light blue trace begins to appear. This blue **motion path** becomes darker when the motion is finished (e). To add the motion of the second car, the user rolls time back to the beginning of the animation (f) and selects the car (g). The user then grabs the car in the *orient to path* region and demonstrates its motion, coordinating the collision by hand (h).

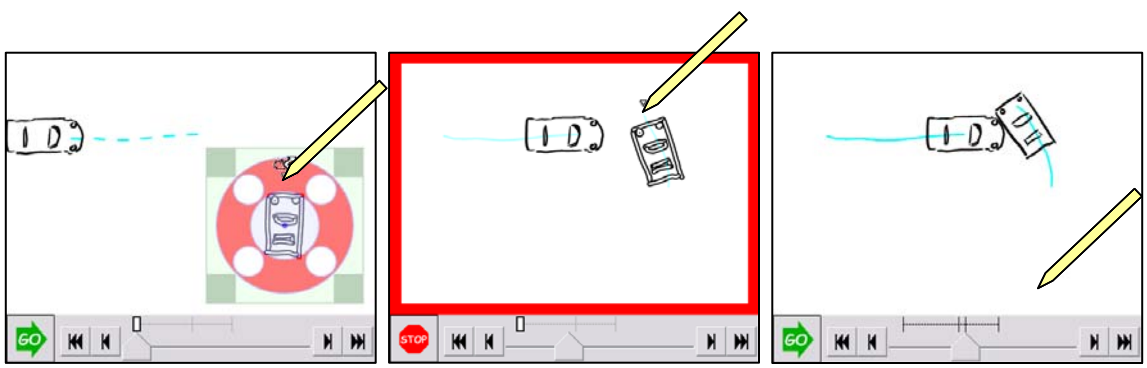
Note that all objects move simultaneously when time advances. Users rely on their intuitive sense of timing to coordinate the movement of objects. If objects are moving too quickly to be coordinated by hand, the user can slow down the animation using a global speed control (see Section 5.1.8). Even with slow moving objects, however, it can be difficult to perfectly demonstrate a collision like the one in Figure 5-6i. This is one reason why K-Sketch makes motion paths visible to the user.



a. Select left car. b. Hover over *move* region. c. Hold Alternate button.



d. Demonstrate motion. e. Release pen to finish. f. Roll back to beginning.



g. Ready to drag right car. h. Use *orient to path*. i. Final collision animation.

Figure 5-6: Coordinating two demonstrated motions to animate the collision of two cars.

Motion paths can help users coordinate moving objects by showing them where objects will go in the future. These paths are visible as long as the objects themselves are visible. Each path is rendered solid for that portion of the motion that has already occurred and dashed for that portion that is yet to come, helping users to remember where they are in time. Motion paths also serve as a visible representation for motions, and they can be selected and manipulated as explained in Section 5.1.5. If there are many objects in an animation with long, simultaneous movements, motion paths can clutter the interface. According to my library of scenarios, this is rare, but when it does happen, users can turn them off with a button (Figure 5-1c).

5.1.4 Overwriting and Adding Relative Motions

The easiest way to modify an object's motion in K-Sketch is to go back in time and demonstrating the motion over again. By default, new motions overwrite existing motions that affect the same group of strokes. Any existing motion that started at some point during a newly demonstrated motion is removed entirely. An existing motion that started before the new motion may be truncated, but it will not be removed entirely. Existing motions that start after the end of a new motion will not be affected.

Figure 5-7 shows how these rules balance the needs of the common case with the occasional need to selectively modify parts of the timeline. Overwriting an entire motion is easy because the time slider bar snaps to events (see Section 5.1.9), and truncating an earlier motion allows users to keep the first part of it and correct a later part. Later motions are kept when they do not overlap the new motion, because the

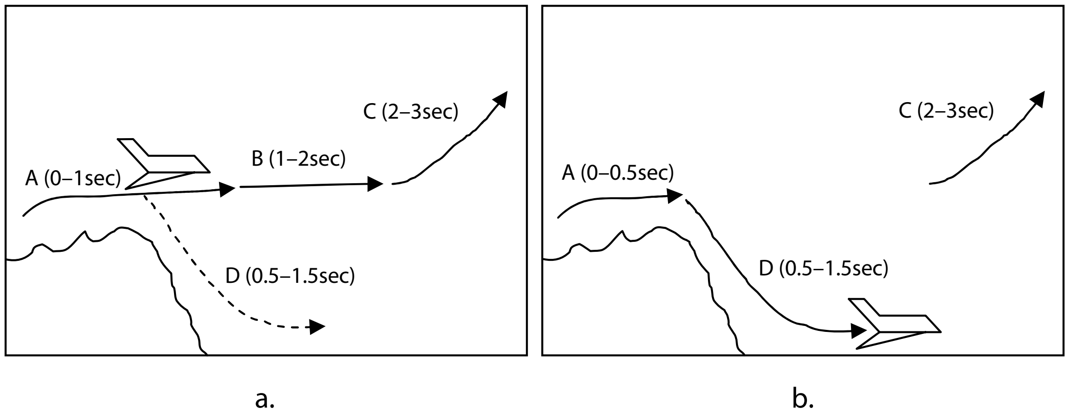


Figure 5-7: Overwrite rules for new motions. *a:* The user has an animation in which a plane makes three consecutive motions (A, B, and C) and is about to add a new motion (D) that overlaps with the first two (A and B). *b:* After motion D is added, A is truncated, B is removed, and C is left unchanged.

user may wish to modify only a small part of a chain of motions. This is a challenging edit to make, because demonstrating a new motion in exactly the same time as an existing motion is difficult. Users making a selective edit of this kind should move later motions out of the way before editing and move them back when finished (Section 5.1.9 explains how).

Users may wish to add a new motion on top of an existing one instead of overwriting it, but this brings up the question of which reference frame to choose for the new motion. When a user draws a new object, that object exists in the *world* reference frame. If the user adds a motion to that object, the motion is *relative* to the world reference frame. This motion also establishes a new reference frame, however, because K-Sketch also allows other objects to move relative to a moving object. I use the convention that the world reference frame is on top, and new ones are added below it.

Consider the rolling wheel animation in Figure 5-8. Translating the wheel down the hill (a) creates a motion relative to the world reference frame (i.e., it establishes a new reference frame below the world). Say that a user wishes to spin the wheel as it moves. She rolls back to the beginning (b) and rotates the wheel, overwriting the translation motion (c). By overwriting the motion, she has created a new motion that is relative to the world and *reused* the previous motion's reference frame instead of establishing a new one. To make the wheel spin, she must somehow establish a new reference frame for rotation motion that is relative to (i.e., below) the previous reference frame for the translation (see Figure 5-9). This requires her to use the 8–*move relative* animation operation. The remainder of this section explains how K-Sketch supports relative motions.

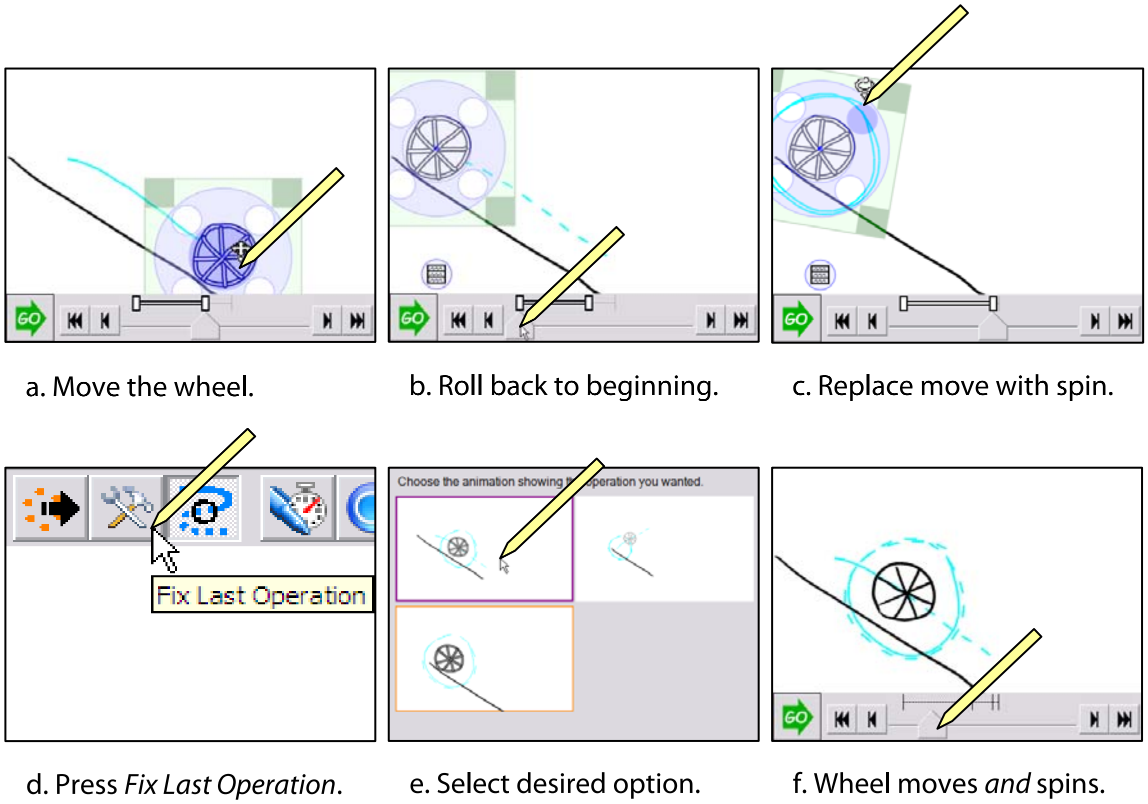


Figure 5-8: Adding a spin (rotation) motion to a moving (translating) wheel with *Fix last operation*. The original motion (*a*) is overwritten by a spin motion (*c*), but pressing the *Fix last operation* toolbar button (*d*) opens the fix dialog (*e*). The animations in this dialog show what the spin motion would have looked like in all possible reference frames, allowing the user to choose an option that puts the spin in a reference frame below the move (*f*).

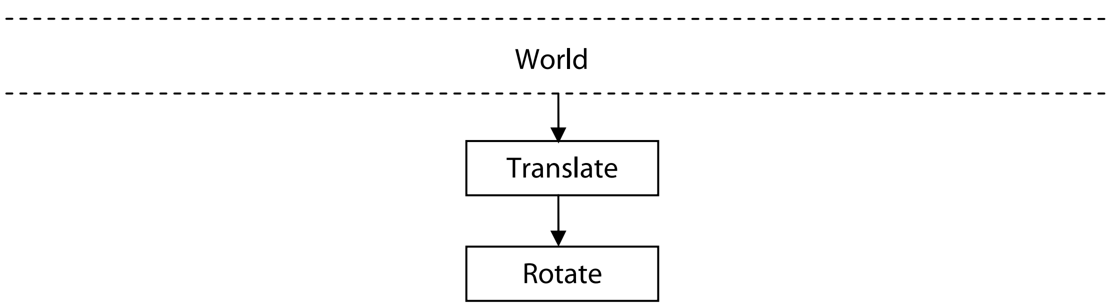


Figure 5-9: A diagram of the reference frames necessary to make a wheel roll down a hill. The world reference frame is on top, a translation motion is relative to the world, and a rotation motion is relative to the translation..

5.1.4.1 Early Support for *Move Relative*

I experimented with several approaches to the 8–*move relative* operation. My earliest attempt required reference frames to be added bottom-up. In the example of Figure 5-8, the rotation was added first, establishing one reference frame. Then the wheel *and* the motion path for the rotation were selected and translated together, adding a second reference frame above the first.

This was a fast way to add a reference frame, and it fit gracefully into the interface, but it had some serious problems. Selecting and moving a motion path in time sometimes had confusing side effects. For example, if the path of an object was moved when the animation was stopped, most users expected it to change the path at all points in time. However, such an action actually inserted an instantaneous change into the timeline, causing the objet to follow the original path at points in time before the change and the moved path at points in time after the change. Another side effect of this design for 8–*move relative* was that adding motions in a particular order required a lot of forethought, which was too difficult for novices.

I began to experiment with interfaces for explicitly navigating between a selected object's reference frames, but I eventually concluded that these were far too complex. Many novices do not have an intuitive understanding of reference frames, and learning this would be too great a burden. My solution was to *predict* each operation's reference frame and to *suggest* alternatives when my prediction was incorrect. This frees users from understanding reference frames: K-Sketch usually does the right thing, and when it does not, they need only select the right motion from a list of live animations (Figure 5-8e) to get the right motion (Figure 5-8f).

5.1.4.2 Heuristics for Choosing Reference Frames

The final design for 8–*move relative* works as follows. When users move strokes that have never been moved, K-Sketch makes this motion relative to the world reference frame, inserting a new reference frame for that motion below the world. When users move strokes that are already moving, K-Sketch uses the heuristics in Table 5-2 to determine the reference frame for a new motion. Heuristics 1 and 2 apply when *matching* reference frames are present. A matching reference frame is an existing frame with a motion that both operates on the same strokes as the new motion and is in effect at the start time of the new motion. (For brevity, I often refer to matching reference frames as *matching frames* or *matches.*)

When matching frames are present, a new motion will reuse an existing reference frame, potentially overwriting the motions in that frame. If matching frames of the same motion type are present, K-Sketch will use Heuristic 1: reuse the highest-level match of the same type. There are five motion types: *translate, scale, rotate, orient to path*, and *instant transform.* An instant transform is created when the user moves an object but does not record the motion in time (causing the object to jump suddenly when the animation is played).

#	Match type	Same motion type present?	Heuristic
1	match	yes	reuse highest-level match of same type
2	match	no	reuse highest-level match
3	sub-match	-	insert just above highest-level sub-matches
4	super-match	-	Insert just below lowest-level super-match
5	none	-	replace top level frame but preserve others

Table 5-2: Heuristics for choosing the reference frame of new motions, in order of precedence. The heuristic used depends on the properties of motions in effect at the start time of a new motion. Match type refers to the strokes affected by existing motions. A normal match affects the same strokes as a new motion, while sub-matches and super-matches affect subsets or supersets of the strokes affected by a new motion. Motion type is *translate, scale, rotate, orient to path, or instant transform.*

If no match of the same motion type is present, then K-Sketch uses Heuristic 2: reuse the highest-level matching reference frame. This heuristic applies when an object needs only one reference frame, which is the most common case. Heuristic 2 makes it easy to create a sequence of motions of different types and makes it easy to correct a motion of the wrong type. Unfortunately, this heuristic can also cause undesirable behavior, as in Figure 5-8b and c when the rotate motion overwrites the translate motion. The next section will explain how to deal with this situation.

If no matching reference frames are present, then K-Sketch uses a different set of heuristics. Two heuristics insert new reference frames while preserve all existing frames, making it easy to define complex motions quickly. Heuristic 3 applies when *sub-matches* are present. A sub-match is an existing reference frame with a motion that operates on a subset of the strokes in the new motion (and is in effect at the start time of the new motion). If sub-matches are present, then K-Sketch inserts a new reference frame *just above* the highest-level sub-matches (i.e., as close to the world reference frame as possible). For example, the man in Figure 5-10a consists of two legs with rotate motions plus other strokes that do not move. Selecting all of these objects and translating them will insert a new reference frame above these motions and preserve the motion of the legs (Figure 5-10b and c). Thus, this heuristic makes it easy to aggregate moving objects.

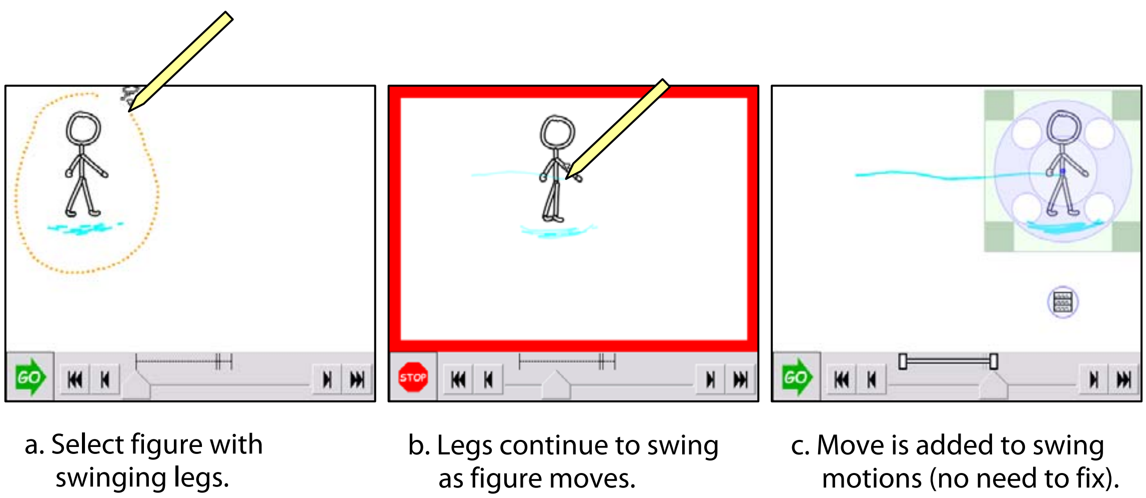


Figure 5-10: Heuristic 3 inserting above sub-matches. The figure has two legs, each with swing (rotation) motions. Moving (translating) the entire figure preserves the motion of the legs.

Heuristic 4 applies when *super-matches* are present. A super-match is an existing reference frame with a motion that operates on a superset of the strokes in the new motion (and is in effect at the start time of the new motion). If a super-match is present, then K-Sketch inserts a reference frame *just below* the lowest-level super-match (i.e., as far from the world reference frame as possible). For example, the man in Figure 5-11a translates across the screen. Selecting and rotating the man's leg will insert a new reference frame below the translation, making the leg swing while the whole man is moving (Figure 5-11b and c). Thus, this heuristic makes it easy to add motions to pieces of an object, while preserving any existing motion.

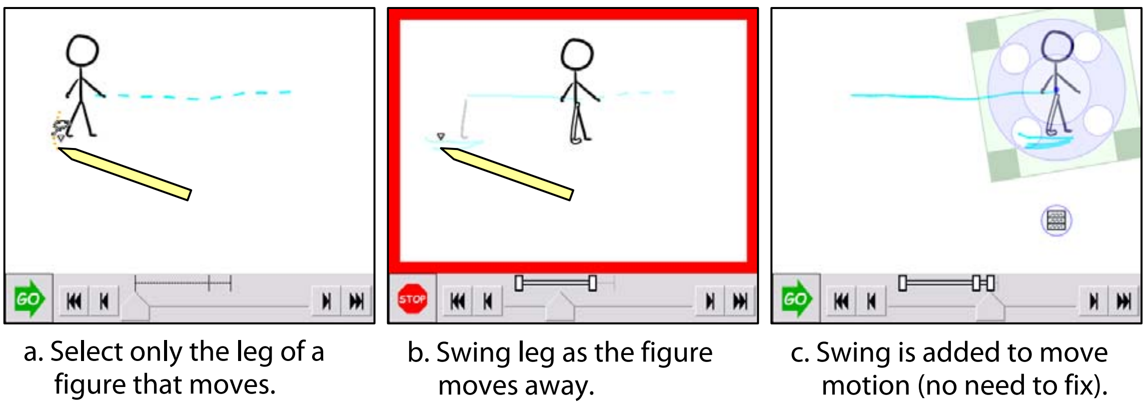


Figure 5-11: Heuristic 4 inserting below a super-match. The whole figure moves (translates) across the screen. Swinging (rotating) a leg by itself preserves the motion of the figure. Because the swing motion is not in the world reference frame, the user swings a ghost of the leg in (*b*).

Note in Figure 5-11b that the man moves away from the user's pen as the leg is being rotated. The user is actually operating on a transparent ghost of the leg in a fixed reference frame while the real man is moving. This happens whenever the user adds a motion that is not in the world reference frame. I made this design choice because I thought that novices would have an easier time manipulating objects in place rather than chasing them across the screen. In the user study presented in Section 7.2 on page 230, however, all users did exactly that, intuitively moving objects in the world reference frame instead of the ghost's fixed reference frame. It became clear that manipulating the ghost is difficult, because it forces users to split their attention between two locations on the screen.

Heuristic 5 applies when motions are present, but there are no matches, sub-matches, or super-matches. In this case, the user has selected *parts* of multiple, disconnected moving objects. This situation rarely occurs, and there is no example of it in my scenario library. In this case, I assume that the user wishes to break off parts of several moving objects and assemble them into a new aggregate. To get this behavior, K-Sketch discards the highest level motion of each part and replaces it with a new reference frame for the new motion. Any existing motions in lower reference frames are preserved below the new reference frame.

5.1.4.3 Fixing Poorly Chosen Reference Frames

The heuristics I just described choose reference frames that usually match novice users' expectations for new motions, but they fail in the important case shown in Figure 5-8b and c. Whenever users do not like the result of an operation, they can fix the problem with two button presses. The first step is to press the *Fix last operation* button in the toolbar (Figure 5-8d). This displays the **fix dialog**, a list of animated thumbnails showing what would have happened if the new motion had overwritten each possible reference frame or been inserted in between each frame. Touching one of these thumbnails causes the last motion to be removed and placed into the chosen reference frame.

In practice, the correction interface is quite intuitive and fast. It is usually needed only when a reference frame needs to be added to an object, as in Figure 5-8. After a new reference frame has been added, K-Sketch will tend to put subsequent motions in the correct reference frame automatically. Because few animations in my studies required more than three reference frames, this list is usually short. Figure 5-12 shows the three choices for the simple case in Figure 5-8e. Generally, if the number

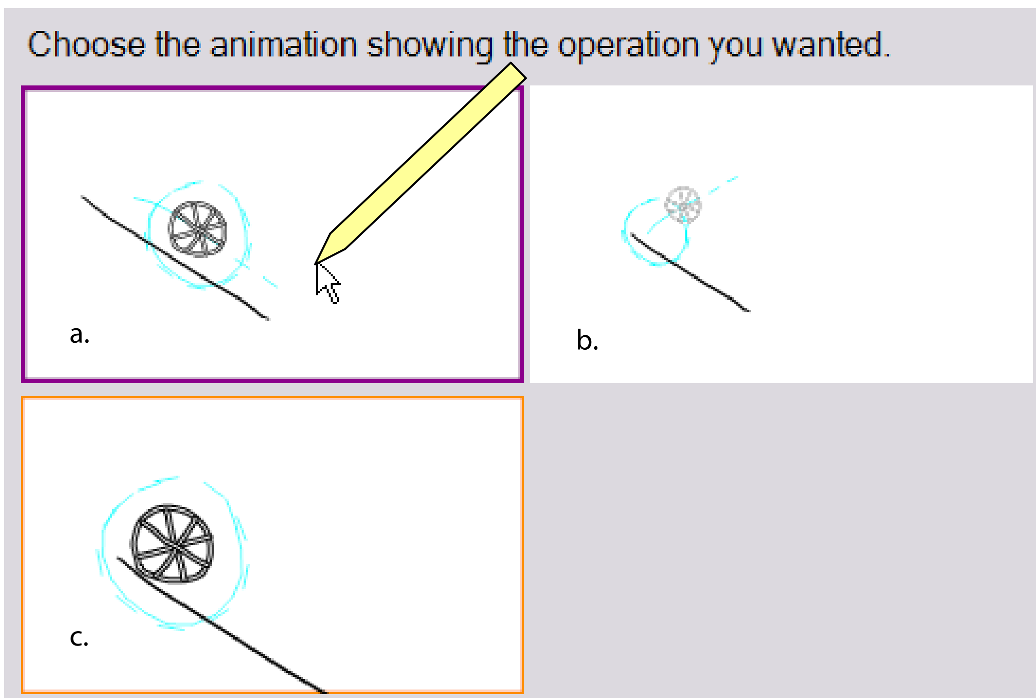


Figure 5-12: The fix dialog after adding a rotation motion to a translating wheel. *a:* The rotation goes in a new reference frame under the translation, causing the wheel to roll down the hill. The purple border indicates that the pen is hovering over this selection. *b:* The rotation goes in a new reference frame over the translation, causing the wheel to spiral out from its starting position. *c:* The rotation goes in the same reference frame as the translation, overwriting it. The orange border indicates that this was the state of the animation before the fix dialog was opened.

of existing reference frames for an object is *n,* then the maximum number of choices will be $2n+1$: one overwriting each existing frame, one inserted between each existing frame, one inserted at the very top, and one inserted at the very bottom. In most cases, some of these choices will be redundant, and K-Sketch attempts to remove these. For example, inserting a scale above or below a rotate will result in the same motion. Also, since most relative motions fall into a few types, it is often possible to put the most likely alternatives near the top of the list.

With these tools for overwriting and adding motions, it is possible to quickly define fairly complex motions. Because all of these capabilities are embedded in the object handle and the *Fix last operation* button, however, novices have very little to learn before they can take advantage of these capabilities. K-Sketch relies on users' intuitions for most cases, and when this intuition fails, users need only remember that one button fixes everything. The study presented in Section 7.2 on page 230 will show that users took naturally to this type of interaction.

5.1.5 Cut, Copy, and Paste of Objects and Motions

K-Sketch also provides the standard editing controls *Cut, Copy,* and *Paste* (see Figure 5-1a). When an object is selected, these operations work as they do in most graphical editors. The presence of a time dimension changes these operations only slightly. When an object is cut or copied to the clipboard, K-Sketch takes a snapshot of its state at that moment. Moving to a new point in time and pasting from the clipboard will paste that snapshot on the page, even if the original object is in a different state.

For example, say the user wishes to copy the walking man in Figure 5-10 on page 127 (in this case, only the object is copied, not the object and its motions). The user moves to the time indicated in Figure 5-10c (when the legs are separated) and copies the man. Then the user moves to the time indicated in Figure 5-10b (when the legs are together) and then pastes the man. In this case, the newly pasted object will have separated legs, even though the original man is present at this time with his legs together.

The previous example shows how objects are copied without their motions, but cut, copy, and paste work on motions as well. Before I discuss how clipboard operations affect motions, I will explain how to select and manipulate motions. Recall from Section 5.1.3 that motions can be selected through the *Select next guess* button in the toolbar. Pressing this button once will select an entire group of strokes if only part of it was selected, but successive presses will cycle through the group's motions. The first set of selected motions will include all motions related to the selected strokes, and later sets will isolate the motions in a particular reference frame. When motions are selected, their motion paths are highlighted in magenta and the object handle is positioned over them. Manipulating selected motions with the object handle will re-position them at all points in time.

Using Figure 5-10 on page 127 as an example once again, say the user selected the man at the time indicated by Figure 5-10c and pressed *Select next guess.* In this case, all three motion paths would be highlighted in magenta to show that they were selected, and the object handle would appear. If the user then moved the selected motions down slightly and then went to the beginning of the animation, the user would find that the path of the man had also been shifted down at this point in time. (Recall from Section 5.1.4.1 on page 124 that in an earlier design for K-Sketch, the same user actions would have produced a different animation. The man and his motion paths would have been in their original position at the beginning of the animation and then moved down suddenly at the point in time where the user moved the motion paths.)

All clipboard functions work on selected motions. This is how K-Sketch provides the 9–*copy motion* animation operation. A *Cut* or *Copy* command will put selected motions on the clipboard. (In the case of *Cut,* this will also remove the selected motions without removing the object itself.) The behavior of *Paste* depends on whether or not an object is selected when pasting occurs. If the selection is clear, a *Paste* command will paste the object and its motions (Figure 5-13e). If an object is selected, pasting will apply the clipboard's motions to the selected object (Figure 5-13h). In both cases, pasted motions start at the current time indicated by the time slider bar. By repeatedly pasting motions and re-positioning motion paths as shown in Figure 5-13(d–f), users can quickly create animations with many objects that move in the same or similar ways.

The rules just given define three kinds of pasting behaviors: pasting objects, pasting motions, or pasting objects *and* motions. The type of pasting that occurs depends on the state of the selection both at cut–copy time and at paste time. When I first started to observe novices using the 9–*copy motion* operation, I noticed that they had difficulty remembering these rules and were often confused by the result of a paste. Because of this, I made pasting a fixable operation. If the result of a paste is surprising, users can press *Select next guess* to fix the problem. The fix dialog shows the results of all types of pasting and allows users to select the desired result, as before. Note that this makes it possible to paste motions even if they were never explicitly selected, as in Figure 5-14.

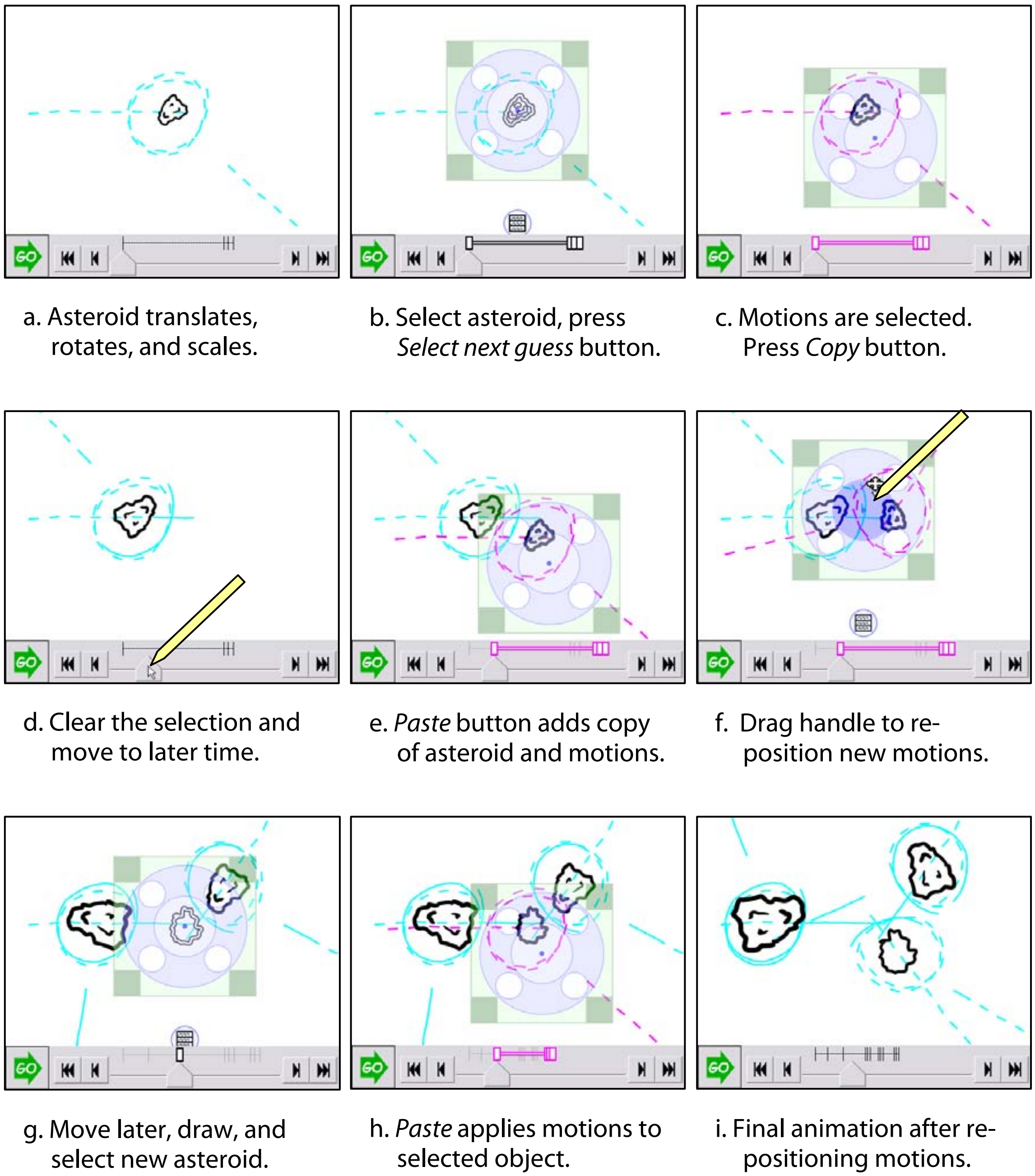
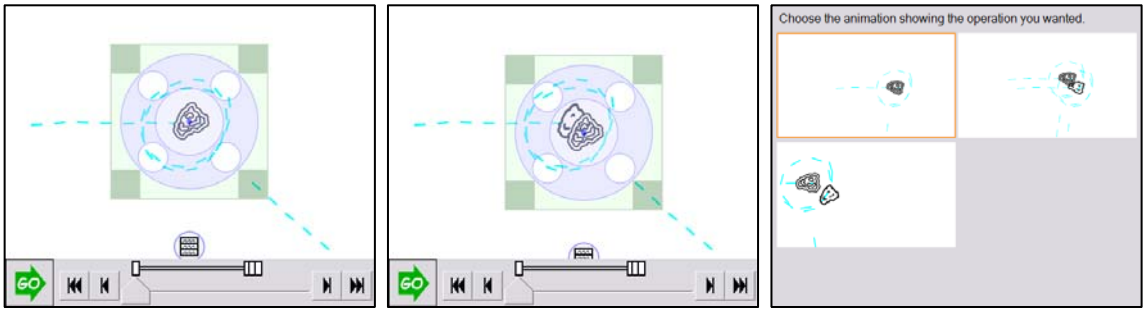


Figure 5-13: Animating an asteroid field with *copy motion*. Use *Select next guess* to select an object's motions. *Copy* commands will put objects and motions on the clipboard. *Paste* will either apply motions to selected objects or add copies of objects and motions (if the selection is clear).



a. Select asteroid and press *Copy* button.

b. Press *Paste,* get asteroid without motion.

c. Press *Fix last operation* to see other paste types.

Figure 5-14: Fixing a paste command. The user wants to make a second copy of the asteroid that has all the motions of the first. He first selects the asteroid, then presses *Copy* (*a*), and then presses *Paste* (*b*). This creates a copy of the asteroid that has no motions. To get the desired behavior, the user should have pressed *Select next guess* after selecting the asteroid, which would have selected the asteroid's motions. Pressing *Copy* would then copy the asteroid's motions instead of the asteroid itself. If the user cannot remember this distinction between copying objects and motions, then he can press *Fix last operation* after pressing *Paste* to bring up a list of alternatives (*c*). Selecting the animation on the top right produces the desired result.

Even though explicitly selecting motions is not necessary, it is still useful when a user wishes to copy or move *one* of an object's motions instead of *all* motions. To select a single motion, users can hold the Alternate button and tap on a motion path. (The more common lasso method does not select motion paths, because motion paths are often hard to avoid when lassoing objects.) Figuring out which motion path corresponds to a desired motion may become difficult as the number of motion paths grows. Users are aided in identifying desired motion paths by the fact that they move with their objects and often take a shape that corresponds to their type: As shown in Figure 5-13a, rotations are circles or semi-circles, translations are lines along the path of motion, and scales radiate from the center. Unfortunately, K-Sketch does not currently allow instantaneous changes (including 1–*appear* and 2–*disappear*) to be selected individually. Adding visible icons for instantaneous changes or adding selection capabilities to the timeline (see Section 5.1.8) may allow this in the future.

One other caveat for 9–*copy motion* is worth mentioning. When pasting objects with their motions, new copies will preserve all motions, even if they are arranged in a complex hierarchy of reference frames. When pasting motions onto another object, however, any motion that does not affect all strokes in the object will be discarded, because K-Sketch has no way of mapping sub-parts of one object to sub-parts of another object. For example, all motions in Figure 5-13a are copied onto the new asteroid in Figure 5-13h, because all motions affect the entire asteroid. The walker in Figure 5-10, however, has legs that rotate separately from the rest of his body. If a user copied the motions of this walker onto a new drawing, then the leg motions would be lost, because K-Sketch cannot determine which parts of the new object are the legs.

5.1.6 Recording Drawings

For K-Sketch to support the 10–*trace* animation operation, users must have a way to record the drawing of ink strokes over time. Any operation performed with the object handle can be recorded over time by holding the Alternate button before beginning the operation, but 10–*trace* cannot work this way. This is because tracing lines must be similar to normal drawing, and the Alternate button is already used to initiate a select operation when in draw–erase mode. (It does not make sense to use the object handle, either, because there is no need to select an object before tracing a line.) Since the 10–*trace* operation is not frequently needed, K-Sketch instead provides a special *Record drawings* mode controlled through a toolbar button (Figure 5-1d). In this mode, recording starts whenever the pen touches the screen for a draw or erase operation and stops when the pen is released.

10–*Trace* can save users considerable time in some situations. Consider the line traced by the machine tread in Figure 3-18b on page 70. Without the ability to record drawings, this line would need to be broken into many short pieces appearing one after the other. With recorded drawings, the effect can be created in a single operation. This is possible in Figure 3-18b, because the machine tread is moving slowly, but recorded drawings can also follow fast moving objects if the whole animation is slowed to a manageable speed (see Section 5.1.8).

5.1.7 Simplified Recording Controls

Instead of using standard VCR-like recording controls with three modes (recording, playing, and stopped), I chose to limit K-Sketch to two modes (going and stopped). This mode is controlled with a single button (Figure 5-1h) that can make the animation *Go* when it is stopped or make the animation *Stop* when it is going. Because of this, it is possible for a user to press *Go* and demonstrate a series of edits in real time until the end of the animation is reached. Controls like the object handle and motion paths are available whenever the pen is hovering over the canvas, even if the animation is going. Playing and recording are therefore virtually identical, except for the red border that appears around the screen when something is being recorded (e.g., when an object is manipulated or a line is drawn while going). I did this to reduce the number of controls and the possibility of confusion between playing and recording modes.

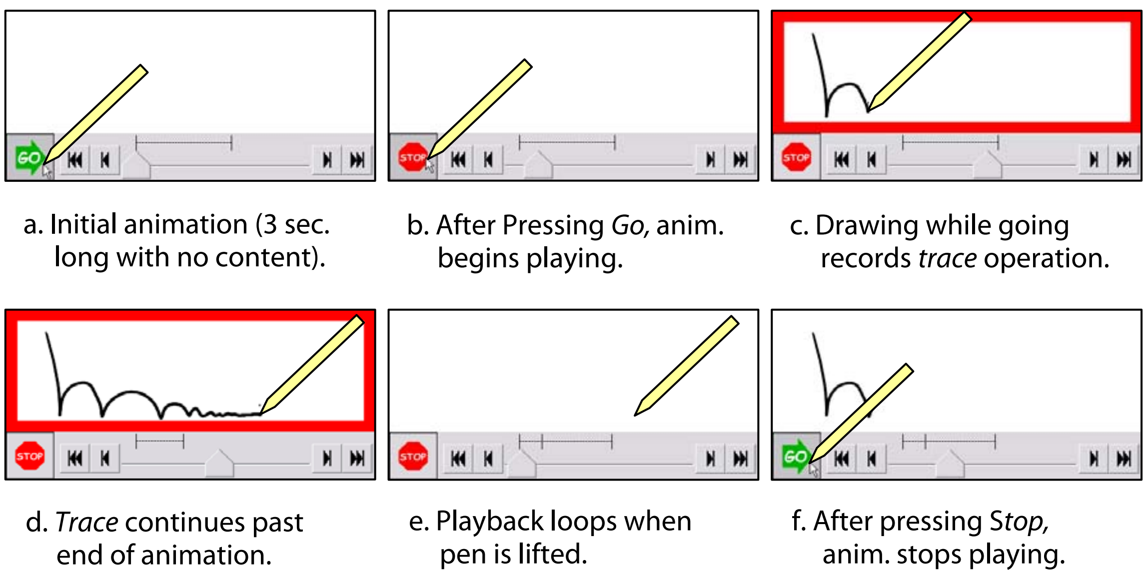


Figure 5-15: Recording a traced line by pressing *Go*. This method is less precise than using *Record drawings* mode, because the user cannot start drawing immediately after pressing *Go* (e.g., see *b* above). In this example, the *Loop playback* toolbar button has been pressed, causing the animation to play from the beginning once the end is reached.

Recording an animation by pressing *Go* and acting it out is somewhat easier for novices. This method frees them from using the Alternate button when animating with the object handle, and it frees them from learning about *Record drawings* mode.

However, this method is not useful for most of the scenarios in my library. First of all, recording will stop when the end of the animation is reached, which means that the length of the animation must be determined first. Secondly, every pause is captured (see Figure 5-15), and while removing these pauses is not impossible (see Section 5.1.8), it is cumbersome.

When playing animations, K-Sketch can perform some looping automatically. If the time slider bar is at the end of the animation, pressing *Go* will start playing from the beginning, as in most media players. Users can cause an animation to loop continually during playback by pressing the *Repeat playback* toolbar button in Figure 5-1d. This behavior is well-defined except when a user is recording a motion during looped playback. If the end of the animation is reached in the middle of a recorded drag or drawing operation, K-Sketch will finish recording the operation (changing the length of the animation) and loop playback as soon as the operation finishes.

5.1.8 Speed Control

Whenever objects are moving too fast for users to coordinate motions by hand, they can slow down the entire animation with the *Speed* button (Figure 5-1d). Pressing this button shows a slider (see Figure 5-16) that allows users to speed up or slow down the global clock. This slider has a logarithmic scale. The initial speed is *100%,* and it can range from *2%* on the left to *50x* on the right. This wide range ensures that users can produce animations of nearly any speed. Also, the speed of the animation is

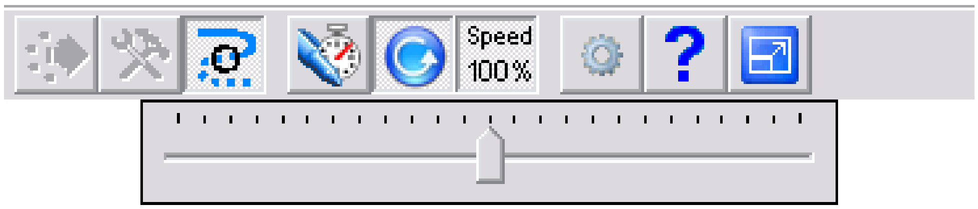


Figure 5-16: The slider that appears when the *Speed* button is pressed. The number on the button changes as the slider thumb is dragged. The settings on this slider are as follows (from left to right): 2%, 3%, 4%, 5%, 7%, 10%, 15%, 20%, 25%, 40%, 50%, 75%, 100%, 1.5x, 2x, 2.5x, 3.5x, 5x, 7x, 10x, 15x, 20x, 25x, 35x, and 50x.

saved with the file, which means that users can begin by moving objects slowly and speed up the animation as a final step.

To help users choose an appropriate speed, K-Sketch shows a short preview animation on the main drawing canvas whenever users manipulate the speed slider bar. This preview continually loops two seconds of the animation at the speed indicated by the speed slider. The two seconds chosen for the preview are in a window around the current time indicated by the time slider bar. The window usually lasts from one second before the current time to one second after, but the window may be shifted if the current time is less than one second from the start or the end of the animation.

5.1.9 Simplified Time Navigation Controls

Many animation tools have a complex timeline that shows every motion of every independently moving object. K-Sketch compresses this information into a simple slider with an iconic overview of edit history (Figure 5-17). By default, this slider is not even labeled with a time scale. Users can get a sense of scale by demonstrating a motion, but they can turn on a time scale (Figure 5-17g) through the options dialog box (accessed through the *Options* toolbar button in Figure 5-1e). This time scale doubles every time the end of the animation is pushed past the end point of the scale, so the slider can always reach any point in the animation.

Whenever the user performs an edit operation, K-Sketch adds two tic marks (or *events*) above this slider bar: one where the edit begins, and one where it ends (or a single event for instantaneous operations). When objects or motions are selected, the timeline highlights events related to the selection, as shown in Figure 5-18b and c. This provides some deeper detail without being distracting or overwhelming.

Users can move through time by dragging the slider thumb, which snaps to the events shown above it. Tapping on either side of the thumb will jump to the next event. The slider also has four navigation buttons (Figure 5-17(a–d)). The outer buttons jump to the beginning and end of the animation. The inner buttons *nudge*

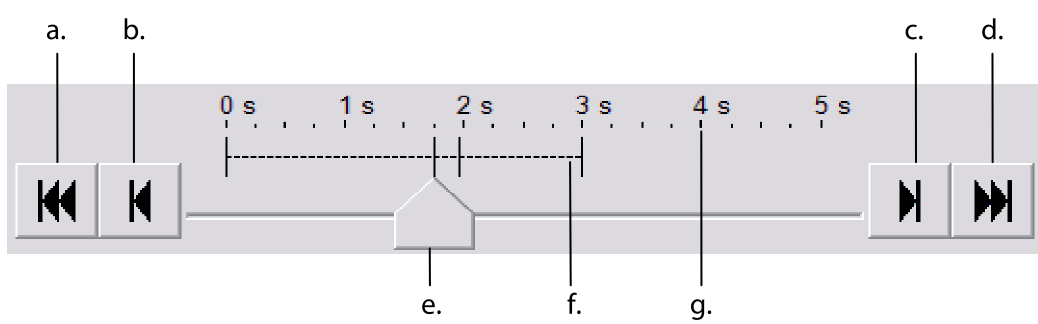


Figure 5-17: Time slider bar. *a: Go to beginning* button. *b: Nudge backward* button. *c: Nudge forward* button. *d: Go to end* button. *e:* Slider thumb. *f:* Event tic marks. *g:* Time scale.

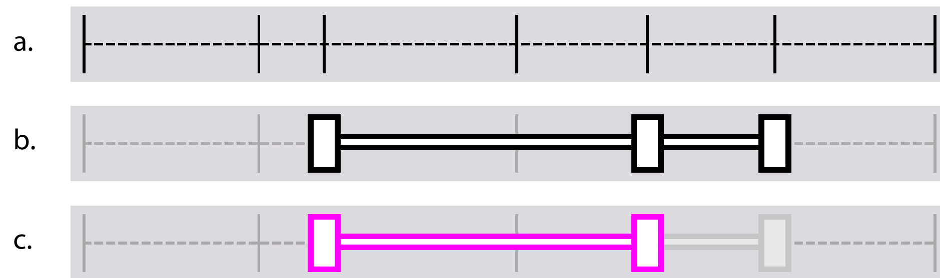


Figure 5-18: Time slider bar event tic marks. *a:* Normally, all events are thin lines. *b:* When an object is selected, events related to that object are highlighted. *c:* When motions are selected, they are highlighted in magenta.

time forward or back by 1/15th of a second (or to the next event, whichever is closer), which can be helpful when creating an animation frame-by-frame.

Users can slide event marks along the timeline to tweak the start or end time of a motion. By default, moving an event also moves any others that occur after it, but holding the Alternate button allows an event to be moved without affecting later events. The order of events is always preserved, however, so moving an event past another will cause both to be pushed out of place. The ability to slide event marks also makes it possible to add time to the beginning or the end of the animation.

The ability to move event markers is a powerful tool, but the current design can sometimes frustrate users' attempts to edit the timeline. All events that occur at the same time are collapsed into one tic mark, and users have no way of separating them once they are combined. For example, if a motion ends at the end of the animation, it is impossible to extend the length of the animation by dragging the last tic mark without also changing the end time of the motion. The ability to select motions through these event markers would also be helpful for reasons described in Section 5.1.5. In the future, I plan to redesign this timeline to enable selection and separation of events.

5.2 Examples of K-Sketch in Action

Before describing K-Sketch's implementation, I will present four detailed examples of K-Sketch in action that tie together the concepts presented in the previous section. Each of these examples is part of a task I used in the studies presented in Chapter 7, and each of these tasks was based on a scenario in my scenario library.

5.2.1 Scenario 55—Chemistry: Particle Collisions with K-Sketch

In this scenario, a positron (β^+) collides with an electron (e^-) and the two explode, producing gamma rays (γ) that fly off in opposite directions. The steps in this scenario are shown in Figure 5-19.

The user begins by drawing a positron and selecting it by holding Alternate and drawing a lasso around it (Figure 5-19a). The object handle appears, and the user demonstrates a translation motion to the right (b) by holding the Alternate button before dragging the handle's *translate* region. When the user lifts the pen, the animation stops (c). The user then rewinds to the beginning of the animation, draws an electron, and selects it (d). When demonstrating the motion of the electron, the user aims for the right end of the previous motion path (e), timing the collision of the two particles by hand (f). Since the collision is sloppy, the user rewinds the animation to the point when the particles first touch each other (g).

To make the particles, explode, the user first selects the eraser in the toolbar and erases both particles (Figure 5-19h). Erasing the particles at this time makes both disappear from this point into the future. A ghost is left behind to remind the user where to draw the explosion, which will appear at this moment in the animation (i). The explosion only appears for a split second, however, so the user *nudges* forward $1/15^{th}$ of a second (j) and erases the explosion (k). Finally, the user draws two gamma rays (l) and demonstrates their motion in opposite directions.

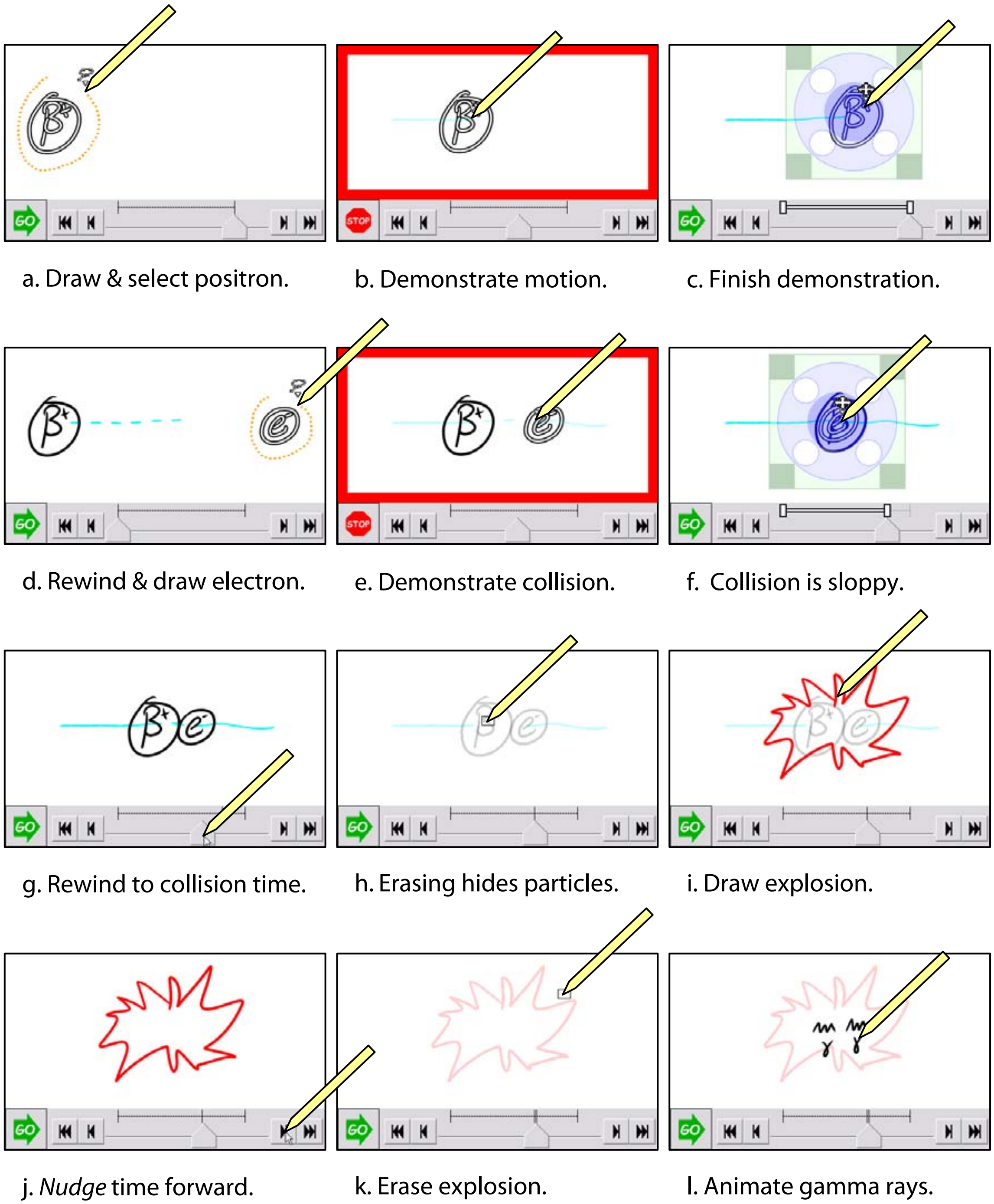
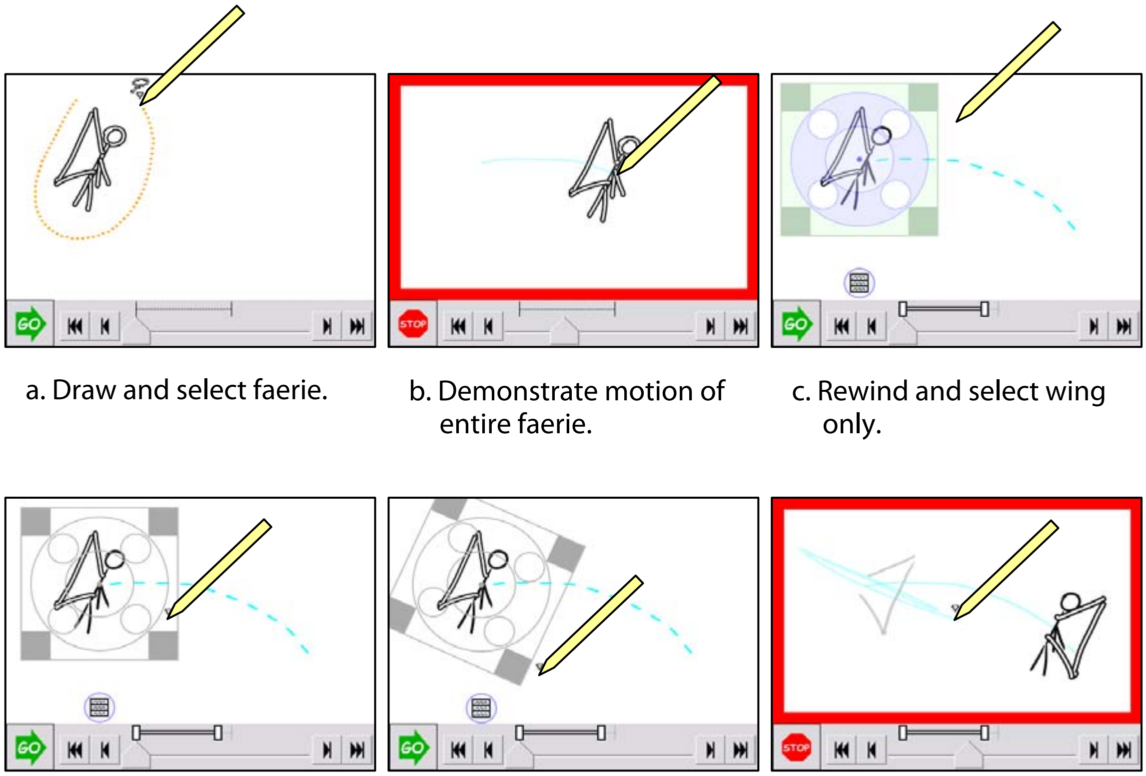


Figure 5-19: Animating scenario 55—*Chemistry: particle collisions.* After demonstrating a sloppy collision, the user rewinds until the particles are just touching and hides them at that point. An explosion flashes in and out in 1/15th of a second. Finally, gamma rays appear and fly off.

5.2.2 Scenario 30— Faerie Adventures

This animation shows a faerie flying across the screen, flapping its wings as it goes. Figure 5-20a shows the faerie in initial position, slightly tilted. The user begins by demonstrating the faerie's motion across the screen (b) and rewinds to the beginning. The user must now de-select the faerie and select only her wings (c). To make the wings flap back and forth, the user must position the object handle so that the center is at the base of the wings and orient it so that it can be scaled perpendicular to the body. To position the object handle, the user selects *Move handle* in the context menu (see Figure 5-5c) and drags the handle to the base of the faerie's wings (Figure 5-20d). To orient the object handle, the user selects *Rotate handle* in the context menu and tilts it into position (e).

The user then demonstrates the motion of these wings by animating a stretch motion back and forth (Figure 5-20f). Because this motion is not in the world reference frame, the user must interact with a ghost that stays in the faerie's initial position. There is no need to *fix* this motion, because K-Sketch assumes that motions applied to a sub-part (the wing) of a larger moving object (the faerie) should add to existing motions instead of overwriting them (see Section 5.1.4).



a. Draw and select faerie.

b. Demonstrate motion of entire faerie.

c. Rewind and select wing only.

d. Use *Move handle* to put center at base of wing.

e. *Rotate handle* until it is square with body.

f. Stretch back and forth to animate flapping wing.

Figure 5-20: Animating part of scenario 30—*Faerie adventures.* The user starts by translating the entire faerie across the screen and then goes back to animate the wing. To get a flapping motion, the user can stretch the wing back and forth with a non-uniform scale, but the handle must be repositioned. *Move handle* positions the center at the base of the wing, and *Rotate handle* turns it so that the stretch motion will be perpendicular to the base.

5.2.3 Scenario 46—Plane Story

At one point in this story, a plane moves into a position, shoots a missile, and moves back. The plane translates, rotates, and scales as it moves. After drawing the plane in its initial position and selecting it (Figure 5-21a), the user demonstrates both a translate and tilt using the *orient to path* region of the object handle (b). The plane could be moved both into and out of shooting position in one operation, but the user demonstrates only the first half so that it will be easier to coordinate motions. The user then rewinds and demonstrates a scale motion to shrink the plane (c). This overwrites the translation, because the new motion affects the same strokes as the existing motion. The user must therefore press the *Fix last operation* toolbar button to call up the fix dialog (d).

After selecting the appropriate animation in the fix dialog, the timeline shows that both motions are present, though they end at slightly different times (Figure 5-21e). To make both end simultaneously, the user grabs the event on the right and moves it on top of the next event. The event snaps into place, and a single event marker remains (f). Now the user can draw a missile and make it fly away (g). To move the plane back into its original position, the user demonstrates another *scale* (h) and *orient to path* (i). Because the plane already has reference frames with these types of motions, there is no need to open the fix dialog again. Note also that because the user did the *scale* operation before *orient to path*, the scale could be defined as if it were in the world reference frame, and there was no need to manipulate a ghost as there was in Figure 5-20f.

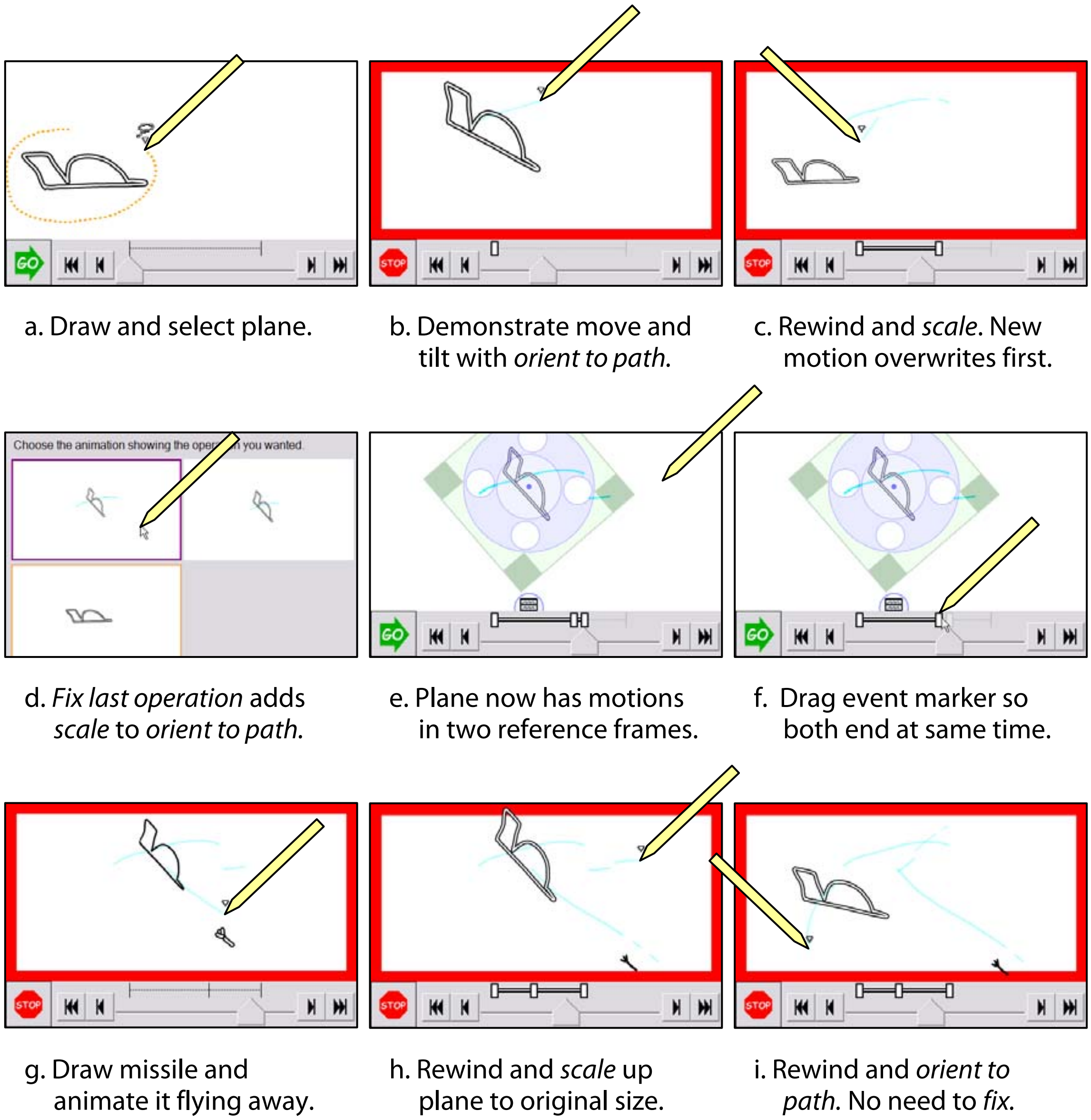


Figure 5-21: Animating part of scenario 46—*Plane story*. The plane moves (*translates*) tilts (*rotates*) and shrinks (*scales*) into position, fires a missile, and then moves back into its starting position and size. Moves and tilts can be done with *orient to path* operations (*b* and *i*). Note that the fix dialog is needed only once (*c* and *d*), and subsequent motions are automatically put in the correct reference frame.

5.2.4 Scenario 51— Detection of Distant Planets

In this final example animation, a star emits a stream of light waves. This can be a difficult animation to demonstrate, because some precision is needed. If the waves do not have the right distance between them, they will not look like waves at all, but a random collection of lines. Fortunately, the *copy motion* animation operation makes this fairly easy in K-Sketch.

The user starts by drawing the star and a single line for a wave (Figure 5-22a). Then the user demonstrates the motion of this wave to the right (b) and erases it to make it look like it disappears off screen (c). To copy the motion of this wave, the user rewinds to the beginning of the animation, selects the line (d), and then presses *Select next guess* in the toolbar to select the wave's motions (e). The user then clears the selection, nudges time forward five steps (one third of a second) using the *Nudge forward* button (f), and then pastes a new wave (g), positioning it over the previous wave (h). By repeating steps f and g, the user can quickly scale up to a large number of waves (i).

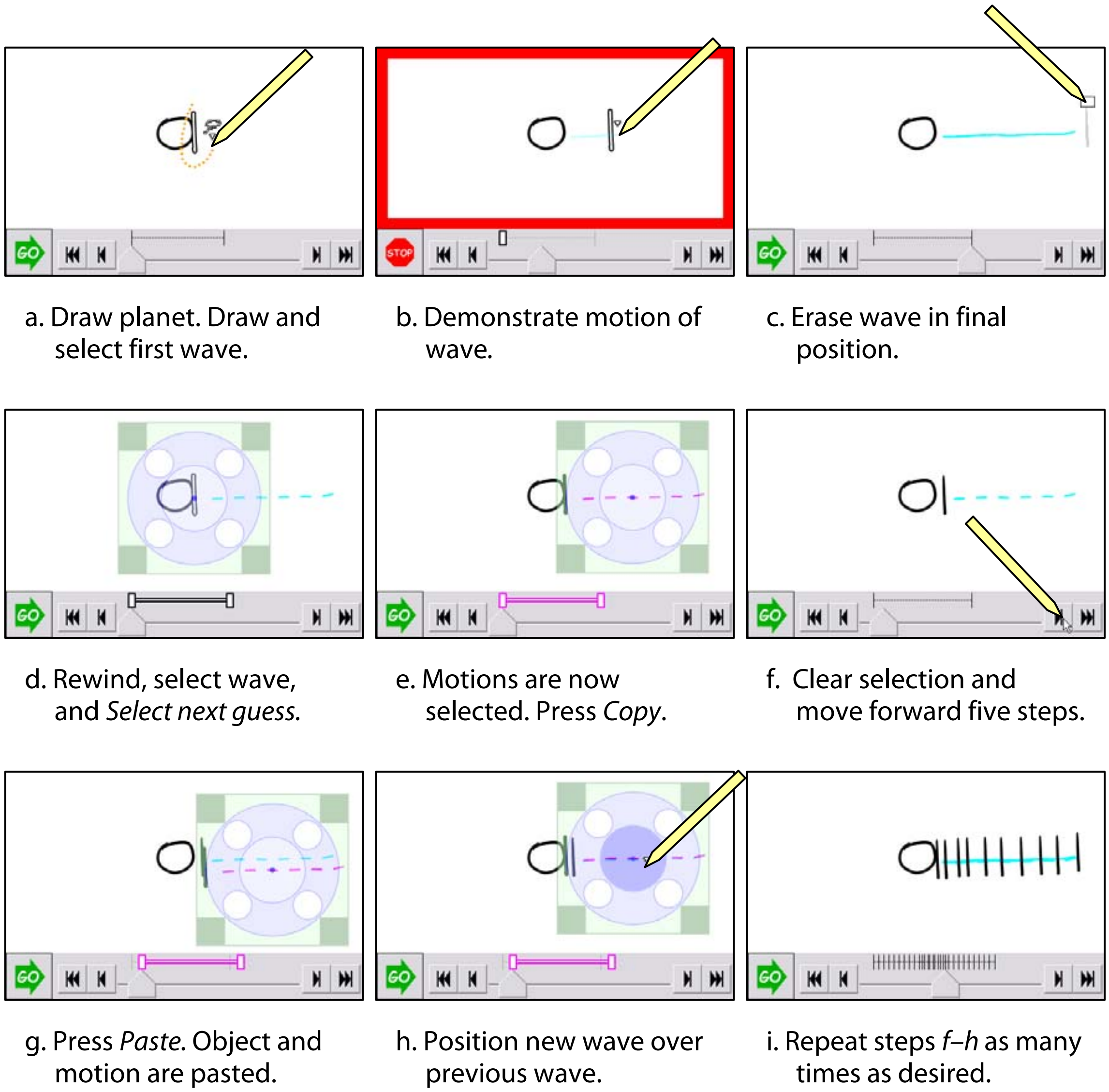


Figure 5-22: Animating part of scenario 51—*Detection of distant planets*. To scale up to a large number of waves, only a single wave needs to be animated and copied. By moving forward exactly five steps for each new wave, the waves are kept a fairly constant distance apart.

5.3 Implementation

The K-Sketch design presented in this chapter has been implemented in C# for the Tablet PC. The implementation makes heavy use of the Piccolo.NET graphical interface toolkit [21], which I modified to use the ink collection, rendering, and selection methods provided by the Microsoft.Ink API version 1.7. K-Sketch has a total of 67 classes with 28,000 lines of code, plus 6 classes with 2,700 lines of test code and 7 classes with 3,900 lines of code added to Piccolo.NET. The installer can be downloaded from http://dub.washington.edu/k-sketch/download/ and takes only about 3.5 megabytes of memory, including Microsoft's ink libraries.

A class diagram for the main classes in K-Sketch's data model appears in Figure 5-23. Pen strokes are represented by instances of the KStroke class, and the motions applied to strokes by instances of the KChange class. Both are descendants of a common class (KObject), because both can have a visible representation. Every KObject has a start time, an end time, and an ordered list of the changes that occur to it over the course of the animation. For KChanges, the start and end times correspond to the time when the change starts to occur and when it completes. For KStrokes, the start time is the moment the stroke starts to appear, and the end time is the moment it is fully visible. (These are the same unless the a stroke is *traced* over time.) In addition, each change has a list of *targets,* which are the objects on which the change operates.

KChange objects are very different from command objects [115], because they do not have their own "Do" or "Undo" methods, and they do not operate on the model. Instead, these changes define the properties of their target objects during the period

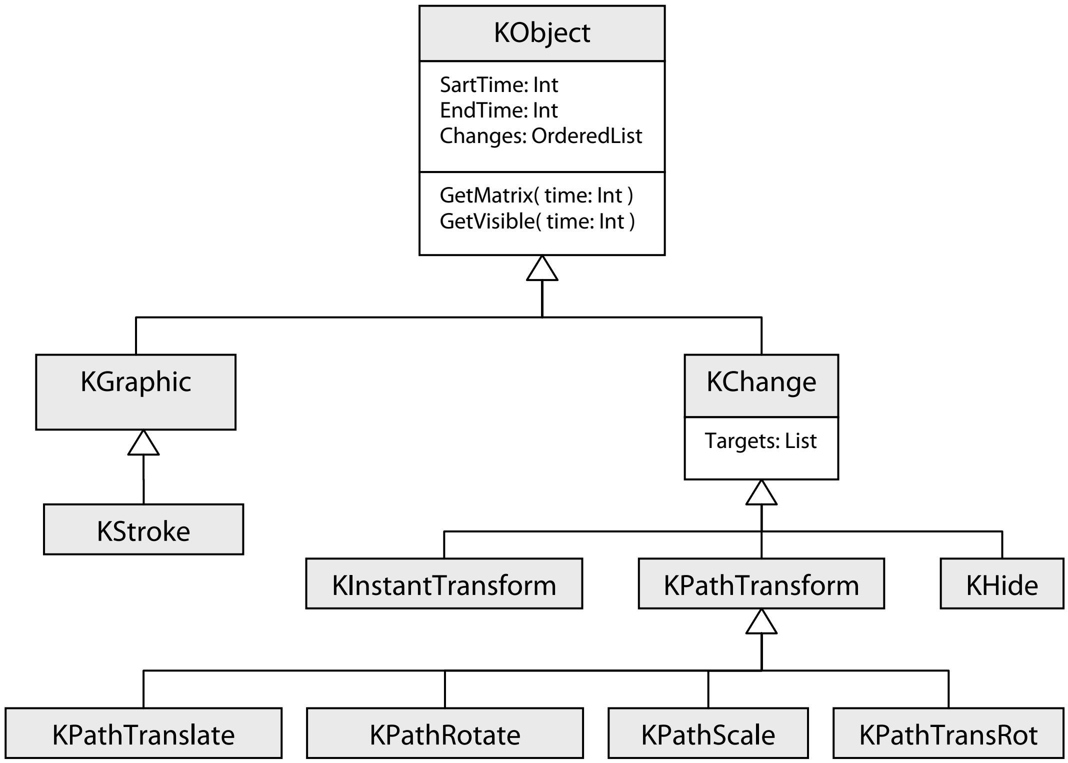


Figure 5-23: Class diagram of the K-Sketch data model. Both graphical objects (KGraphics and KStrokes) and motions (KChanges) inherit from a common superclass, KObject. All objects have start and end times, a list of changes, and queries that can return property values for any time in the animation. The names of most objects are self-explanatory, except for KPathTransRot, which is for *orient to path* motions.

that they are active. Currently there are two types of changes: those that define a stroke's position (KInstantTransform and subclasses of KPathTransform) and those that define its visibility (KHide). Thus, whenever an object is queried for its position or visibility, a time parameter must be specified. This design allows fast access to every object's properties at every moment in time.

Whenever a user performs an edit, K-Sketch must check to see if it conflicts with other changes. If so, these other changes must be modified or removed before the new change can be inserted. This checking is performed in a single class called KOpController that defines how changes should interact. Much of the complexity of the system is concentrated in this class.

KStrokes and KChanges are assembled into model objects that represent whole animations. Figure 5-24 shows an example of what a model can look like under normal use. Here, the world reference frame is at the top, changes add reference frames below the world reference frame, and strokes are at the bottom. This model has two objects: the object on the left has two reference frames with two motions each, and the object on the right has one reference frame with one motion. As the figure illustrates, this data structure can become more complex than a conventional scene graph. Most scene graphs take a tree-like structure, but a K-Sketch model is a directed, acyclic graph.

This model architecture has many benefits. Because the properties of each motion are contained in a single object, motions can be added, removed, or modified by operating on one object. This makes code easier to maintain, and it keeps the memory footprint of command objects small, allowing deep undo stacks. Because the model is not forced into a tree-like structure, strokes can easily move from one group to another during the animation. This makes K-Sketch more flexible than many drawing programs. In PowerPoint animation, for example, if an object is already moving with one group, taking it out of that group to move it with another will cause

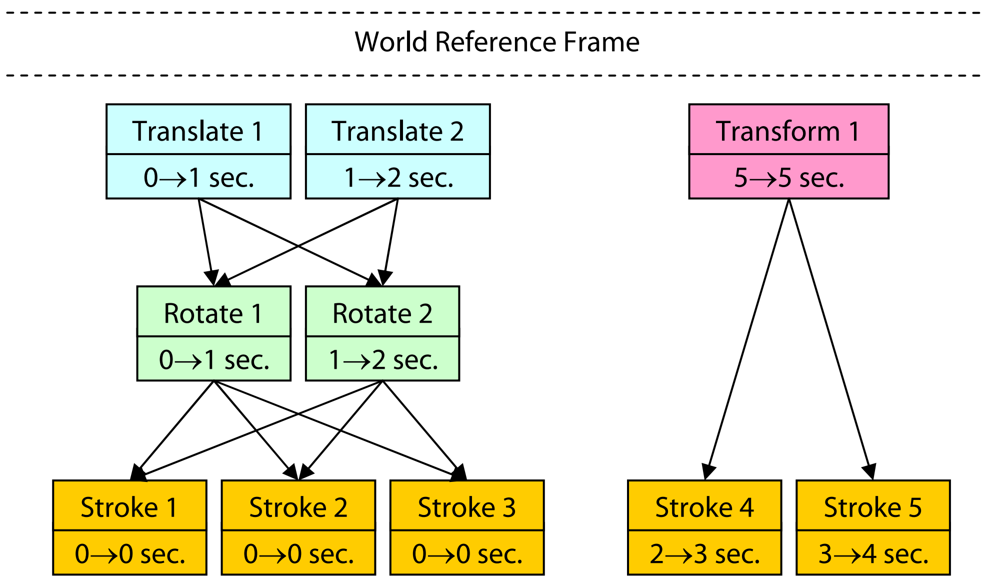


Figure 5-24: An example K-Sketch scene graph. This graph contains two objects. The object on the left has three strokes that are visible at the start of the animation, and it is both rotating and translating over time. The object on the right has two strokes that are traced out after the first object finishes moving, and then transforms instantaneously.

all of the original group's motions to be lost.

As support for the *move relative* operation evolved, however, the complexity of K-Sketch's scene graph became a major engineering obstacle. One example will illustrate the problems with this scene graph structure. A set of strokes is defined to be a *group* if there exists a change object that operates on that set of strokes. This definition makes it hard to determine the reference frame for new motions, because no connections are maintained between changes in the same reference frame. To determine the current set of reference frames, K-Sketch must search through the entire object graph. These searches can be so time consuming that they are skipped in some cases, splitting reference frames when they should remain united and creating an overly complex hierarchy of motions.

To eliminate this problem, future revisions of K-Sketch's data model will have a simpler, tree-like structure that groups strokes into objects and changes into reference frames. K-Sketch can still perform the grouping implicitly as before. It may also be possible to preserve the ability to move strokes from one group to another during the animation. This would require a modified architecture that allows strokes and groups to exist in several parts of the tree at once. I leave this for future work.

Every user action in K-Sketch creates a set of hierarchical command objects [115] that add, modify, or remove KObjects. Since each user action can cause complex changes to the object graph, I use a transactional model for command objects similar to the one used by Edwards and colleagues [50]. When the user initiates a command, a transaction begins, and command objects are collected on the undo stack. When the command finishes, these smaller command objects are bundled into one large command object so that users can undo or redo them as a unit.

K-Sketch separates view and controller functionality from the data model, as do many user interface applications. The controller is contained in a set of classes that inherit from a common Piccolo.NET interactor class. The view is controlled by subclasses of Piccolo.NET windows and graphical objects. Whenever a command object changes the model, the view adds, modifies, or removes a graphical object for every KObject. Because these graphical objects are nearly identical to KObjects, this strict separation of model and view wastes time and space.

As this strict separation of model and view is not helpful for K-Sketch, my future revisions may modify the view to use KObjects directly. Multiple views would still be possible if view-dependent options (e.g., scale) were passed as arguments to a view rendering function. If view-dependent data needs to be kept with KObjects (e.g., highlight state), then the architecture could be extended to allow views to store data temporarily in KObjects.

Since there is no standard programming support for physical mode switches in Tablet PCs, the current K-Sketch implementation uses most keyboard keys as Alternate buttons. Tablet PCs allow physical buttons to be mapped to keyboard keys, and these are the most common Alternate buttons when K-Sketch is used on a Tablet PC in slate mode. In user tests, I also provide users with a handheld remote control with buttons mapped to keyboard keys.

While K-Sketch is a prototype system, it is full-featured and robust enough for use in real work situations. Animations can be exported to Shockwave Flash format (.swf) for posting on the web. Unfortunately, the current implementation of Flash export discards variable width information for pen strokes, but most users can ignore this. K-Sketch can also keep extensive logs of usage data that are posted periodically to a server (if users give permission). This gives a detailed picture of how people use

K-Sketch in the real world. An analysis of the logs I collected appears in Section 8.1.2 on page 281.

The K-Sketch system presented in this chapter gives novices a fast, simple way to express a variety of animations. The use of sketching preserves creative flow by helping users avoid focusing on little details when they need to be thinking about higher level issues. K-Sketch also greatly simplifies the animation process by taking advantage of users' intuitive sense of space and time and providing a simple set of tools for adding, editing, and coordinating motions. The following three chapters will present evidence of these facts by describing evaluations I have conducted over the life of the K-Sketch project, including formative evaluations, summative laboratory evaluations, and real-world evaluations.

6 Formative Evaluations

The K-Sketch system presented in Chapter 5 evolved over a period of several years. Before presenting the summative evaluations I conducted to judge the work as a whole, I present in this chapter formative evaluations that guided the design of K-Sketch. Section 6.1 begins with an overview of the design changes that occurred over many design iterations. Formal and informal evaluations took place continually throughout this iterative process, and I present two of the formal evaluations in the following two sections. In Section 6.2, I describe an evaluation that compared an expert's performance when creating ten animations with Flash, PowerPoint, and an early version of K-Sketch. I then present a laboratory evaluation of an earlier K-Sketch version in Section 6.3.

6.1 Design Iterations

Like many user interface designs, the K-Sketch interface evolved gradually. From June 2003 to May 2005, the design evolved through sketches and animated mock-ups. Many fragmented ideas appeared during this period, but a coherent design evolved in five stages. Implementation began in June 2005, and the design went through at least

23 identifiable revisions until August 2008. Instead of a tedious list of every detail of every evaluation and revision, I simply give an overview of my early design sketches for K-Sketch and then explain how important aspects of the implemented interface evolved.

6.1.1 Early Design Sketches

The earliest sketches of K-Sketch's interface consisted of fragmented ideas for expressing various types of motions and transitions. The very first sketches contained ideas for supporting what came to be known as the *appear, disappear,* and *physically simulate* animation operations. It soon became clear, however, that there was a plethora of animation operations available but no clear way to choose between them. This was the impetus for exploring interface optimization as part of this project.

Gradually, a coherent design emerged. The first full design had many of the elements found in the final K-Sketch user interface, including a manipulator similar to the current object handle, a time slider bar, motion paths, and ghosts. This design also modeled animation as a sequence of editing steps over time and allowed motions to be defined through demonstration. Later revisions showed how to coordinate multiple moving objects.

Throughout these early sketches, I assumed that users would access some commands (such as cut, copy, and paste) through pen gestures. I abandoned pen gestures when implementation began, however, because novices can have difficulty recalling gestures. Recall is even more difficult when recognizer inaccuracy causes the system to reject correct gestures, as was frequently the case with the recognizer included with the Microsoft.Ink API. Instead of pen gestures, the current implementation uses a standard toolbar for issuing commands. Research has produced numerous alternatives to standard toolbars that could preserve users' creative flow slightly better (e.g., crossing widgets [14]), and I leave this for future work.

Because design feedback is especially valuable in the early stages of design, I wanted to test low-fidelity prototypes of my design on actual users. Unfortunately, I was unable to find a low-fidelity prototyping method that was appropriate for K-Sketch. Most low-fidelity prototyping methods, such as paper prototyping [127], simulate interfaces through a series of discrete steps, with long pauses between user actions while the designer reconfigures the interface. This is an acceptable way to test many designs, but the purpose of K-Sketch is to support the flow of users' ideas with minimal obstacles or distractions. Therefore, long interruptions can completely destroy a user's experience.

I began to address this prototyping problem by designing SketchWizard, a system that captures user sketches on a digitizing tablet and allows designers to manipulate them in real-time on a remote computer [43]. This system gives designers the ability to transform user sketches faster than they can with any other prototyping technique, but it falls short of allowing a designer to simulate K-Sketch. SketchWizard cannot currently simulate real-time interactions with K-Sketch's object handle, nor is it able to capture and play back animations. I plan to address these limitations in the future versions of SketchWizard by including facilities for attaching pen behaviors to graphics like the object handle and by integrating K-Sketch itself into SketchWizard.

After showing my early design sketches to other user interface researchers, some responded with concern that demonstrating a motion would be too difficult, even when a rough motion was acceptable. I believed that most of these concerns would disappear when people began using the system to produce rough animations. However, I made plans to add a speed control (described in Section 5.1.8 on page 140) should these concerns persist once K-Sketch's implementation took shape. Feedback throughout the implementation process confirmed that demonstrated animation is easy for most users after a little practice, but I added the speed control anyway to comfort more timid users. Later analysis of logged K-Sketch usage found that this speed control is needed in about 20% of animations, and in those animations it is used only twice on average (see Section 8.1.3.2 on page 292). This suggests that the speed control is comforting for users, but they do not need it frequently.

Implementation began when it became clear that I had obtained most of the substantive feedback I could get from sketches and mock-ups. At that time, the interface optimization method was still evolving, but the design provided most of the animation operations that the final implementation provides. Early interface optimization results made the *orient to path* operation appear more important than later results did, and this is part of the reason why it remains in the final interface. Some details of the *copy motion* operation were unclear, but I had a plan to provide it through copying and pasting of motion paths. *Move relative* was the only operation that did not have explicit support, but some relative motions were still possible, because motion paths could be manipulated as first-class objects (i.e., in the same way as any other stroke, as explained in Section 6.1.6).

6.1.2 Selection

As explained in Section 5.1.3 on page 113, K-Sketch users select objects by holding the Alternate button and drawing a loop around ink strokes or tapping on them. Selected strokes are rendered in outline, and the object handle appears over them when the selection operation completes. This basic selection behavior was present in the first iterations of K-Sketch's implementation, and it is unchanged in the final version. However, the selection process did continue to evolve in subtle but important ways.

K-Sketch's object handle can be distracting once a user has finished manipulating an object, and de-selecting is therefore a common operation. In the initial version of K-Sketch, users de-selected objects either by holding the Alternate button and selecting empty space or by drawing an object somewhere else. In a user study, however, I found that many users would de-select by simply tapping outside the widget without holding the Alternate button, causing animations to be littered with stray ink marks. I therefore revised the design to keep the pen in select mode as long as an object was selected, thus requiring users to tap on empty space to deselect objects before they could draw. This brought about new confusion, unfortunately, as many users were baffled when they tried to draw and saw a selection loop instead. This happened even though the cursor clearly showed a lasso icon, indicating that the cursor was not very helpful. I eventually reverted back to the original design. Stray ink marks caused by de-selection are still fairly common in K-Sketch files, but due to the rough nature of these animations, few users seem to mind.

In many early formal and informal studies, it also became clear that the absence of explicit grouping made it hard for users to select objects consistently. If users did not draw a loop around every part of an object when selecting it, some strokes would be left behind when the object was moved. Believing that implicit grouping was important for preserving creative flow, I tried several strategies for dealing with this problem. First, I displayed a convex hull outline around implicitly grouped objects to make grouping more visible and to guide users in drawing selection loops (see Figure 6-1). I later removed this outline because of performance problems and to reduce visual clutter.

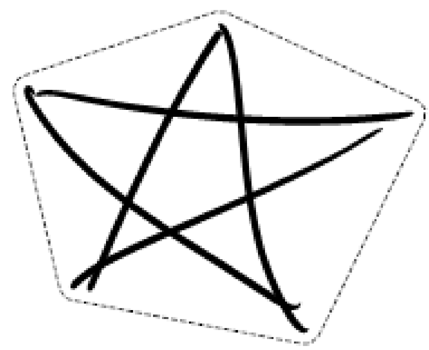


Figure 6-1: A convex hull outline that was present in some iterations to make grouping visible and help users draw selection loops.

A second strategy was to provide a mechanism for modifying a selection when a selection loop omitted strokes or added undesired strokes. In the original selection scheme, the object handle appeared immediately after the user's pen finished a selection loop or tap. In the modified scheme, shown in Figure 6-2, the handle did not appear until the user released the Alternate button. This gave users the opportunity to add or remove strokes from the selected set by tapping on them. This scheme was a surprising failure, because users almost universally forgot to release the Alternate button after completing a selection loop. In retrospect, this behavior is consistent with Guiard's kinematic chain theory of bimanual interaction [64]. This theory suggests that the action of the non-dominant hand should precede the action of the dominant hand when the two are working together. The original selection scheme was consistent with this theory, but the modified scheme required an action from the non-dominant hand both before and after the user's selection loop.

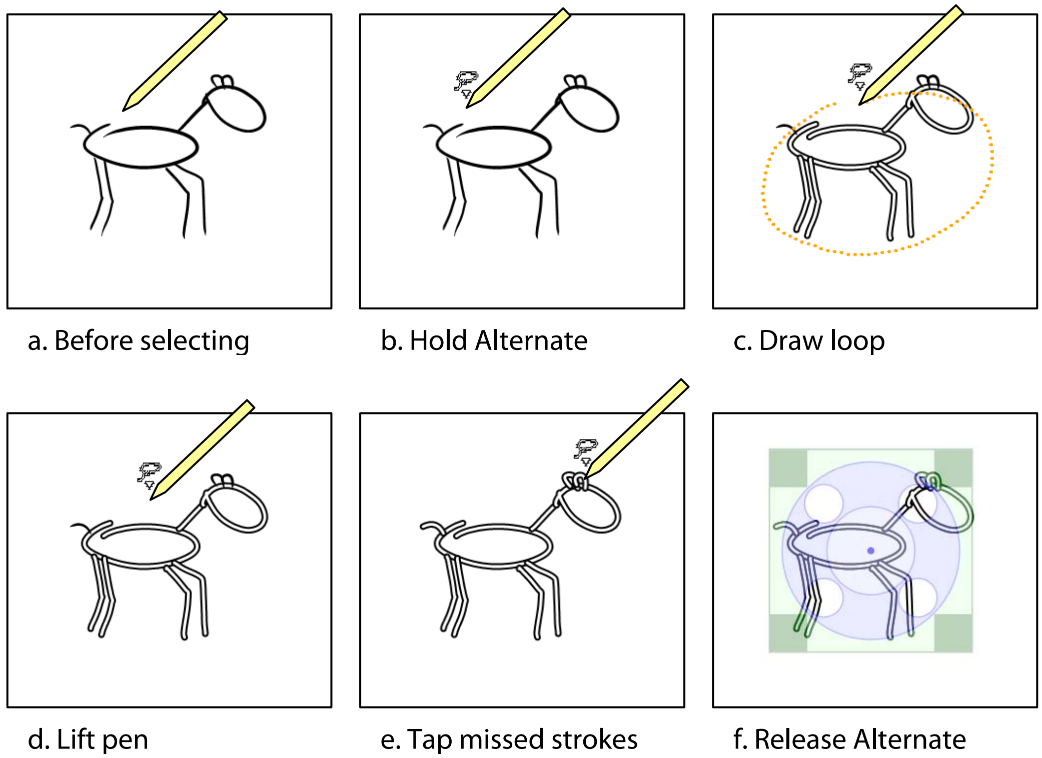


Figure 6-2: A surprisingly unsuccessful selection scheme. The allowed users to grab missed strokes, but they consistently forgot to release the Alternate button.

My third strategy for dealing with this selection problem was to automatically select entire groups of objects when the user tapped on a stroke from that group. While helpful, this was not a complete solution, because strokes could exist in many different groups as part of a hierarchy of groupings. Later iterations of K-Sketch added the *Select Next Guess* button to cycle through the different groups associated with a stroke. This enables users to precisely select groups of strokes that have been grouped in the past, but users have found it awkward in practice. Section 6.1.6 explains how this button evolved along with support for the *move relative* operation. In future versions of K-Sketch, I may refine this mechanism further.

The mechanism for selecting motion paths has also gone through some revisions. At first, motion paths were selected with loops just like any other stroke. This made the common case of selecting objects more difficult, because it was hard to avoid selecting motion paths. By accidentally selecting and manipulating motion paths in their animations, novice users could introduce errors that were hard to diagnose. For this reason, motion paths were turned off by default in early versions. When it became clear that motion paths should not be first-class objects (see Section 6.1.6), I made them selectable by tapping only, and turned on motion paths by default. I also made selection feedback for motion paths more noticeable, as shown in Figure 6-3.

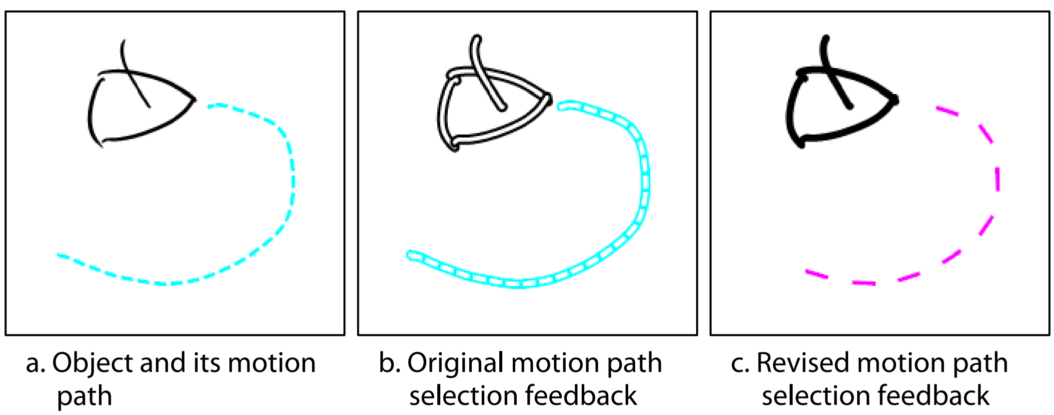


Figure 6-3: Evolution of motion path selection feedback. Originally, a selected motion path and its object were rendered in outline (*b*). The revised feedback (*c*) made the selected motion path more noticeable and rendered the object differently to distinguish it from normal selection.

6.1.3 Object Handle

The design of K-Sketch's object handle has gone through some revision since the first design iteration. The basic appearance has remained the same, but the original design omitted the context menu button below the handle, as shown in Figure 6-4a. The *Rotate Handle* function did not exist, and the *Move Handle* function was accessed by directly dragging the dark center dot. In early user tests, many novice users had trouble differentiating the *Move Handle* function from the *Move* function and accidentally moved the handle when intending to move the object. I hoped to remove this confusion by requiring an explicit command before moving the handle.

Since the toolbar was getting crowded, I decided to put this explicit command in a context menu, and this gave me a natural place to put a *Rotate Handle* function, which users were requesting. I chose to make this menu accessible through a button, because it fit well with my existing design without requiring me to add another pen mode or to use a scheme that interrupts creative flow, such as *press-and-hold*. Users unfortunately have trouble accessing this button when the handle is moved to the

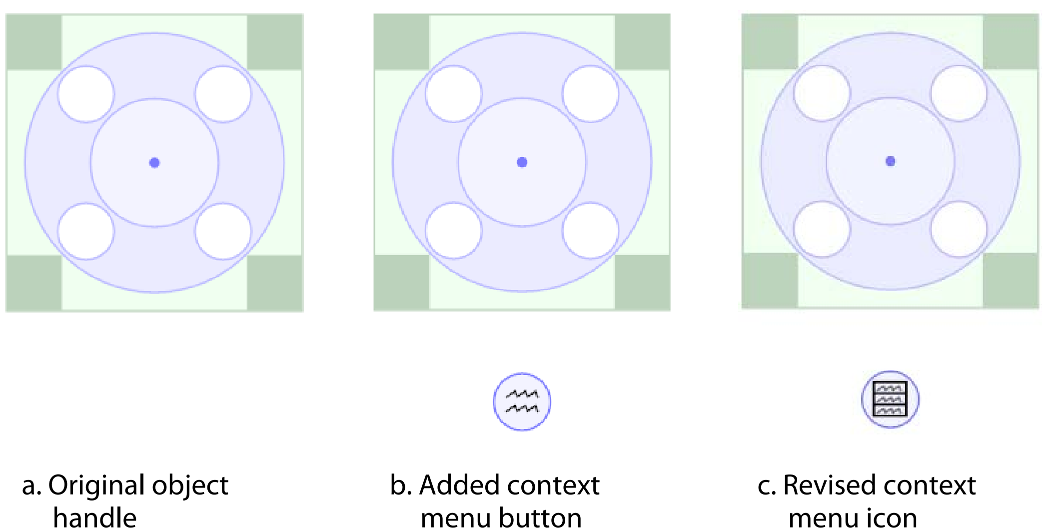


Figure 6-4: Evolution of the object handle. A context menu button was added to access less frequently used commands. The icon changed slightly in later iterations.

bottom of the screen, and I may address this in future versions by moving the button when it would be hidden off screen.

6.1.4 Recording Controls

The final state of K-Sketch's recording controls is described in Section 5.1.7 on page 138, but these also went through many revisions. Though there are important differences in their behavior, the earliest recording controls are very similar in appearance to the final controls. I merged playing and recording into a single mode called *going* to reduce the number of controls in the interface and to make mode error less likely. To further reduce the controls, I made the two modes (*going* and *stopped*) controllable through a single button.

The earliest design is shown schematically in Figure 6-5. Pressing *Go* made time advance, and new motions were recorded while all previously recorded motions played back. If the animation was empty, or if the time slider bar was positioned at the end of the animation, pressing *Go* would continue recording until the user pressed *Stop*. If the time slider bar was positioned before the end of the animation, pressing *Go* would run to the end and stop. This allowed the *Go* button to function both for recording and playing an animation.

While extremely simple in theory, this design had some practical difficulties. Timing was so imprecise that it was useful for only the simplest animations, because pauses were inserted into the timeline when the pen moved from *Go* (Figure 6-5 b)

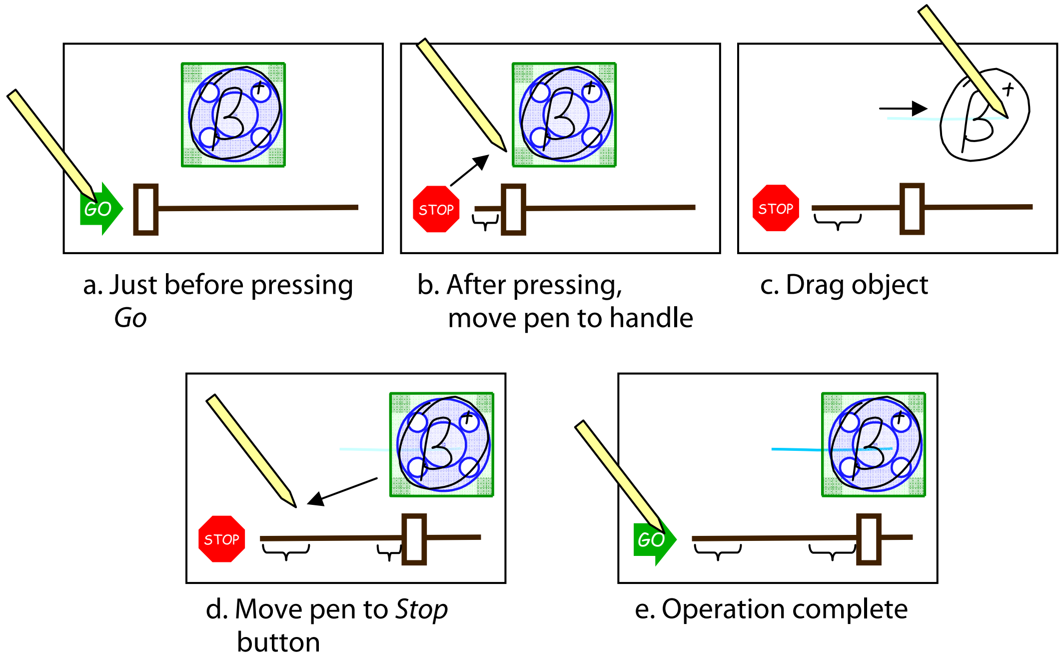


Figure 6-5: Animating motions in the original K-Sketch design. The bracketed portions of the timeline are pauses that are inserted as the pen moves to the handle (*b*) and away from it (*d*).

and back to *Stop* (Figure 6-5 d). Also, it was possible for users to record long pauses at the end of animations if they forgot to press *Stop*. Thus, mode error was a danger even with two modes. All of the following changes to the recording controls can be seen as attempts to deal with these two problems: pauses and mode error.

The first full iteration of K-Sketch introduced an *active* mode for the object handle. When the object handle was active, manipulating it would cause time to advance for the duration of the manipulation, even if the animation was not already going. It was therefore possible to remove all pauses before and after each motion by avoiding the *Go* button and using the object handle in active mode. The standard, passive mode was still present for positioning objects with the handle and moving them instantaneously in time. In the first iteration of K-Sketch, the object handle was active by default, and holding the Alternate button before manipulating the handle made it passive. I chose this, because I thought the active mode would be used more frequently. I reversed these modes after a later study, however. Study participants saw the active mode as an augmented version of the passive mode, and it felt more intuitive to do a deliberate act (like holding a button) to enter this augmented mode. After these modes were reversed, the active mode could be thought of as an object handle recording shortcut, accessed through the Alternate button.

The new recording shortcut for the object handle did not remove pauses in the case of *trace* operations, however. Hoping to find a consistent way to eliminate pauses for both traced lines and moving objects, the second iteration removed the Alternate button recording shortcut and introduced a *trigger* for the *Go* button (see Figure 6-6). If the user pressed and held the *Go* button for a short time (which varied from ½ a second to a full second in various iterations), a pen image would appear over the button. This indicated that the trigger was set, and recording would begin the moment the pen touched the screen, ending when the user pressed *Stop*. This design dealt with beginning pauses in a consistent way, though it did not deal with end pauses at all. A serious problem with this design was the need to hold the *Go* button before every recording operation, which put an unacceptably distracting break in users' creative flow. For this reason, the Alternate button recording shortcut returned shortly after, but the trigger remained to make the *trace* operation possible.

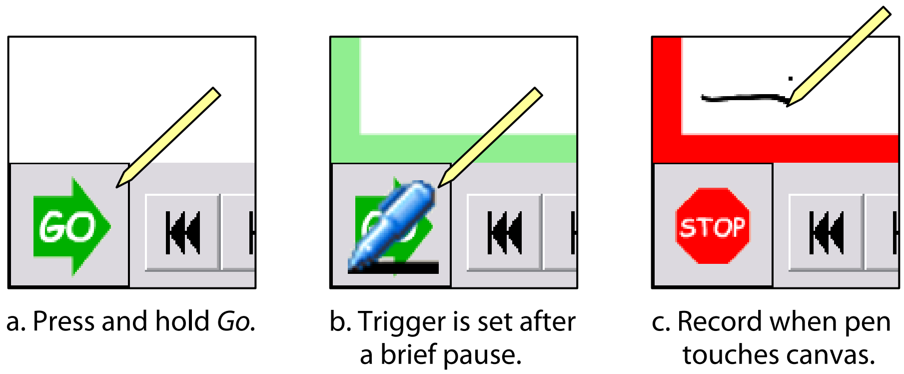


Figure 6-6: *Go–Trigger* recording controls. Pressing and holding *Go* sets a trigger, which causes recording to begin the moment the pen touches the screen for a draw or handle drag operation. Recording continues until the user presses *Stop*. An icon overlay signals that the trigger is set. Later iterations added colored borders to the canvas when the trigger is set (green) and when recording (red).

In a second attempt to remove pauses in a consistent way, I separated the recording and playing modes and made a button for each (see Figure 6-7). In this design, pressing *Record* entered a *record-ready* state (Figure 6-7b), advancing time only when the pen touched the screen (Figure 6-7c). This design made it impossible to accidentally record long periods of time with no activity. This design also allowed playback to loop to the beginning of the animation, as most media players do (see Figure 6-7e and Figure 6-7f). Also note that this version introduced colored borders around the canvas to remind the user of the current mode (green for record-ready and red for recording). I used borders similar to these in all future iterations (as in Figure 6-6b and Figure 6-6c).

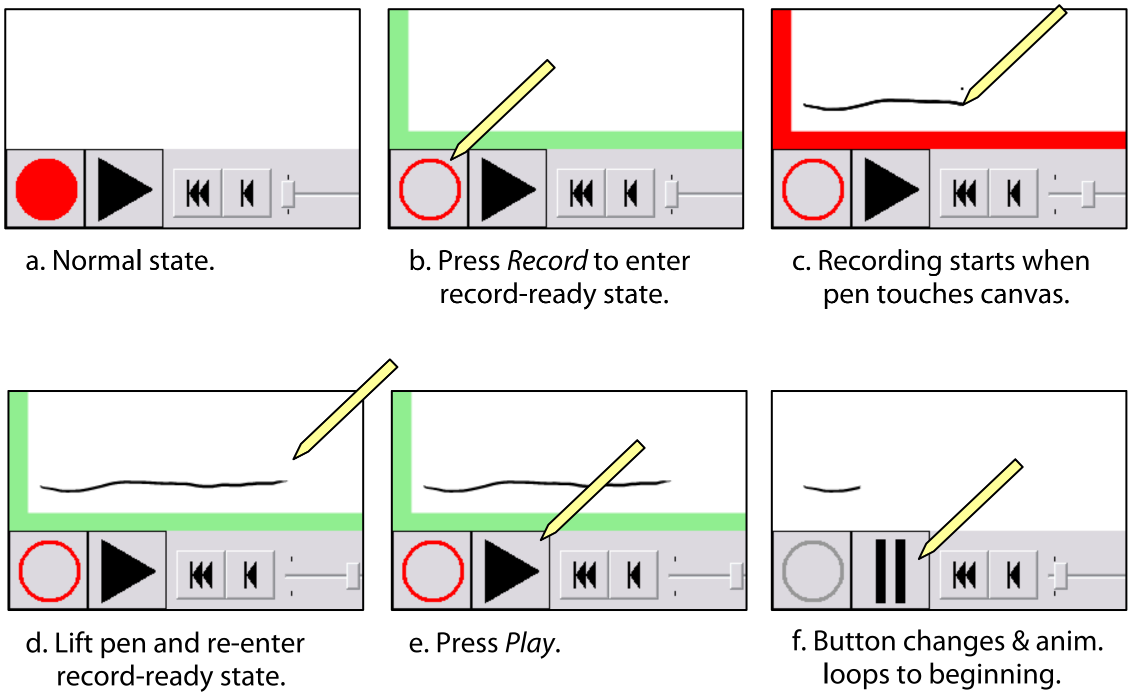


Figure 6-7: *Record–Play* recording controls. Pressing *Record* enters a record-ready state, recording every pen draw or handle drag operation and then re-entering the record-ready state. Pressing *Record* again leaves the record-ready state. Pressing *Play* runs the animation to the end or, if already at the end, rewinds and runs from the beginning.

I now had two different design directions, each with strengths and weaknesses, and I decided to run a short user study to compare them. The *Record–Play* design had similarities to existing metaphors in conventional audio and video recording devices, and it dealt with moved objects and traced lines in a consistent way. The *Go–Trigger* design (with object handle recording shortcut) appeared to be somewhat simpler in the case of moving objects, and more complex in the case of traced lines.

This short comparison study had three participants, all graduate students in the humanities. Participants completed three tasks with each design. (The procedure for this study was very similar to the procedure used in the formative study that will be presented in Section 6.3. Refer to that section for further details.) At the end of the study, *Go–Trigger* was the clear winner. There were more instances of mode error with *Record–Play,* because most tasks required users to repeatedly enter the record-ready state before animating and leave it before drawing a new object. Some users also had trouble with the concept of *Record–Play's* record-ready state, and they could not figure out how to insert pauses (where are sometimes desirable). With *Go–Trigger,* the one major problem I observed was that participants would more frequently forget to rewind the animation before adding new motions. This may have been a side-effect of experiencing creative flow. All three participants said that they preferred *Go–Trigger* because the object handle recording shortcut gave them a sense of immediate control and freed them from having to manipulate buttons on screen. For these reasons, I settled on *Go–Trigger* for many iterations.

A variant of the *Record–Play* design did briefly resurface several iterations later, however. In the formative user study presented in Section 6.3, many participants had trouble using the trigger to draw the traced line. Looking once again for better solutions, I thought perhaps that my previous *Record–Play* design failed because the record-ready state did not exist in existing tape recorder metaphors. The re-designed version (shown in Figure 6-8) had three buttons with more conventional behavior: *Play* would play to the end and stop (looping if *Play loop* was checked), *Record* would

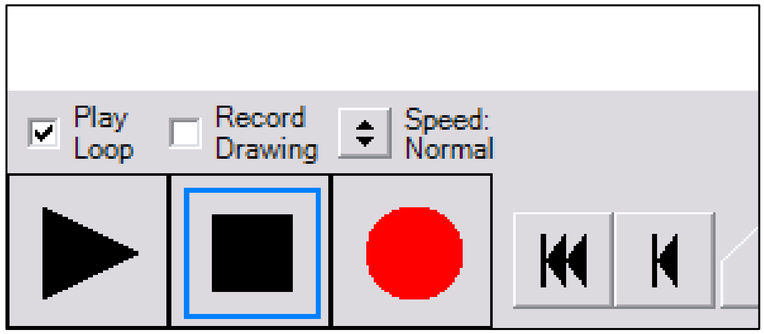


Figure 6-8: Modified *Record–Play* design. The main buttons (from left to right) are *Play, Stop,* and *Record.* Pressing each button simply enters the corresponding mode. A blue box is shown around the current mode. Recording and playback options appear above these controls.

advance time indefinitely, and *Stop* would halt the advance of time in both cases. The object handle recording shortcut was still present, and I introduced a *Record drawing* mode, which captured *trace* operations by advancing time whenever the user drew. I also used this redesign as an opportunity to introduce a speed control similar to the one found in the final version.

This redesigned version was universally rejected as too complex. Although I expected to see reduced mode confusion as compared to my previous *Record–Play* design, I actually saw more. The cause seemed to be the increased number of buttons. Many people commented that this design was visually cluttered and less pleasing than the *Go–Trigger* design, but they liked the added controls for controlling speed, looping playback, and recording drawings. While the *Record drawing* mode presents a new opportunity for mode error, is occurs less frequently because *trace* is a less frequently used operation. Subsequent iterations returned to the *Go* button, discarded

Figure 6-9: Final toolbar buttons for *Record drawings*, *Loop playback,* and *Adjust speed.*

the trigger, and moved the remaining controls to the toolbar, as shown in Figure 6-9. This design was very close to the final design.

One final problem needed to be resolved. Since the first iterations of K-Sketch, pressing *Go* when the time slider bar was at the end of the animation would advance time indefinitely. This design allowed users to record long animations with pauses intact, but users often recorded long pauses accidentally. Users appeared to think of the *Go* button as something that could not alter the animation, and they expected that pressing *Go* at the end of the animation would loop back to the beginning. Since this is the behavior of most media players, I changed K-Sketch to have this behavior. This removed the ability to record indefinitely, but that feature has not been missed.

This modification was well received, but there were two drawbacks. First, when presented with an empty animation, new users were confused by the fact that pressing *Go* seemed to do nothing. I added three seconds of blank time to every new animation so that these users would at least see the time slider bar advance when they pressed *Go*. Second, the final design has no obvious way to add pauses at the end of the animation. Fortunately, later iterations added the ability to drag timeline tic marks, enabling uses to add pauses by simply dragging the last tic mark (see Section 6.1.5).

The difficulty of designing a simple set of recording controls was perhaps the most surprising challenge I faced during this design process. The effort has paid off, however, as the final design fits well with users' intuitions, helps them remove unwanted pauses, and reduces mode error.

6.1.5 Time Slider Bar

The time slider bar underwent some revision as design iterations continued. Initially, users navigated through time with a standard slider control, as shown in Figure 6-10a. As soon as possible, however, I enlarged the slider thumb so it would be easier to grab with a pen (Figure 6-10b). Since many users were getting lost in the timeline and found it difficult to navigate to a desired time, I also added tic marks above the timeline. Tic marks indicate important events in the timeline (e.g., when objects appear or disappear or when motions begin or end). The time slider bar snapped to these tic marks, and tapping on either side of the slider thumb would go to the next tic mark. These changes significantly reduced navigation problems.

Figure 6-10b shows tic marks as they appear when no object is selected. The appearance of these *normal tic marks* has not changed since they were first introduced. When K-Sketch users select an object, however, the time slider bar changes to show *selection tic marks,* which give users more detail on the selected object. The level of

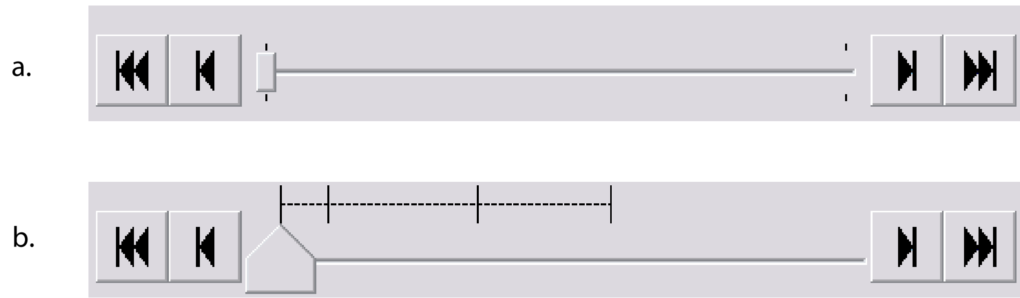


Figure 6-10: Time slider bar designs. *a:* Original design. *b:* Enlarged thumb with normal tic marks.

detail in these selection tic marks went through several revisions. The first revision is shown in Figure 6-11b. A thick horizontal line extended from the time the selected object became visible to the time it disappeared. The curvature of tic marks indicated the type of event at that time. Squares were for instant transforms, and half-circles were used for the beginning and end of timed motions. (If a second motion began at the same moment a previous motion ended, then the second motion's start tic mark would override the first motion's end tic mark, as shown in Figure 6-11b and Figure 6-11c.) Also, if a motion operated on other objects in addition to the selected object, then the tic marks for that motion were rendered in grey instead of black.

To see how this design reveals useful information about a selected object, consider the sequence of events in Figure 6-11a. Two objects move together briefly, and then split apart, after which object 1 continues while object 2 sits still for a moment and then disappears. In the original design of selection tic marks, selecting object 2 at the beginning of the animation produced the view shown in Figure 6-11b.

This view enables the user to observe that object 2's motion progresses in four phases. The grey-outlined, semicircular tic mark and the grey line on the far left show that object 2 moves with another object in the first phase. The black-outlined, semicircular tic mark that follows indicates that object 2 moves by itself in the second phase. The third semicircular tic mark points the other way, indicating that movement stops in the third phase. Finally, the thick black line disappears, indicating that object 2 disappears.

User's liked the added information in this selection feedback, but some improvements were needed. As shown in Figure 6-11c, later iterations added a faded display of normal tic marks behind selection tic marks, since coordination with other events was often necessary. Also, the distinction between grey-outlined and black-outlined selection tic marks was confusing and added little benefit. I therefore simplified selection tic marks by rendering all in black (see Figure 6-11c). Finally, because users seemed comfortable interpreting tic marks, I made them movable so that the time of events could be shifted. Most users welcomed this addition. The absence of tic marks at the moment an object appeared and disappeared was an oversight, and I corrected this in later iterations, as shown in Figure 6-11d. In future iterations I may revise the design again to make more edits possible.

The final revision to selection tic marks came when the design of the *move relative* animation operation started to take its final form. It became necessary to

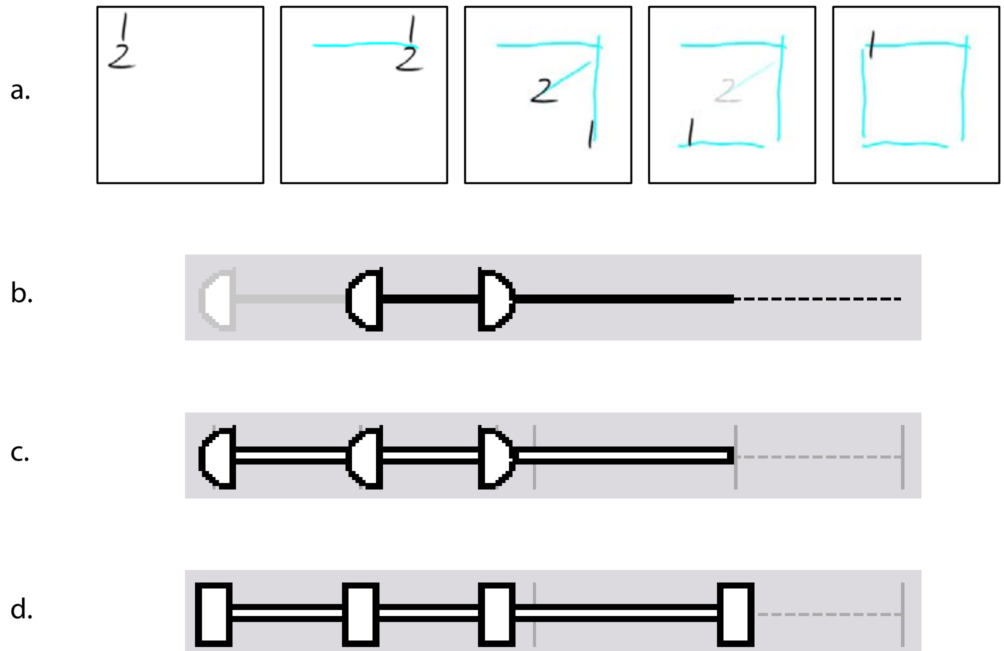


Figure 6-11: Selection tic mark evolution. *a:* A sequence of events in which two numbers begin to move in a square pattern, and then "2" separates off and disappears before "1" finishes moving. Selecting "2" at the beginning of the animation produces the following sets of tic marks. *b:* The original selection tic marks distinguished start events, end events, and events related to additional objects outside the selection. *c:* Revised tic marks distinguished only start and end events, but added normal tic marks to the background. *d:* The final design does not distinguish between start and end events and adds a tic mark for the disappearance event.

display events for *all* reference frames related to a selected object. Because of this, it was possible for motions in multiple reference frames to overlap on the same display, making it more difficult for users to map tic marks to events. To avoid information overload, I removed the distinction between instantaneous events and timed events, rendering all selection tic marks as squares, as shown in Figure 6-11d and Figure 6-12c. These later revisions also added a magenta highlight for selected motion paths, and I changed the color of tic marks for selected motions to match this, as shown in Figure 6-12d. These changes helped, but my second laboratory study showed that users can still have difficulty when the timeline becomes crowded with short, fragmented motions (see Section 7.2.5.5 on page 261). I may need to revise this feedback in the future to separate events in different reference frames onto different timelines.

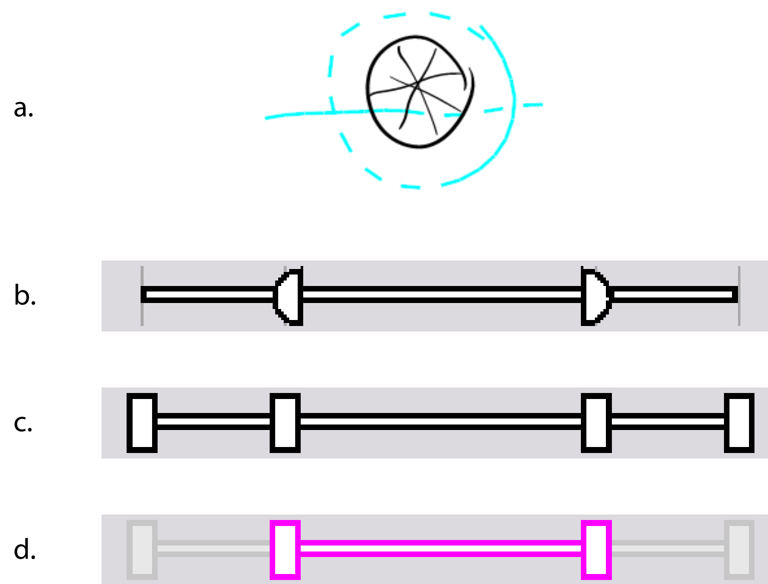


Figure 6-12: Timeline tic marks in the presence of multiple reference frames. *a:* A rolling wheel with two reference frames: one for the rotate motion and one for the translate motion. *b:* Selecting the wheel in earlier versions showed events in the lowest-level reference frame only (the rotation in this case). *c:* Selecting the wheel in the final version shows events for all reference frames. *d:* Selecting the rotation's motion path in the final version highlights the tic marks related to that motion.

6.1.6 Adding Motions in New Reference Frames

Interface optimization showed *move relative* to be an important animation operation (as explained in Section 4.3 on page 95), but finding a simple interface for adding motions in new reference frames proved challenging. As I mentioned in Section 6.1.1, my early designs made motion paths first-class objects, meaning that they could be manipulated as any other stroke. This made it possible to define new reference frames explicitly, as shown in Figure 6-13. When the user rotates the wheel in Figure 6-13b, the rotation motion is rooted in the world reference frame and defines a new reference frame for the wheel. Selecting the rotation's motion path (Figure 6-13d) and translating it (Figure 6-13e) defines a new reference frame above the rotation. (Again, I use the convention that higher reference frames are closer to the world reference frame.) The translation is rooted in the world reference frame and defines a new reference frame for the rotation.

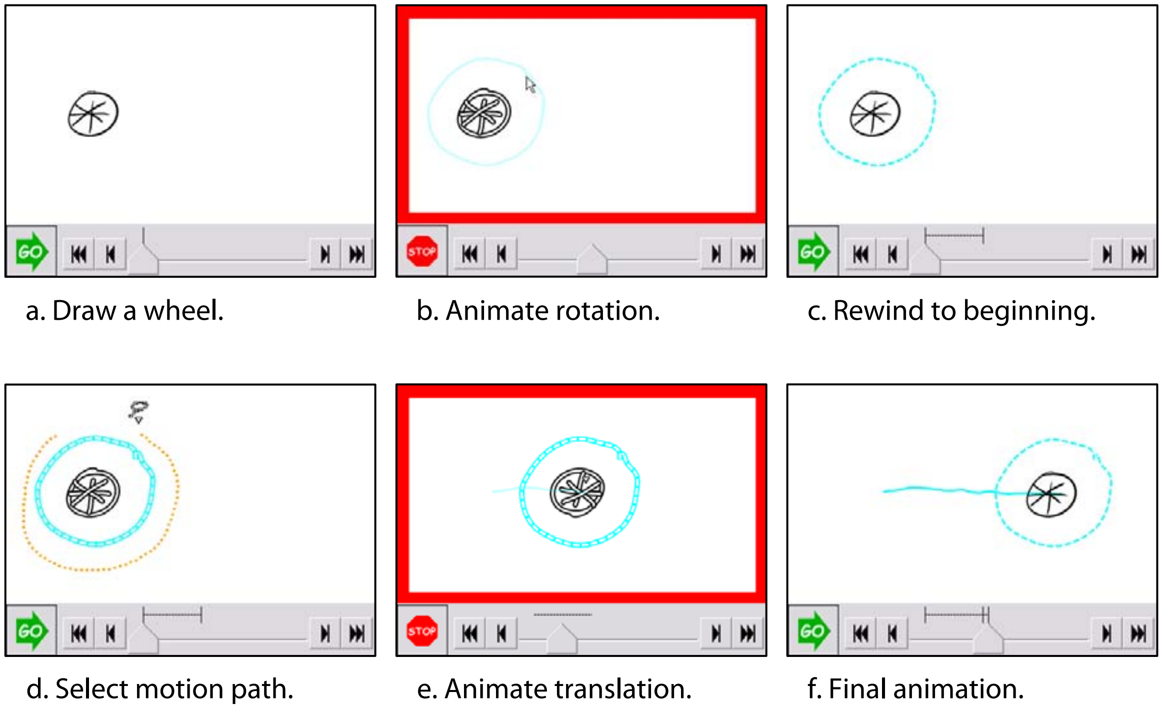


Figure 6-13: Defining new reference frames explicitly. Selecting a motion path (*d*) selects the whole reference frame. Animating it (*e*) adds a new reference frame above it. (Higher reference frames are closer to the world reference frame).

Users did not respond approvingly to this design. Selecting motions to add reference frames required advanced spatial reasoning that caused many novices to struggle. At the very least, users needed to understand which motions were closer to the world reference frame, because all motions in lower reference frames had to be defined first. In Figure 6-13, for example, the user must spin the wheel first and then translate the spin motion. Reversing the order of these operations (i.e., spinning the translate motion) would create a spiraling motion instead. Even when users did have the spatial reasoning ability to make the correct choices, they complained that it was often inconvenient to specify motions in the order required by this design. To make matters worse, users also found it hard to distinguish between selecting and moving a motion *at all times* (e.g., to shift its position slightly) and selecting it and moving it to define a relative motion.

My first redesign of this operation simplified the process of adding reference frames, but it still required users to add them explicitly. Motion paths were no longer manipulated in the same way as ink strokes. Motions could still be selected and moved, but doing so would shift the position of objects in that reference frame at all points in time. To remove the order restrictions, I gave users a way to add new motions in any reference frame at any time. I added three buttons to the object handle, as shown in Figure 6-14a. A *Context menu* button provided access to the menu shown in Figure 6-14b. A *Select next guess* button made it possible to select groups of objects or their motions, as described in Section 6.1.2. Finally, an *Add reference frame* button allowed a user to create and move between an object's reference frames.

The number displayed on the *Add reference frame* button facilitated navigation between reference frames by indicating the current reference frame. Frame 0 was the top (world) reference frame. (I planned to resolve through user testing whether it would be better to place the world reference frame conceptually at bottom or the top.) All edits in the top frame behaved normally. Pressing the button would increase the

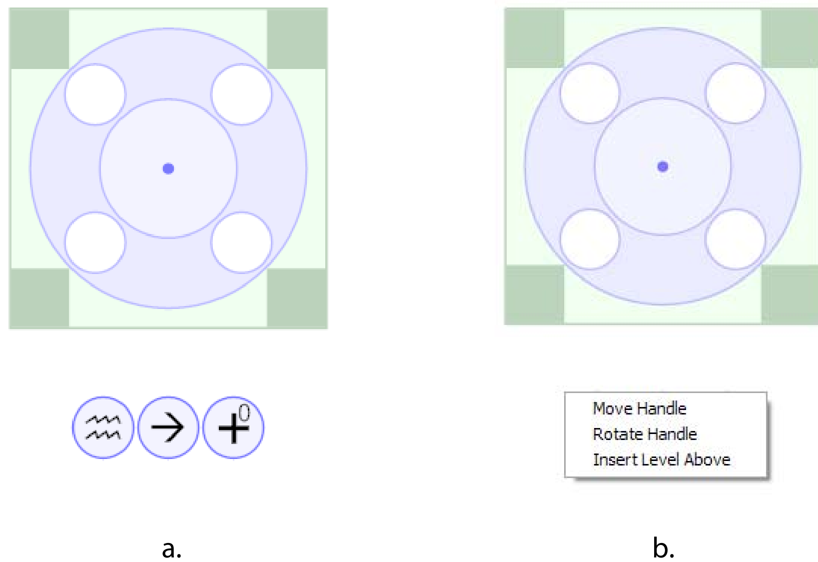


Figure 6-14: Modified object handle that removed restrictions on operation order when adding reference frames. *a:* The three buttons from left to right are *Context menu*, *Select next guess*, and *Add motion*. The number over *Add motion* indicates the depth. *b:* The context menu. *Insert level above* was later renamed *Add relative motion*.

number and move to the next level down. Edits in lower reference frames manipulated a *ghost* as described in Section 5.1.4.2 on page 125. After reaching the first empty reference frame, clicking *Add reference frame* again would cycle back to 0 (i.e., if there were n reference frames, the highest possible number was $n + 1$). This design allowed the user to add new reference frames below an existing frame, but a different mechanism was needed to add a reference frame at the top. I added an *Insert Level Above* command to the context menu (see Figure 6-14b) to make this possible.

While this design removed certain restrictions, it still required users to understand the placement of reference frames relative to the world. It also added the new danger of getting lost in the hierarchy of reference frames. It was becoming clear that explicitly managing a hierarchy of reference frames was going to be a challenging problem for novices, regardless of the interface. I then began to explore implicit management of reference frames through a suggestive interface. Soon, the interface for handling multiple reference frames started to take its final form (described in detail in Section 5.1.4 on page 120). Using heuristics, K-Sketch would attempt to put new motions into the correct reference frames. When these heuristics failed, users could execute a *fix* command, which allowed them to choose an alternative from a list of animated possibilities. As the second summative laboratory study in Section 7.2 on page 230 explains, this design has been a success.

Some minor modifications to the fix interface occurred in the final iterations. As shown in Figure 6-15a, the fix command was first accessed through the selection's context menu. In addition to a command for fixing the most recently added motion, the initial design had a command for fixing any selected motion. This added command was seldom used in practice, and I removed it for simplicity. I also created an *Add Motion* command so that users could give an explicit signal that the following motion should be in a new reference frame. Since users were comfortable fixing a motion that was placed in the wrong reference frame, however, this command seemed redundant, and I removed it. These changes made the context menu simpler (see Figure 6-15b), but the fix command was used often enough that the extra tap involved in calling up the context menu was onerous. In the final iteration, I moved the command to the toolbar (see Figure 6-15c). Note that, for a few iterations, *Select Next Guess* was stripped of the ability to select groups of strokes, and its name was changed to *Select Motions*. This capability was missed, however, and the command returned to its previous name and function in the final version, though it also moved to the toolbar.

This completes my overview of the design history of K-Sketch. Basic controls like selecting, recording, and time navigation evolved over the course of many evaluations that took place throughout the development process. In the remaining pages of this chapter, I will highlight two formative evaluations that gave me valuable design

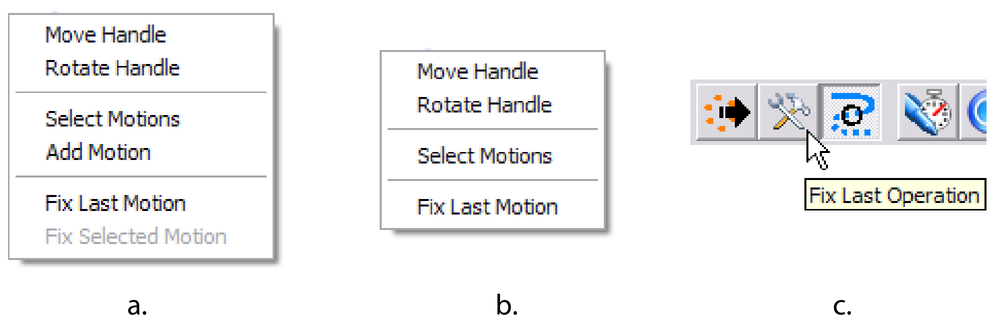


Figure 6-15: Evolution of commands for fixing and selecting motions. *a:* Fixing is possible on the most recently created motion and on a selected motion. *Add motion* command explicitly adds a reference frame. *Select next guess* becomes *Select motions*. *b:* The *Add motion* and *Fix selected motion* commands are removed because they are not needed. *c:* Both remaining motion-related commands move to the toolbar. *Fix last motion* becomes *Fix last operation*, because it now applies to paste operations in addition to motions. *Select motions* changes back to *Select next guess*.

feedback and allowed me to track progress toward my final goal of creating a fast, simple, and expressive interface.

6.2 Expert Animator Study

The K-Sketch project is motivated by the claim that no existing animation tool is simultaneously fast, simple, and expressive. I saw concrete evidence of this when I interviewed non-animators (see Section 3.2 on page 64) who said that they did not produce animations in part because existing tools were too complex and took too long to use. To verify my understanding of the animations participants described, I began using Flash to produce them. I was astonished at how long it took to produce most of these animations and felt that my interview participants' fears were justified. I later expanded this effort into a more formal study that compared K-Sketch to two existing general-purpose animation tools: Flash and PowerPoint.

In this study, I measured the time I needed to produce nine animations from my scenario library with all three tools. Seven of these animations came from non-animator scenarios, and two came from my own experience but were classified as non-animator scenarios. Given my experience with these three tools and with animation in general, this study cannot be used as an indicator of novice animator performance. However, comparing the time needed for an expert to perform a variety of tasks with three tools does shed some light on the relative speed, simplicity, and expressivity of those tools. For this reason, the study has some value as a summative evaluation, but I classify it as a formative evaluation, because the K-Sketch interface changed considerably after I ran this study.

I produced these animations on three separate occasions, first with Flash, then with K-Sketch, and then with PowerPoint. Within each group, the production took place over the course of several days or weeks. I took care to make this as fair a comparison as possible. For each tool, I used the hardware setup that I thought would allow me to work fastest: a plain desktop PC for PowerPoint, a desktop PC with a Wacom Cintiq display tablet for Flash, and a Tablet PC for K-Sketch.

I left animations rough in all three cases, as dictated by the medium; Flash and K-Sketch had rough drawings, while PowerPoint had simple vector shapes. I used the same number of moving parts and the same types of motion in all three conditions, except when the tool did not allow me to produce the exact same motion. The data I recorded includes the amount of time it took me to draw the objects in each animation. However, I factored out any time I spent planning how I would produce the animation, because including that count in my measurements would give later tools an unfair advantage. I also did my best to factor out any time I spent learning how to use the animation tool. This was most important for Flash, because my expertise with Flash was not as developed as my expertise with K-Sketch and PowerPoint.

The results of this study are presented in Table 6-1. Each animation is listed with a name and reference number that can be used to locate it in lists of scenarios (e.g., Appendix B or http://www.k-sketch.org/dissertation/index.html#apB). The length and maximum number of moving objects is shown to give some sense of each animation's complexity. Since Flash learning effects may have been present, I present animations in the order they were produced in Flash. For completeness, the table includes two animations with incomplete data (66–*Gear reduction* and 68–*Bachelor party*).

The data show that most animations took four to eight times longer with Flash than with K-Sketch, and some took significantly longer. Much of the added time in Flash went to managing timelines and frames and making timing adjustments. 54–*Contra dance* was particularly hard due to the many overlapping motion paths, which were hard to manage in Flash. 69–*Tack vs. jibe* was difficult in Flash because *orient to path* and *move relative* were difficult to use. 58–*Construction equipment tread* took a long time with Flash due to the absence of a *trace* operation. With PowerPoint, most animations took three to seven times longer than with K-Sketch, due to an awkward timeline interface that is awkward for long sequences of events. Cantor set construction took a particularly long time, because of the need to use vector graphics—in this case, a pen and eraser were much easier to work with.

#	Animation name	Length (seconds)	Max. moving objects	K-Sketch time (minutes)	Flash time (minutes)	Flash increase	PowerPoint time (minutes)	PowerPoint increase
66	Gear reduction	16	4		29			
54	Contra dance	27	5	8	151	18.9x	20	2.5x
67	Automobile accident	27	6	12	76	6.3x	41	3.4x
51	Detection of distant planets	2	34	22	153	7.0x	53	2.4x
55	Chemistry: particle collisions	7	6	6	46	7.7x	43	7.2x
56	Chemistry: rusting reaction	6	5	21	83	4.0x	67	3.2x
57	Chemistry: battery reaction	6	9	10	52	5.2x	34	3.4x
58	Construction equipment tread	13	3	5	42	8.4x	31	6.2x
59	Cantor set construction	6	1	1	4	4.0x	16	16.0x
68	Bachelor party	180	5[a]	68	347	5.1x		
69	Tack vs. jibe	6	9	4	61	15.3x	26	6.5x
	Average	*11.1*	*8.7*	*9.9*	*74.2*	*7.5x*	*36.8*	*3.7x*

[a] This example had eight objects moving simultaneously in the last 7 seconds.

Table 6-1: Data from the expert animator study. The averages in the bottom row were computed by averaging rows with full data only (excluding 66–*Gear reduction* and 68–*Bachelor party*). The average Flash increase and PowerPoint increase were computed from the average times. If I had instead averaged the data in these columns, the average Flash increase would have been 8.5 and the average PowerPoint increase would have been 5.6.

K-Sketch's highly tuned interface was paying off. Both Flash and PowerPoint had numerous features that K-Sketch lacked for adding precise details, but these features were more of a hindrance than a help in this study. K-Sketch was optimizing the right operations to make these tasks faster. I also had reason to believe that K-

Sketch would look even better if I counted planning time in future studies. Planning the time of all events is necessary in Flash and PowerPoint, but many animations can be demonstrated in K-Sketch with users' intuitive sense of time and no advance planning.

However, this study also revealed some areas for improvement in K-Sketch. Though the timing of events remained rough in K-Sketch, there were many occasions when I needed to re-record a motion in order to fix a gross timing problem. This was particularly problematic in animations that told a story (like 68–*Bachelor Party*), and I knew that the full set of scenarios contained many stories. The version of K-Sketch I was testing did not have the ability to change the timing of objects by dragging timeline tic marks, and I planned to add this ability as soon as possible. Also, the version tested did not have the ability to copy objects or motions. The ability to copy objects and motions would have been particularly helpful with animations like 51–*Detection of Distant Planets* and 57–*Chemistry: Battery Reaction,* which had many similar objects moving in repeating patterns.

This evaluation also revealed evidence of two other looming problems with K-Sketch. This version of K-Sketch had the original design for the *move relative* operation, and I could see that using this operation required a deep understanding of K-Sketch's inner workings. Only two examples in this particular set required additional reference frames (*Tack vs. jibe* and *Bachelor Party*), but I knew that the full set of scenarios contained many more. Another feature this K-Sketch version lacked was the speed control, and I found that this feature would have been particularly helpful for *Construction equipment tread*. This animation required a traced line to follow a moving object, which was very difficult unless the tread was moving very slowly. This example was a sign that the speed control would indeed be desirable in some situations.

This study gave me a good sense that K-Sketch was reaching its goals, but there were signs that improvements were needed. I decided to verify these findings by testing this version of K-Sketch on actual users as well, as described in the following section.

6.3 Formative Laboratory Evaluation

I conducted three formal user tests while the K-Sketch interface was still evolving. The second test, presented here, was the largest. It evaluated the same version of K-Sketch that was evaluated in the expert animator study of the previous section. The purpose of this test was both to evaluate the K-Sketch interface and to test the experimental procedures that I planned to use in summative evaluation. All documents and detailed data for this study appear in Appendix D.

6.3.1 Method

Each of the eleven participants in this study began by filling out consent forms and a questionnaire that asked for demographics information and about their interest in animation. Following this, I gave participants a verbal K-Sketch tutorial and then gave them the option of completing a practice task. I allowed participants to spend up to 30 minutes learning to use K-Sketch, during which they could ask as many questions as they liked. After this, I asked participants to complete three experimental tasks while asking for as little assistance as possible (except in the case of task confusion or tool malfunctions).

All three experimental tasks were based on tasks in my scenario library: 55—*Chemistry: particle collisions*, 54—*Contra dance*, and 58—*Construction equipment tread*. I chose these tasks because they covered a range of animation operations and required participants to coordinate objects in a variety of ways. Tasks were simplified to make them possible within the time limits of the study, but each contained all the motions present in its corresponding scenario. Task instructions were presented through printed words and diagrams rather than animations to be copied (see Appendix D). I did this in hopes that participants would feel free to produce animations at comfortable speeds and to use any style. I also told participants that they did not need to make the objects or motions look perfect, though they did need to represent the full sequence of events.

When the study began, I planned to vary the order in which tasks were presented. After the third participant, however, it became clear that *Construction equipment tread* was more difficult than the others, and participants needed the practice with simpler animations before moving to a more difficult one. Therefore, all but participants two and three performed tasks in the order shown in the previous paragraph (and in Table 6-2). As the study progressed, I also varied whether or not motion paths were visible (six participants) or invisible (five participants) by default.

I recorded the time participants needed to complete each task as well as any difficulties they experienced. After each task, I gave participants a *comfort-level questionnaire* that assessed their comfort showing their animation to others, as well as creating it in front of others. (Questionnaires similar to this one were used in both summative laboratory experiments presented in the following chapter.) I knew participants' comfort was likely to depend on the audience. Therefore, for each situation (showing or creating), I asked participants to rate their comfort for eight different audiences: no audience, 1 colleague in a meeting, 10 colleagues in a meeting, 1 student while tutoring, 30 students in a class, 300 students in a class, 30 professionals watching a presentation, 300 professionals watching a presentation. Responses were on a seven-point Likert scale (1=extremely uncomfortable, 7=extremely comfortable). At the end of the study, participants made general comments and filled out a questionnaire asking them how they enjoyed the experience of using K-Sketch and how likely they would be to use it again (if the hardware were available to them). These responses were on a five point Likert scale (1= not at all, 5 = very much).

6.3.2 Participants and Environment

I recruited eleven participants from around the University of Washington. As Table 6-2 shows, about half of these participants were students or researchers: three education graduate students, one biology post-doc, one biology undergraduate, and one business undergraduate. The other half did administrative work. Nine participants were women and two were men, and they ranged in age from 20 to 45 (two declined to state their age). All reported that their drawing skills were “fair” or worse, and none had significant experience with animation tools. Five participants had used the custom animation features of PowerPoint to do animation, but only one remembered doing this more than once or twice.

The most notable differences in these participants were their interest in animation and their level of programming skill. The students and researchers in this study were recruited with an advertisement seeking people interested in animation but with little experience. Not surprisingly, most of them had a medium level of interest in the topic. Those participants employed in administrative jobs were recruited with a less specific call for study participants, and they had little or no interest in animation. Programming skill in both of these groups ranged from *none* to *fair*. As the following section explains, this difference in animation interest appeared to have a strong influence on results, but programming skill did not.

All sessions took about one hour and were held in a conference room at the University of Washington. Participants performed all tasks on an HP–Compaq tc1100 Tablet PC. At the end of the study, participants were given a $20 Amazon.com gift certificate for their time.

				Learning time and task times (in minutes)			
#	Occupation	Animation Interest	Programming skill	Learning	55–Particle collision	54–Contra dance	58–Construct. tread
1	student (U): biology	medium	fair	23	5	5	6
2	student (G): education	medium	none	19	4	6	7
3	administrative	none	none	27	10	9	11
4	post-doc: biology	medium	none	17	4	7	4
5	student (U): business	medium	poor	29	4	4	6
6	administrative	none	fair	18	7[a]	6[a]	
7	administrative	low	poor	22	4	3	22
8	administrative	none	none	33	12		
9	administrative	none	fair	18	6	8	9
10	student (G): education	medium	none	24	5	6[a]	6[a]
11	student (G): education	medium	none	29	5	5	9[a]
	Average (medium interest in animation)			*23.5*	*4.5*	*5.5*	*6.3*
	Average (low or no interest in animation)			*22.3*	*6.7*	*6.7*	*10.3*

[a] Participant omitted important features or represented them poorly.

Table 6-2: Data from the formative laboratory evaluation. Averages are separated by level of interest in animation. The averages for low or no interest omits participants 6 and 8, who could not complete all tasks in time. Because of this, those averages are deceptively low.

6.3.3 Results

Table 6-2 shows a summary of the results from this study. Since task times varied considerably between participants (with two failing to complete all tasks), I looked for statistical signs that task time correlated with interest in animation or with programming skill. I averaged the time across tasks for each participant and assumed that both interest in animation and programming skill were scalar variables. I omitted participants 6 and 8 from this analysis, because they did not have full data. Bivariate (Pearson) correlation found a significant correlation between average task time and interest in animation (r=-.863, p<.01), but no significant correlation between average task time and programming skill (r=-.078, p>.05). Because of this, I separate my discussion based on participants' interest in animation.

The six participants with an interest in animation took four to seven minutes to complete each of 55–*Chemistry: particle collisions* and 54–*Contra dance*. 58–*Construction equipment tread* proved more difficult because many users forgot how to perform the *trace* operation. Participants with interest in animation finished the construction tread task within nine minutes, but two did not represent the traced line correctly. Participants with little or no interest in animation tended to take longer on all three tasks, and two failed to complete some tasks. It seemed that many of these participants had never before reasoned about time as a dimension, often unable to distinguish between sequential and simultaneous events. While their task times indicate that they performed poorly, these participants quickly learned important ideas about animation and were delighted by the experience.

In spite of the roughness of their animations, participants were fairly comfortable sharing their animations with others. Since most participants' responses to the comfort-level questionnaires followed the pattern of becoming less comfortable as the audiences became larger and less familiar, I simply averaged their responses. Across all participants and audiences, the average comfort showing animations to others was 4.82 (with seven being the most comfortable). Participants were also fairly comfortable creating animations in front of others, with an average response of 4.77.

Producing animations was seldom a problem-free experience, however. Many problems concerned selection and manipulation of the object handle. Seven participants experienced some problems selecting objects, and fragments of moving objects were left behind on several occasions. Motion paths were also selected accidentally by four of the six participants for whom motion paths were visible. Three participants repeatedly grabbed the center dot of the object handle accidentally, moving the handle when they meant to translate the object.

Many participants also struggled with issues related to time. Many participants struggled with the process of converting the static descriptions of animations into a series of events in time. Three participants were unable to produce correct animations without prompting on at least one occasion. Seven participants had trouble with the recording controls, making the animation go by mistake on several occasions and recording long periods of blank time. Four participants forgot how to initiate a *trace* operation, and needed to be reminded.

Finally, participants requested several features that would help them manage time. Three requested the ability to delete a motion path as a way to erase a motion. Four requested the ability to change the time of recorded events, with one specifically requesting the ability to move timeline tic marks. Also, two requested the ability to slow down the animation to a more manageable speed. I took these comments to heart and included all three features in subsequent versions of K-Sketch.

In spite of these difficulties, participants were very positive about the experience of using K-Sketch in the closing questionnaire. On a five-point scale, the average response to the question "How did you enjoy the experience of using K-Sketch" was 4.64. When asked how likely they were to use K-Sketch again if it were available (on the same scale), participants' average response was 4.36. Two said that they would use it to illustrate specific scientific processes to students. One said she would have language students make movies and then explain the action in a foreign language.

6.3.4 Discussion

The fact that novice participants were able to quickly create fairly complex animations within 30 minutes of encountering K-Sketch was a good sign that K-Sketch's interface was both fast and simple. Even participants who struggle with the whole concept of animation were able to learn something by attempting to create animations through demonstration. However, given the time constraints of laboratory experiments, I resolved in future experiments to recruit only participants who expressed some interest in animation.

Many of the problems participants experienced were similar to those I experienced in the expert animator study of Section 6.2. The selection problems were more evidence that I needed to provide better grouping mechanism (which became the *Select next guess* button) and that I needed to prevent the accidental selection of motion paths. Accidental shifting of the object handle relative to selected objects was a cause for taking the *Move handle* functionality out of the handle itself and placing it in a context menu. Problems with the recording controls and the *trace* operation were motivation for the ill-fated redesign shown in Figure 6-8 on page 176 and the subsequent changes described in Section 6.1.4. There was also further evidence that I needed to add features like movable tic marks, a speed control, and cut, copy, and paste operations for motion paths.

The experience I gained while running this study pointed to the need for several additional changes to my study procedures. I clearly could not assume that all animation operations were equally easy to learn. In later studies, I either chose tasks of equal complexity or presented tasks to users in approximate order of increasing complexity. It was also clear that static descriptions of tasks were too hard for users to interpret. While I thought participants would naturally produce animations at a comfortable speed, some participants still needed to change the speed as they worked. In later studies, I decided to present tasks as animations that were to be copied.

All of the evaluations presented in this chapter played an important part in the evolution of the K-Sketch interface. The two evaluations I have presented here in detail, the expert animator study and the formative laboratory evaluation, also foreshadow the summative evaluations found in the following chapters. In all of those evaluations, participants are novice animators: people with interest in animation but little or no experience. Also, the summative evaluations put K-Sketch to much more challenging tests, either direct comparison with another animation tool or usage in the field to accomplish real work.

7 Laboratory Evaluations

Recalling the central thesis of this dissertation, my summative evaluation needs to demonstrate that an informal, sketch-based animation system designed through analysis of usage scenarios will allow novices to create a wide variety of animations quickly and easily and to use animation in new ways. There is a two-fold contribution here: the K-Sketch system itself and the method used to design it. My evaluation approach is to put this system into the hands of novice animators and show how their experience reflects on my design method and thesis.

My summative evaluation began with two laboratory studies that show how K-Sketch reaches its goals of being a fast, simple, and expressive animation interface. These studies validate both the interface optimization results and the interface design and implementation. Since the contributions of this project lie more in the design method used to integrate interaction techniques, neither study evaluates low-level interaction behavior. Instead, novice animators are asked to perform complete tasks based on scenarios in my library. The first study, described in Section 7.1, specifically evaluates the speed and simplicity of K-Sketch's interface. The second study, described in Section 7.2, uses fewer participants and four times as many tasks, allowing me to evaluate K-Sketch's expressivity.

Both laboratory studies need another animation tool as a basis for comparison. Since the contributions of this work lie in design methods rather than interaction techniques, comparing K-Sketch to a simplified version of K-Sketch that omitted certain techniques would provide little value. The ideal test would compare two animation interface designs produced by two groups working independently under controlled conditions. Such a study would be difficult to control adequately and would require more time and resources than are available for this dissertation. Instead, I compare K-Sketch to existing animation tools for novices designed with different methods.

While some animation *sketching* tools for novices exist, none support enough animation operations for a fair comparison. This is not surprising, since most were designed for isolated evaluations of a few interaction techniques. Instead, I chose tools for comparison that were likely to perform well under the conditions of each study. The first study compares against Microsoft PowerPoint [106], which has powerful *custom animation* features for quickly adding animation *effects* to a presentation with little learning. The second compares K-Sketch to a more general animation tool for novices called The TAB Lite [48], which allows users to express a variety of animations using a different set of animation operations from K-Sketch.

7.1 Laboratory Study 1: Speed and Simplicity

My first study sought to evaluate speed and simplicity by comparing K-Sketch to PowerPoint, a tool that seeks to help its users understand animation quickly enough to add simple animations to presentations in short order. PowerPoint supports most of the animation operations supported by K-Sketch, though it is missing *trace* and *orient to path*, and its support for *set timing* is very limited. PowerPoint also provides nearly 200 other animation operations (i.e., effects) for quickly making eye-catching presentations, but it is easy to ignore about 85% of these if desired. It is therefore a good example of a formal, general-purpose animation tool for novices.

Like most office productivity tools, PowerPoint encourages formality. This evaluation thus highlights the differences between formal and informal tools when performing rough tasks. In spite of my attempts to give each tool the most favorable conditions for fast learning and animating, K-Sketch significantly outperformed PowerPoint. Though this experiment does not exercise all of the capabilities of either K-Sketch or PowerPoint, it shows how my design method made K-Sketch faster and simpler than conventional tools.

7.1.1 Method

This was a within-subjects comparison of PowerPoint and K-Sketch. After filling out a demographics form, participants did a practice task followed by two experimental tasks with one tool, and then repeated the process for another tool. The independent variables were thus the tool used and the task performed. All forms and documents used in this study appear in Appendix E.

7.1.1.1 Procedure

Just as in the formative laboratory experiment of Section 6.3 on page 194, the experimental tasks were based on scenarios in my library: 55–*Chemistry: particle collisions* (Task A, shown in Figure 7-1) and 54–*Contra dance* (Task B, shown in Figure 7-2). To keep each participant's time commitment under four hours, I chose tasks that required participants to learn only four animation operations: *translate, appear, disappear,* and *set timing (*or *interpolate,* in the case of PowerPoint). Tasks were shortened from their original scenarios, but both contained all the essential motions in those scenarios. Also, the graphics of these tasks were simplified to make them easy to create with both PowerPoint and K-Sketch.

After filling out a consent form and a demographic questionnaire, participants completed an 8 to 10 page written tutorial that showed them step-by-step how to complete a practice task with one tool. The practice task was designed to teach participants everything they needed to know to complete the experimental tasks as quickly as possible. Both participant and experimenter were free to interact in any way to facilitate learning during this phase. When the practice task was completed, participants were allowed to keep this tutorial as a reference for later tasks. During the experimental phase, participants were asked to avoid seeking help unless they

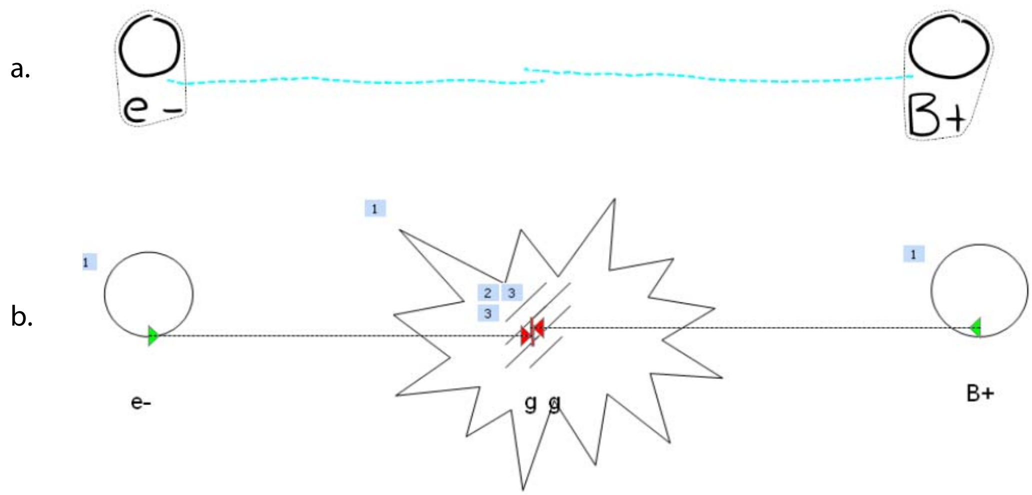


Figure 7-1: Laboratory study 1 Task A (55–*Chemistry: particle collisions*) animations produced by participant 10. *a:* The participant first created this version with K-Sketch in 5 minutes. *b:* Confused by PowerPoint's interface, the participant later took 24 minutes to produce this version with PowerPoint.

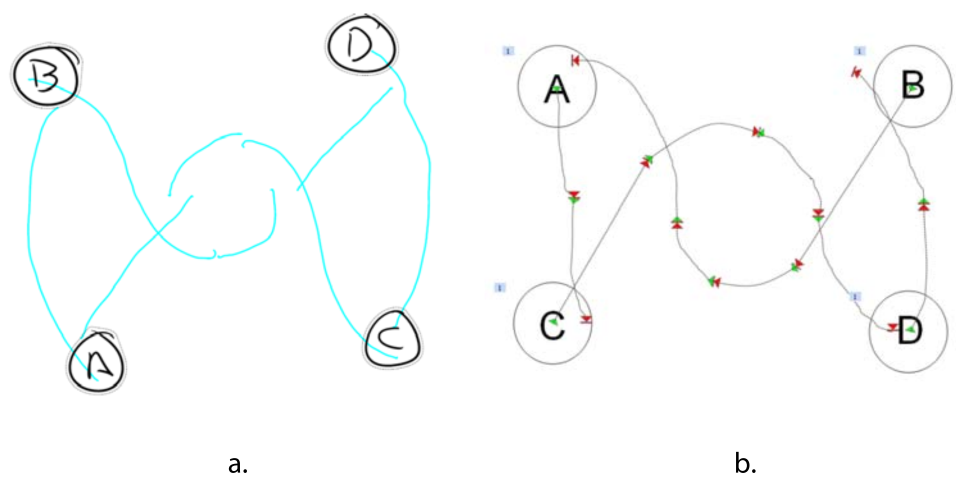


Figure 7-2: Laboratory study 1 Task B (54-*Contra dance*) animations produced by participant 5. *a:* The participant first created this version with K-Sketch in 5 minutes. *b:* Struggling to perfect the motion paths under time pressure, the participant later took 15 minutes to produce this version with PowerPoint but was less satisfied with the results.

were stuck. When they did ask for help, I intervened in gradual stages: avoiding it at first, then pointing to relevant sections of the tutorial, then making verbal suggestions or short answers to questions, and finally intervening with open discussion of confusing points. After each task was finished, participants filled out a comfort-level questionnaire. After performing both experimental tasks with one tool, participants completed the practice task and experimental tasks for the other tool. The order of tools and tasks was counterbalanced.

All tasks were presented as animations that participants needed to create. Participants viewed task animations in the QuickTime player, allowing them to replay and scan through the animation as much as they wished. These animations were formal so that participants could quickly form a clear mental picture of the task they needed to perform. Task animations were produced with a third animation tool (Flash). This made it difficult to reproduce objects or their motions exactly with either tool, giving neither tool an advantage. To create a sense of time pressure, participants were instructed before each task to complete the task as quickly as possible. The instructions stressed that participants did not have to make the animations look perfect, and that it was more important for them to work quickly than it was to reproduce objects or their motions precisely. They were required, however, to keep the sequence of events the same.

7.1.1.2 Measures

The primary dependent variable was the time to complete each task. As in the formative laboratory study of Section 6.3, I also asked participants to fill out comfort-level questionnaires after each task. These questionnaires asked participants to rate their comfort showing their animation to others or creating it while being watched by others. As before, I asked these questions for eight different audiences, shown in Table 7-1. Responses were on a seven-point Likert scale (1=extremely uncomfortable, 7=extremely comfortable).

I measured three variables that depended only on the tool and not the task. After using each tool, participants took the NASA TLX cognitive load self assessment [65]. This is a 100-point scale that measures cognitive load on six axes: mental demand,

Number	Audience
1	No one[a]
2	A single colleague in a private meeting
3	Ten colleagues in a private meeting
4	A student I am tutoring
5	30 students in a class I am teaching
6	300 students in a class I am teaching
7	30 professionals during a business presentation I'm making
8	300 professionals during a business presentation I'm making

[a]For comfort showing, this was listed as "myself (keep in private notebook only). "

Table 7-1: Audiences used in comfort-level questionnaires. On a seven-point Likert scale, participants rated how comfortable they were showing their animation to each of these groups and creating it while being watched by each of these groups.

physical demand, temporal demand, performance, effort, and frustration level. I also asked participants for subjective feedback on both tools at the end of the experiment. One question asked how easy it was for them to work with both tools (7-point Likert scale, 1=very easy, 7=very hard), and the other asked how fast they were at operating both tools (7-point Likert scale, 1=very fast, 7=very slow).

Finally, I counted the occurrences of three types of incidents: normal help, bugs, and task confusion. Since normal help was given in stages, these incidents were of three types: tutorial references, requests for help (with no response), or actual interventions (whether requested or not). When participants encountered confusing bugs or appeared not to understand the task, I intervened immediately.

7.1.2 Participants

Since my formative laboratory experiment showed that participants performed better if they had some interest in animation, my recruiting poster called for people who were "interested in creating animation but have never done so." A total of 18 participants participated in the study. Of these, two were discarded from the final analysis because they could not complete all tasks in the time available. Partial demographic information on the remaining 16 participants appears in Table 7-2 (with full demographic information appearing in Appendix F). Seven of these participants were men, and nine were women. Thirteen were between 18 and 45 years old, and 3 were above 45. Nearly half were students, and the others worked as artists, technology professionals, teachers, or dental assistants.

#	Occupation	Sex	Drawing skill (1 = none, 5 = excellent)	Expertise (1 = complete beginner, 7 = expert)			Animation experience (hours)
				Computers	PPT /Word	Tablet PCs	
1	(none given)	F	3	5	2	1	0
2	illustrator and designer	F	5	5	1	1	0
3	comp. bio. grad student	F	2.5	6	4	1	0
4	engineering manager	M	3	7	4	1	0
5	artist	M	5	4	2	1	0
6	student (many subjects)	M	3	5	4	2	6
7	mechanical eng. student	M	3	5	2	2	15
8	dental assistant	F	3	5	2	1	0
9	marketing director	F	2	4	3	1	5
10	medical student	M	2	4	3	1	5
11	teacher, software consult.	F	4	7	6	3	32
12	baker, student	F	4	5	2	1	0
13	technology manager	F	3	5	6	1	10
14	student (many subjects)	F	3	4	3	1	0
15	photographer	M	3	6	2	4	0
16	electrical eng. student	M	3	6	5	1	8
	Averages		*3.2*	*5.2*	*3.2*	*1.4*	*5.1*

Table 7-2: Laboratory study 1 participant demographics. Note that the demographic questionnaire did not ask about experience with Microsoft Word, but some participants mentioned that they had such experience, and I group it here with PowerPoint (PPT) experience.

Participants rated their drawing skills using one of five ordinal responses: none, poor, fair, good, or excellent. Two participants (both artists) rated their skill as excellent, two were good, nine were fair, two were poor, and one was undecided between fair and poor. Participants also rated their expertise with computer tools on a 7-point scale (1=complete beginner, 7=expert). Participants were somewhat experienced with PowerPoint (*M* = 3.19, *SD* = 1.52) and very inexperienced with Tablet PCs (*M* = 1.44, *SD* = 0.89).

Participants were also asked how often they would create animations if they had the time and skill to do so. Responses were ordinal on five levels: at least once a day (n = 0), once a week (n = 3), once a month (n = 3), once a year (n = 6), or not sure (n = 4). This shows that most participants' desire to create animations was sporadic, which is another obstacle to gaining expertise with complex animation tools. When asked why they wanted to create animation, the purpose varied, including new works of art and animations for a company web site. Participants had little or no experience with PowerPoint's custom animation interface (11 had none, 5 had 1–5 hours) or other animation tools (11 had none, 5 had 5–30 hours). Finally, on a 7-point Likert scale (1=disagree strongly, 7=agree strongly), participants were asked if they were discouraged from creating animations by the time required (*M* = 5.23, *SD* =1.54) or by the complexity of animation tools (*M* = 4.43, *SD* = 1.83). The responses indicate that speed is the primary concern for these participants, but simplicity is also desirable.

7.1.3 Environment

All sessions took place in a private room of the Computer Science and Engineering building at the University of Washington. K-Sketch tasks were completed on a

Compaq tc4200 tablet PC, and participants had the option of using a bezel button or a hand held remote control for the Alternate button. The version of K-Sketch being used was slightly different from the final version: The *Select next guess* and *Fix last operation* toolbar buttons were missing, and grouped objects were outlined with a dotted line (see Figure 6-1 on page 166). PowerPoint tasks were completed on a Dell desktop PC running PowerPoint 2003. This desktop had a second monitor on which participants viewed the task animations that they were to produce. Sessions took between 2½ and 4½ hours to complete.

7.1.4 Results

When analyzing the results of this study, I looked for statistical significance whenever possible. I was concerned that some variables would not be normally distributed, particularly time measurements and incident counts, so I subjected each variable to a one-sample Kolmogorov-Smirnov test. When this test gave no significant result ($p>.05$), I retained the null hypothesis that the data was normally distributed and used a parametric test, such as an ANOVA or t-test. If this test showed a variable to have a distribution significantly different from normal, I say so in the following analysis and either analyze the natural logarithm of that variable (if it is normally distributed) or use a non-parametric test, such as Friedman or Wilcoxon.

I use an alpha of .05 for most tests of significance. However, some measures reported here are closely related. For example, both subjective measures were collected at the same time, as were measurements of comfort sharing and comfort showing. To account for the relationships between these variables when testing for significance, I used a Bonferroni corrected alpha of $.05/n$, where n is the number of related tests. For ease of interpretation, I report p-values only and note when significance depends on the use of a corrected or uncorrected alpha.

The following analysis is divided into sections for clarity. The first examines variables related to K-Sketch's speed: task time and the subjective speed measure. The second examines variables related to K-Sketch's simplicity: practice time, the subjective simplicity measure, and the NASA TLX cognitive load self assessment. I then examine participants' comfort sharing animations with others and creating them in front of others. Finally, I look at incident counts and other results. Complete data for this study can be found in Appendix F.

7.1.4.1 Speed-related Measures

Table 7-3 shows the data collected for the two speed-related measures, task time and subjective speed. The task time was not normally distributed (z=1.55, p<.05), which is a common problem in studies where some participants finish tasks quickly while others experience difficulty and take much longer. To address this problem, I computed the natural logarithm of task time, which was normally distributed (z=.95, p>.05).

I analyzed this log of task time in a two (K-Sketch vs. PowerPoint) by two (Task A vs. Task B) within-subjects ANOVA. This test found a significant difference between tools ($F_{1,15}$=113.23, p<.001), but no significant difference between tasks ($F_{1,15}$=.001, p>.05) and no interaction between tool and task ($F_{1,15}$=1.65, p>.05).

	Task time (minutes)				How fast? (1–7, 1 = fastest)	
	K-Sketch		*PowerPoint*		*K-Sketch*	*PowerPoint*
Participant	A	B	A	B		
P1	20.50	15.93	27.35	21.45	3	5
P2	5.10	10.82	40.67	60.00	5	7
P3	4.25	15.53	13.08	15.58	3	4
P4	9.67	3.17	15.83	13.87	2	4
P5	7.10	4.63	18.13	15.40	3	5
P6	3.50	3.97	24.03	12.45	1	3
P7	3.17	5.10	12.90	13.52	2	4
P8	7.48	4.38	15.05	16.25	3	5
P9	6.55	3.78	16.53	13.75	2	5
P10	5.10	5.47	23.90	10.93	2	5
P11	11.12	4.02	14.03	10.97	3	3
P12	4.37	12.18	25.88	59.37	3	6
P13	4.92	9.05	20.90	9.80	4	4
P14	3.17	3.62	15.17	10.22	1	5
P15	3.45	6.85	13.83	20.03	6	4
P16	3.33	5.02	13.40	12.05	1	3
Mean	**6.76**[a]		**19.57**[a]		**2.75**	**4.50**
Std. dev.	**4.30**[a]		**12.24**[a]		**1.39**	**1.10**

[a] Means and standard deviations for task times are shown, but my statistical analysis was conducted on the natural log of task time: K-Sketch M = 1.76 (5.81 minutes), SD = .526; PowerPoint M = 2.85 (17.35 minutes), SD = .451.

Table 7-3: Laboratory study 1 speed-related measures. The differences in both task time and subjective speed are significant. Task times are separated for Task A (55–*Chemistry: particle collisions*) and Task B (54–*Contra dance*). However, means for these tasks are not separated, because no significant difference was found between tasks.

Because the difference in tasks was not significant, Table 7-3 separates means and standard deviations for tools but not for tasks. The table shows that the time to complete experimental tasks was about three times lower with K-Sketch than with PowerPoint. Since the subjective speed measure was taken only once per tool, I analyzed this variable with a paired sample t-test and found that K-Sketch felt significantly faster than PowerPoint (t_{15}=-4.87, p<.001).

7.1.4.2 Simplicity-related Measures

All three measures of simplicity were analyzed with paired sample t-tests that found significant differences between tools (see Table 7-4). Participants needed about half as much time to complete the practice task with K-Sketch as with PowerPoint (t_{15}=-5.69, p<.001). Participants also thought K-Sketch felt easier than PowerPoint (t_{15}=-4.67, p<.001). The NASA Task Load Index was about two times higher for PowerPoint than for K-Sketch (t_{14}=-5.44, p<.001).

	Practice time (minutes)		How easy? (1–7) (1 = easiest)		NASA-TLX (1–100)	
Participant	*KSK*	*PPT*	*KSK*	*PPT*	*KSK*	*PPT*
P1	26.17	30.00	1	4	70	68
P2	33.75	54.00	3	5	- [a]	- [a]
P3	31.32	32.72	3	4	35	66
P4	24.25	34.63	2	4	30	39
P5	19.93	43.22	2	4	45	70
P6	26.17	61.33	1	5	6	27
P7	30.57	39.30	1	4	19	54
P8	28.05	48.12	2	5	22	73
P9	20.82	45.35	1	4	22	55
P10	25.75	36.87	2	6	51	91
P11	26.13	30.40	2	4	35	53
P12	21.70	70.42	2	5	11	56
P13	25.88	43.00	4	3	44	54
P14	19.33	34.68	1	4	29	60
P15	36.20	51.45	6	3	52	40
P16	25.70	40.87	1	4	8	44
Mean	**26.36**	**43.52**	**2.13**	**4.25**	**32.0**	**56.7**
Std. dev.	**4.80**	**11.36**	**1.36**	**0.78**	**18.1**	**15.9**

[a] This participant did not complete the final questionnaire necessary to compute cognitive load.

Table 7-4: Laboratory study 1 simplicity-related measures. All differences are statistically significant. *Legend:* KSK: K-Sketch; PPT: PowerPoint.

	Comfort showing (1-7, 1 = extremely uncomfortable)				Comfort creating (1–7, 1 = extremely uncomfortable)			
	K-Sketch		*PowerPoint*		*K-Sketch*		*PowerPoint*	
Participant	A	B	A	B	A	B	A	B
P1	4.75	5.88	5.63	5.50	5.00	4.75	4.75	5.13
P2	6.13	3.25	4.63	3.13	6.13	3.25	4.75	3.00
P3	4.38	3.13	4.88	5.50	4.50	4.25	4.00	4.00
P4	6.25	6.25	6.25	5.88	6.25	6.25	6.25	5.50
P5	5.17	6.00	3.13	4.63	6.57	7.00	3.25	2.75
P6	7.00	7.00	6.50	7.00	7.00	7.00	6.38	7.00
P7	5.63	5.63	5.38	5.75	5.63	5.38	5.38	4.88
P8	7.00	7.00	5.13	3.75	7.00	7.00	2.25	3.13
P9	4.88	4.75	5.13	2.25	5.00	4.75	2.88	2.00
P10	2.38	4.25	2.50	4.88	4.13	4.38	1.88	2.75
P11	5.25	5.50	4.88	6.25	5.50	6.50	3.88	4.25
P12	4.25	3.38	3.13	1.88	4.25	3.75	2.50	3.13
P13	3.13	4.25	5.88	5.88	4.63	4.38	5.13	5.88
P14	5.75	6.38	4.63	4.75	6.25	6.25	3.75	4.75
P15	5.00	4.63	6.00	5.00	4.50	4.13	3.75	3.38
P16	6.13	5.00	6.38	4.63	6.75	6.50	5.38	3.50
Mean	5.17		4.89		**5.46**		**4.10**	
Std. dev.	1.25		1.30		**1.14**		**1.35**	

Table 7-5: Laboratory study 1 comfort showing and creating animations. Both measures were averaged across all eight audiences to produce this table. The mean and standard deviation for comfort creating are shown in bold, because the difference is statistically significant. Measures are separated for Task A (55–*Chemistry: particle collisions*) and Task B (54–*Contra dance*). However, statistics for these tasks are not separated, because no significant differences were found between tasks on comfort showing or comfort creating.

7.1.4.3 Comfort Showing and Creating

To compare participants' comfort showing their animations to others and creating them in front of others, I averaged each participant's responses across all eight audiences and analyzed them with a two by two within-subjects ANOVA. Table 7-5 shows the full data used in this analysis. Oddly, participants were slightly *more* comfortable showing their K-Sketch animations to others, but this difference was not significant ($F_{1,15}$=.82, p>.05). Also, there were no significant differences between tasks on comfort showing animations ($F_{1,15}$=.22, p>.05), nor was there any interaction between tool and task, ($F_{1,15}$=.30, p>.05). I then looked for significant differences in each audience separately. Since only audience 6 (300 students) was normally distributed, I used non-parametric Friedman tests to compare all four tool–task combinations (K-Sketch–Task A, K-Sketch–Task B, PowerPoint–Task A, PowerPoint–Task B) for each audience. None of these tests found significant differences in comfort showing animations (all p-values >.05).

However, participants were significantly more comfortable creating animations in front of others with K-Sketch than they were with PowerPoint ($F_{1,15}$=14.83, p=.002). There were no significant differences between tasks on comfort creating animations ($F_{1,15}$=.62, p>.05), and no interaction between tool and task ($F_{1,15}$=.51, p>.05). Again, I examined each audience separately using Friedman tests, because only audiences 5 (30 students) and 7 (30 professionals) were normally distributed. These tests found significant differences in comfort showing animations for all eight audiences, even with a Bonferroni corrected alpha of .05/8 = .00625 (all p-values <.006).[3] Audience 3 (10 colleagues) stood out as the most significant (p=.0001)

7.1.4.4 Incidents

As mentioned earlier, I counted incidents of normal help and help due to bugs and task confusion. Counts for normal help incidents of all three types (tutorial references,

	Tutorial references				Help requests				Help interventions			
	K-Sketch		*PowerPoint*		*K-Sketch*		*PowerPoint*		*K-Sketch*		*PowerPoint*	
Participant	A	B	A	B	A	B	A	B	A	B	A	B
P1	-	-	1	1	-	-	-	1	2	-	3	1
P2	-	-	-	-	-	-	2	1	-	-	2	3
P3	-	-	-	-	-	-	-	-	-	-	-	-
P4	-	-	-	-	-	-	-	-	-	-	-	-
P5	-	-	-	-	-	-	2	-	-	-	1	-
P6	-	-	-	-	-	-	1	-	-	-	-	-
P7	-	-	-	-	-	-	-	-	-	-	-	-
P8	-	-	-	-	-	-	-	-	-	-	-	-
P9	-	-	-	-	-	-	-	1	-	-	-	-
P10	-	-	3	-	-	-	-	-	-	-	1	-
P11	-	-	-	2	-	-	-	-	-	-	-	-
P12	-	-	-	6	-	-	-	-	-	-	-	5
P13	-	-	-	-	-	-	1	-	-	-	-	-
P14	-	-	-	-	-	-	-	-	-	-	-	-
P15	-	-	-	-	-	-	-	-	-	-	-	-
P16	-	-	-	1	-	-	-	-	-	1	-	-
Total	0		14		0		9		3		16	

Table 7-6: Laboratory study 1 incidents where participants sought or received normal help. To aid readability, only trials with non-zero incident counts are shown. The total number of help incidents for K-Sketch was 3 and the total for PowerPoint was 39.

[3] To demonstrate conclusively that the differences found by the Friedman tests were caused by tools and not by tasks, I would have to perform pair wise comparisons of each tool–task combination for each audience (48 tests). I omit it for the sake of brevity.

Incidents	Reason for seeking or receiving help
K-Sketch	
2	Confusion: sequential vs. concurrent motions
PowerPoint	
18	Confusion: effect speed, delay, or ordering options
5	Confusion: adding an effect vs. changing an effect
4	Confusion: general trouble adding an effect
4	Forgot how to add or configure a *disappear* effect
2	Closed custom animation window and could not reopen
2	Problems defining a curve

Table 7-7: Laboratory study 1 normal help incidents that occurred twice or more.

help requests, and help interventions) are shown in Table 7-6. Help was rarely needed with K-Sketch. Only one person (participant 1) needed help while using K-Sketch, and one other (participant 16) asked for an animation operation that was not available (*interpolate*), but was content to work without it. There were many help incidents of all three types with PowerPoint, a total of 39. Table 7-7 lists the most common reasons why participants sought help.

Table 7-8 shows counts of bug and task incidents. Since K-Sketch is prototype software, it is not surprising that the number of bugs encountered was larger. The most common bugs encountered are listed in Table 7-9. The most common bug occurred when users held the Alternate button and dragged the object handle to animate an object. If users pressed the Alternate button a split-second too late, K-Sketch would appear to record the motion without actually recording it. Participants were also confused by another bug that made it impossible to move timeline tic marks when an object was selected. Interventions due to task confusion were rare. Most were reminders that their animation need not look perfect (PowerPoint only) or prompts to fix motions that did not match the task definition.

It is interesting to note that most of the problems I observed in the formative laboratory study of the previous chapter (see page 200 of Section 6.3.3) did not occur in this study. Participants in this study did not struggle to convert static task descriptions into animations, because they viewed animated task instructions instead. This helped make task interventions rare. Also, participants in this study had very little trouble with K-Sketch's recording controls, thanks to my redesign after the formative study. Finally, I observed no trouble with the *trace* operation in this study, because neither of the tasks in this study required it. (Participants seemed comfortable with *trace* in the second laboratory study, indicating that my redesign of this operation after the formative study was successful.)

Statistical analysis of incident counts was difficult because occurrences were sparsely distributed. I summed all normal help incident counts together to simplify the analysis and increase the probability of detecting a significant effect. All three types of incidents (helps, bugs, and task interventions) failed my normality test (all p-values <.001), so I analyzed them with non-parametric tests.

Friedman tests did indicate a significant difference among number of help incidents for tool and task combinations (χ^2=10.77, N=16, df=3, p<.05), but not among bugs (χ^2=3.91, N=16, *df*=3, *p*>.05) or task interventions (χ^2=2.57, N=16, *df*=3, *p*>.05). I then used Wilcoxon tests to verify that the significant effect found by the Friedman test on help incidents was due to the tool being used. The number of help incidents was significantly lower with K-Sketch than PowerPoint for both Task A (*z*=-2.21, *p*=.027) and Task B (*z*=-2.023, *p*=.043), but only if I use an alpha of .05 instead of a Bonferroni corrected alpha of .05/2 = .25. Thus, there is statistical

	Bug interventions				Task interventions			
	K-Sketch		*PowerPoint*		*K-Sketch*		*PowerPoint*	
Participant	A	B	A	B	A	B	A	B
P1	1	-	-	-	1	-	-	1
P2	2	1	-	-	-	1	-	1
P3	-	-	-	1	-	-	-	-
P4	-	-	-	-	-	-	-	-
P5	-	-	-	-	-	-	-	-
P6	-	-	-	-	-	-	-	-
P7	-	-	-	-	-	-	-	-
P8	2	-	-	-	-	-	-	1
P9	-	-	-	-	-	-	-	-
P10	1	-	1	-	-	-	1	-
P11	2	-	2	1	-	-	-	-
P12	-	4	-	-	-	-	-	-
P13	-	2	-	-	-	-	-	-
P14	-	-	-	-	-	-	-	-
P15	-	-	-	-	-	-	-	-
P16	-	-	-	-	-	-	-	-
Total	15		5		2		4	

Table 7-8: Laboratory study 1 incidents where participants encountered a confusing tool bug or needing clarification of the task. To aid readability, only trials with non-zero incident counts are shown.

Incidents	Bug encountered
K-Sketch	
5	Get recording feedback without actually recording
4	Can't move timeline tic marks when object is selected
2	Strange behavior after shrinking motion to zero time with tic marks
2	Other objects don't move when demonstrating motion
PowerPoint	
3	Effects later in list do not override earlier effects as they should
2	Some objects disappear during preview animations

Table 7-9: Laboratory study 1 bug incidents that occurred twice or more.

evidence that users need less help when using K-Sketch, but some would not consider it strong evidence, because it does not correct for multiple tests of significance.

I also looked for correlations between task time and number of incidents. For this correlation to be statistically valid, I had to sum each variable across all tool and task conditions for each participant. After summing these variables, both task time and number of task interventions were found to be non-normally distributed, so I used non-parametric Spearman tests to check for correlations. The correlations I found were not significant if I used a Bonferroni-corrected alpha of $.05/3 \approx .017$. If I had used an uncorrected alpha, help incidents would have shown the strongest correlation with task time (ρ=.575, p=.02), while bugs (ρ=.517, p=.04) and task incidents (ρ=.507, p=.045) would have been somewhat weaker. Thus, there is some statistical evidence that the number of incidents correlates with task time, but only if I do not correct for multiple tests of significance.

7.1.4.5 Other Results

At the end of the study, participants were asked what they liked or disliked about K-Sketch and PowerPoint. The most common comments appear in Table 7-10. Many

Occurrences	Comment
K-Sketch	
likes	
6	Liked the feel of paper, drawing, or using the pen
5	Simple or easy to learn
4	Felt natural or intuitive
3	Liked the timeline
2	More control over timing
2	Liked seeing things move simultaneously while animating
dislikes	
11	Needs vector drawing or editing tools
3	Disliked roughness of motions or drawings
2	Disliked using both hands at once (Alternate button)
PowerPoint	
likes	
4	Liked the similarity to other software tools
4	More vector drawing and editing tools
3	Liked the precision of graphics
2	Felt logical or structured
2	More control over timing
dislikes	
7	Felt complicated or overly technical
4	Felt time consuming or tedious
3	Felt inflexible or too structured
2	Needed to plan ahead of time
2	Disliked the timeline

Table 7-10: Laboratory study 1 participant comments that occurred twice or more.

participants liked using the pen and found K-Sketch's interface to be simple or intuitive. However, eleven participants commented that K-Sketch needed more tools for creating and editing precise graphics as in PowerPoint. When asked what they liked or disliked about PowerPoint, four participants said they liked PowerPoint's similarity to tools they were familiar with, and four said they liked the presence of precise graphical tools. On the other hand, many said that PowerPoint felt complicated, tedious, or inflexible.

It is also worth mentioning that six participants used the timeline tic marks in K-Sketch to move events in time. This is notable, because participants were not required to do this in any task. The feature was mentioned in the practice task tutorial, but these participants' decision to experiment with it was spontaneous. The use of tic marks for this purpose was so prevalent that tic mark bugs became a common cause of bug interventions (see Table 7-9).

7.1.5 Discussion

These results show that K-Sketch's simple, informal interface has strong benefits. Experimental tasks took an average of one-third the time with K-Sketch. K-Sketch also required less cognitive load, and participants felt that it was easier and faster. The simplicity of K-Sketch's interface also meant less practice time was needed before participants could perform tasks. Participants' comments show that they appreciate this. One commented, "The flow is more natural, easy to think of." Another said, "So easy to use… pretty intuitive." K-Sketch's gave participants the right capabilities at the right time, while PowerPoint, in contrast, frequently frustrated users' plans. Another participant commenting on K-Sketch said, "I didn't have to look through several menus and submenus to hunt for commands to do what I wanted. I could just do what I wanted."

7.1.5.1 Benefits of Demonstration and K-Sketch's Timeline

Many participants took naturally to demonstrating animation and to K-Sketch's timeline, but they were confused by PowerPoint's timeline and its many menu options for adding and coordinating effects. Figure 7-1 on page 208 shows an example of a task that took longer with PowerPoint primarily because the user was confused by PowerPoint's timeline. As one user commented about K-Sketch, "The time slider was great. Being able to see the motions at the same time I was drawing also was really nice." Another said, "I felt like I had more control over when things were happening over time." Many participants expressed delight when they demonstrated a motion for the first time, exclaiming, "That's cool!" or, "Oh my God, that's so much easier!" The relative absence of requests for help was a sign that they took easily to K-Sketch's conceptual model. User's grasp of this model can also be seen in the number of participants that manipulated events in K-Sketch's timeline, even without prompting.

7.1.5.2 Benefits of Informality

The goal of informal interfaces is to help users defer the specification of details, and the results of this study give strong evidence of this benefit. Participants were asked repeatedly to work fast and avoid making objects or motions perfect, but they still spent time perfecting PowerPoint animations. The impulse to perfect in PowerPoint seemed involuntary, and some users even seemed aware of this fact. Unable to stop fretting about details while animating Task B (54–*Contra Dance*), one participant laughed and said, "It probably doesn't matter how big the letters are, or does it?" In contrast, the roughness of K-Sketch animations was liberating and probably contributed to fact that K-Sketch felt easier and faster. Figure 7-2 on page 208 is an example in which a user completed a K-Sketch animation quickly, but had difficulty letting go of imperfections in his PowerPoint animation.

It is also noteworthy that the extra time participants spent on PowerPoint tasks was not sufficient to make them more comfortable showing their animations to others. This is a reminder that the benefits of formality are wasted when working under serious time pressure. (Also note that participants' mediocre self-rating of their drawing skills did not cause them to value their K-Sketch animations less than their PowerPoint animations.)

Finally, these results hint that K-Sketch's informality could make spontaneous animation a practical medium in collaborative environments. Participants were significantly more comfortable creating animations in front of others with K-Sketch, and their cognitive load was significantly lower. They also commented that using K-Sketch "was like drawing on paper," and that it was "intuitive" and "natural." These are important qualities for a system that supports spontaneous collaboration.

This laboratory study has shown that K-Sketch has made significant strides toward allowing novices to animate faster and more easily. PowerPoint also attempts to make animation fast and simple for novices, but its interface is designed for precise tasks, and it cannot compete on rough tasks such as these. Because K-Sketch leaves animations in a rough state, its interface is simpler, and it does not pressure users to add precision when it is not necessary. The strength of K-Sketch, then, lies partially in its informality. The next study demonstrates that K-Sketch's strength lies also in its expressivity, which is due to a careful choice of animation operations.

7.2 Laboratory Study 2: Expressivity

My second laboratory study sought to evaluate the expressivity of K-Sketch by evaluating novice performance on a much larger set of tasks. The first laboratory study effectively demonstrated the benefits of informality, so for this study's baseline I chose to compare against a less formal animation tool. I chose The TAB Lite [48], because it also balances speed, simplicity, and expressivity; it was designed for novice animators; and it could be configured easily to work with a pen. I begin this section by describing the TAB Lite, then describe my experimental method, participants, and environment, and finally present results and discussion.

7.2.1 The TAB Lite

As shown in Table 7-11, the TAB Lite provides a different set of animation operations from K-Sketch. The TAB Lite does not include *set timing, move relative, trace, copy motion,* or *orient to path.* However, it adds three operations not included in K-Sketch: *interpolate, deform,* and *morph.* The TAB Lite also includes a fourth operation not present in my analysis, called *swing*. This operation can be thought of as a special form of *interpolate* that reduces the number of key frames that must be defined for certain swinging motions. Finally, the TAB Lite has somewhat limited support for two operations: *translate* (straight paths only) and *rotate* (less than 180° for each in between), meaning that some motions must be broken into pieces. My

Provided by both tools	*Provided by K-Sketch only*
translate: straight path	set timing
rotate: less than 180°	move relative
scale	trace
appear	copy motion
disappear	orient to path
	translate: any path
Provided by neither tool	rotate: any angle
repeat motion	
define cels	*Provide by the TAB Lite only*
physically simulate	interpolate
move forward/back	morph
move limb	deform
	swing (new)

Table 7-11: Animation operations provided by K-Sketch and the TAB Lite. The *swing* operation provided by the TAB Lite was not included in my earlier analysis and can be thought of as a special case of interpolate for certain swinging motions.

analysis predicted that these differences would allow the TAB Lite to support only 32% of the animations in my library[4] compared to K-Sketch's 81% (see Table 4-2 on page 101). This led me to predict that K-Sketch would prove itself to be a more expressive tool.

The TAB Lite's key-frame animation approach makes it very different from K-Sketch, but similar to most popular animation tools. Instead of a timeline, however, the TAB Lite has a filmstrip control that shows every frame of the animation. This control is simple for short animations, but with 12 frames per second, it can quickly become hard to manage. Users animate by using this control to define key frames and generate in-between frames. The TAB Lite in betweens by transforming one key frame into another without requiring users to define separate tracks for each moving object. This often makes animation easy, but the system generates bad in betweens when it fails to detect the motions that users desire. The filmstrip control has no slider for quickly moving between frames, but users can activate an *onion skin* that overlays past and future frames to give a sense of how the animation changes over time.

The study described here demonstrates K-Sketch's superior expressivity in two ways. First and foremost, it shows the benefits of using interface optimization to carefully choose a set of animation operations. K-Sketch's operation set allows it to

[4] This prediction does not take the *swing* operation into account.

significantly outperform The TAB Lite in many tasks while matching it in all others. Second, this study shows how K-Sketch's simple timeline and simple approach to animation work together to provide better support for creative improvisation than traditional key frame approaches.

7.2.2 Method

This study was very similar in structure to the study described in Section 7.1: a within-subjects experiment with tool (K-Sketch or The TAB Lite) and task as independent variables. The forms and documents used in this study are very similar to those used in the previous study. All can be found in Appendix G.

7.2.2.1 Procedure

Each participant completed four sessions of approximately three hours each. In the first two sessions, participants completed as many tasks ask they could with one tool, and then switched to the other tool for the last two sessions. I could not perfectly counterbalance the order of tools, because I recruited an odd number of participants. Four participants used K-Sketch first and the TAB Lite second, and three used them in reverse order. This gave the TAB Lite a slight advantage, because working with the first tool may have given participants a familiarity with the tasks and with animation in general that helped them work faster with the second tool.

To stress the expressivity of both tools, I chose nine experimental tasks that reflect the variety found in my scenario library. These tasks, shown in Table 7-12, were shortened versions of tasks chosen randomly from the library. (Some of the originals are described in Appendix B). Two examples of these tasks are shown in Figure 7-3 (Task C) and Figure 7-4 (Task F). Table 7-12 also shows the animation operations that could be used to accomplish each task. When shortening animations

Task	Based on	Description	Operations used
A	39–*Cobalt*	Face with mouth opening while stars move behind	translate: straight, scale, appear, disappear
B	15–*Man watching sun*	Sun moves in arc as man turns to watch	appear, disappear, translate: any[k]
C	66–*Gear reduction*	Gears turn each other (8 times)	copy motion[k], rotate: any[k], interpolate[t], repeat motion
D	25–*Ant walking*	Ant swings his legs back and forth	rotate: < 180°, set timing[k], swing[t], move limb
E	33–*The trickster mummy*	Juggler coordinates tossing of two balls	rotate: < 180°, translate: any[k], physically simulate, move limb
F	26–*Lady with torch tumbling*	Tumbling figure holding a flickering torch	rotate: < 180°, set timing[k], move relative[k], orient to path[k], translate: any[k], interpolate[t], morph[t], deform[t], repeat motion, define cels
G	46–*Plane story*	Plane flies around and destroys another plane	translate: straight, rotate: < 180°, scale, appear, disappear, move relative[k], trace[k]
H	30–*Faerie adventures*	Figure moves & flaps its wings, camera zooms	rotate: < 180°, scale, set timing[k], move relative[k], translate: any[k], repeat motion
I	51–*Detection of distant planets*	Star emits waves as a planet revolves around it	appear, disappear, copy motion[k], translate: any[k], repeat motion

[k] This operation was provide by K-Sketch only.
[t] This operation was provided by the TAB Lite only.

Table 7-12: Study 2 experimental tasks with descriptions and operations used. Not all operations must be used to produce each animation, but each could be used in at least one approach.

for this study, I took care to include features that used each of these animation operations. Every animation operation is accounted for in this set except for *move forward/back,* which is provided by neither K-Sketch nor the TAB Lite and is used in only two animations in the library. The tasks therefore come close to spanning the full range of tasks present in the scenario library.

Because participants needed to learn more concepts than they did in the previous study, I presented experimental tasks in the order shown in Table 7-12, which

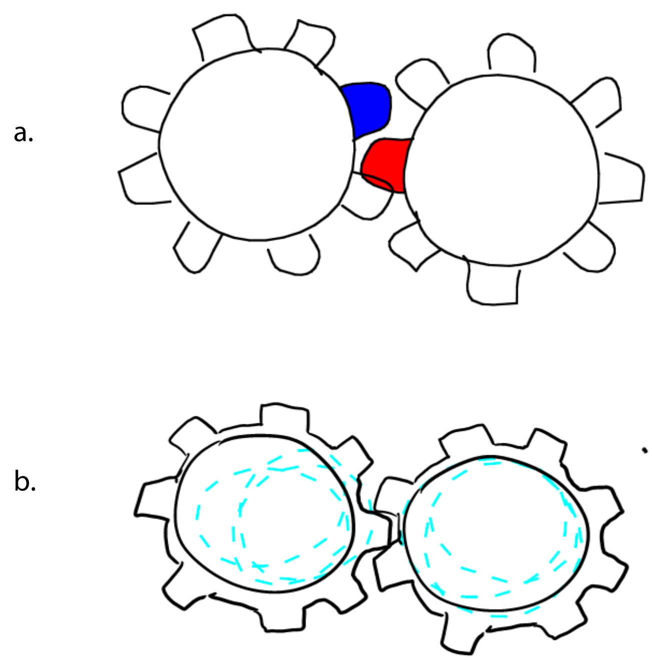
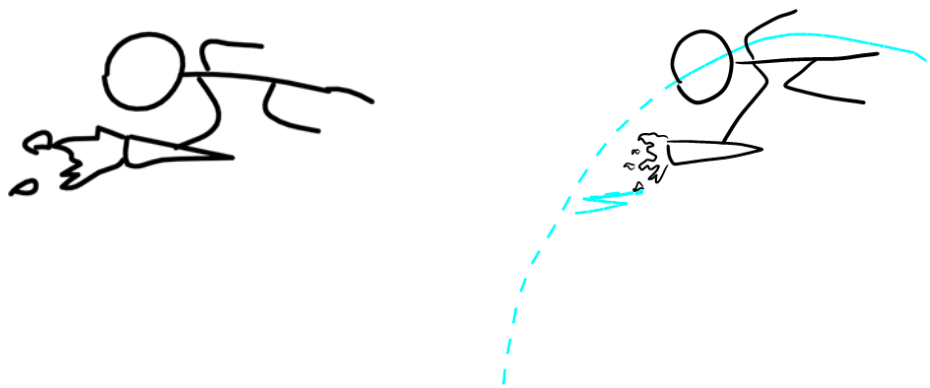


Figure 7-3: Laboratory study 2 Task C (39–*Cobalt*) animations produced by participant 4. *a:* This participant created this version in the TAB Lite in 42 minutes (plus an additional 14 minutes after a task intervention). He found it very difficult to keep track of the gears' key frames and added colored tabs to help. *b:* The participant later created this version in K-Sketch in 8 minutes. The gears' teeth did not stay aligned throughout the animation, but he was equally satisfied with the results.



a. b.

Figure 7-4: Laboratory study 2 Task F (26–*Lady with torch tumbling*) animations produced by participant 5. *a:* The participant first created this version in the TAB Lite in 8 minutes using 8 key frames. *b:* The participant later produced this version in K-Sketch in 2 minutes using a single *orient to path* motion and a single *scale* motion.

reflected their increasing complexity. This allowed me to introduce concepts gradually for both tools and avoid overwhelming participants. Training occurred before each experimental task (except for Task B, which followed immediately after Task A), and consisted of a discussion of concepts followed by a practice task. I chose not to give participants formal, written tutorials so that I could tailor the training to each participant's needs. I also coached participants on how they could improve their performance after each experimental task.

As before, participants were free to ask questions during training phases, but were asked to avoid asking questions during experimental phases unless they got stuck. Participants were given written manuals for both tools, but they rarely referred to these manuals unless I directed their attention to them. Again, I gave help in stages, avoiding it at first, then pointing to relevant sections in a manual, then making verbal suggestions or short answers to questions, and finally intervening with open discussion of confusing points.

7.2.2.2 Task Instructions

Task instructions were again presented as animations to be copied, and participants were again reminded that they did not need to make their animations look perfect. I was concerned that presenting formal animations to participants would pressure them to perfect their animations, so I made these animations rough. To avoid giving an advantage to either tool, I produced the K-Sketch task instruction animations with K-Sketch and the TAB Lite task instruction animations with the TAB Lite. This had the unfortunate side effect of biasing participants toward representing task features with approaches that I used, but I did observe participants using different approaches.

I was also concerned that participants would omit features from animations due to confusion or fatigue. To level the playing field as much as possible, I gave participants a written list of vital features for each task. I also instructed them to keep the sequence of events the same and keep the speed of moving objects "within about 50%" of what they saw. Some participants had trouble interpreting these instructions, creating an undesirable mental burden. As a compromise, I did not intervene with task clarifications until participants announced that they were finished. A side effect of this policy was that participants sometimes needed to make extensive and time-consuming edits to their animations before tasks were complete. (Note, I did answer any questions participants asked to clarify tasks, but I did not record these questions. The *task incidents* I recorded include interventions only. I call them "task incidents" merely to be consistent with other types of incidents I counted.)

7.2.2.3 Measures

Once again, the time to complete each task was the primary dependent variable. However, my policy of avoiding task incidents until participants said they were finished made it possible to measure two times for each task, each with different caveats. First, I measured the time it took for the participant to complete the task *as they understood it,* (i.e., the time until the first task incident). This variable captures the speed at which users can convert their mental images into lower-quality animations. Second, I measured the full time participants needed to make their animations meet my specifications. This variable captures the speed of converting mental images to somewhat higher-quality animations, because it includes editing time. The difference between these measures is not drastic, but it is significant enough to illuminate some important properties of editing with K-Sketch.

Because training was highly tailored to each participant, I did not measure training time as a dependent variable. I did measure the number of tasks completed over two sessions with each tool, which gives a rough sense of tool complexity. To save time in this study, I also refrained from measuring cognitive load using NASA TLX. I did record participants' subjective assessment of how fast (1=very fast, 7=very slow) and how easy (1=very easy, 7=very hard) each tool felt on a 7-point Likert scale. Instead of taking these measures at the end of the study, however, I took them after each task.

As before, I also asked participants to fill out comfort-level questionnaires after each task. These questionnaires asked participants how comfortable they were showing their animations to various audiences and creating them while being watched by various audiences. The audiences were nearly identical to those used for the previous study (see Table 7-1 on page 211), except for audience 1 (no one). I handled this case differently, because calling this an "audience" seemed awkward. In the comfort showing questions, I instead asked participants to rate their satisfaction with their animation (on a 7-point Likert scale, 1=extremely unsatisfied, 7=extremely satisfied). In the comfort creating questions, I removed this audience altogether.

Finally, I counted incidents of normal help, help due to bugs, and task confusion. I did not count manual references separately, because they were virtually nonexistent. I also did not count unanswered requests for help, because I encouraged participants to think aloud more often, and it was difficult to separate actual questions from participants' internal dialog.

7.2.2.4 Demonstrating Expressivity

Before diving into the outcomes of this study, it is helpful to consider how the measures presented in the previous section will reflect the expressivity of these tools. My assumption in this research has been that an animation tool's expressivity is related to the fluidity with which a wide variety of animations can be created. The expressivity of a tool could therefore be reflected in any measures of speed and simplicity, including both measures of time and subjective measures (e.g., how fast and how simple). It is possible, however, to tease apart the effects of a tool's expressivity from the effects of its speed and simplicity, because a more expressive tool would prove to be fast and simple with a wider variety of tasks.

The goal of the present study is to demonstrate K-Sketch's expressivity using measures that are similar to those of the laboratory study of Section 7.1. I do this by making three assumptions about the relative speed, simplicity, and expressivity of K-Sketch and the TAB Lite. Then, I make a series of predictions based on those assumptions. If these predictions turn out to be true when checked against measurable data, then there is some evidence that my assumptions were valid.

My first assumption is that K-Sketch and the TAB Lite have roughly equivalent speed. This is reasonable, because the use of sketching was a major factor in the speed improvements of the previous study, and both tools use sketching in this study. My second assumption is that K-Sketch and the TAB Lite have roughly equivalent simplicity. This is reasonable, because both are designed for novices and both require participants to learn a similar number of animation operations (K-Sketch has 10 and the TAB Lite has 9). My third assumption is that K-Sketch is more expressive than the TAB Lite. This consistent with my earlier analysis showing that K-Sketch's animation operations allow it to support 81% of my scenarios, while the TAB Lite's animation operations allow it to support only 32%.

These assumptions allow me to predict the relative performance of K-Sketch and the TAB Lite for each experimental task. If the speed and simplicity of both tools are the same, then the relative performance will depend entirely on each tool's

Task	Based on	Prediction
A	39–*Cobalt*	***No difference:*** straightforward with both tools.
B	15–*Man watching sun*	***K-Sketch faster:*** ability to *translate* along curved paths gives K-Sketch a slight advantage.
C	66–*Gear reduction*	***K-Sketch faster:*** gears hard to draw with both tools, but *rotate* more than 180° gives K-Sketch a big advantage.
D	25–*Ant walking*	***No difference:*** leg motion is easy either with K-Sketch's *set timing* or the TAB Lite's *swing*.
E	33–*The trickster mummy*	***K-Sketch faster:*** ability to *translate* along curved paths gives K-Sketch a slight advantage with balls' motions.
F	26–*Lady with torch tumbling*	***K-Sketch much faster:*** K-Sketch has three advantages—*translate* along curved paths and *move relative* help with tumble, and *set timing* helps with flicker.
G	46–*Plane story*	***No difference:*** straightforward (but long) with both tools.
H	30–*Faerie adventures*	***K-Sketch faster:*** The TAB Lite's *interpolate* is better for zooming, but K-Sketch has three other advantages—*translate* along curved paths helps with figure translation, *move relative* and *set timing* help with flapping.
I	51–*Detection of distant planets*	***K-Sketch faster:*** K-Sketch's *copy motion* helps with waves, and *translate* along curved paths helps with revolution.

Table 7-13: Study 2 predictions for task outcomes.

expressivity. This implies that the experimental measures of this study should show significant differences between tools when the operations provided by one tool would give it an advantage. Furthermore, no significant differences between tools should be found when neither tool has an advantage. If the predictions turn out to be true, then there is some evidence that K-Sketch is more expressive than the TAB Lite.

Table 7-13 lists my prediction for each task. These predictions are based on the capabilities of each tool (see Table 7-11) and the requirements of each task (see Table 7-12). In six of the nine tasks, I hypothesize that K-Sketch would have an edge over the TAB Lite. There are only two tasks for which the TAB Lite's operation set might give it the advantage: Task C (66-*Gear reduction*) and Task D (25–*Ant walking*). In both of these cases, however, K-Sketch has other operations that nullify this advantage. This leaves three tasks for which I predict neither tool will have an advantage. Section 7.2.6.1 on page 265 explains how the results of this study measure up against these predictions.

7.2.3 Participants

I recruited seven participants for this study using the same poster as the previous study. Partial demographic information on these participants appears in Table 7-14 (with full demographic information appearing in Appendix H). One participant worked part-time as a software developer, but all were students at the University of Washington (five in technical communications, one in East Asia studies, and one in law). Participants ranged in age from 21 to 53. Three were men and four were women.

#	Occupation	Sex	Age	Drawing skill (1 = none, 5 = excellent)	Expertise (1 = complete beginner, 7 = expert)			Animation experience (hours)
					Computers	PowerPoint	Tablet PCs	
1	comm.. student	F	22	2	5	5	1	30
2	law student	F	39	4	6	5	1	0
3	teacher	M	33	2	3	3	1	1
4	Asia studies student	M	37	3	4	4	2	0
5	comm. student	F	21	3	4	4	1	16
6	software developer	M	53	2	6	3	1	0
7	comm. student	F	27	3	6	4	1	30
	Averages		*33*	*2.7*	*4.9*	*4.0*	*1.1*	*11.0*

Table 7-14: Laboratory study 2 participant demographics. Many participants were studying technical communications, including participant 6, who did this in addition to working as a software developer.

Participants in this study answered a demographics questionnaire that was nearly identical to the one in the previous study, and the results were similar. Again, participants rated their drawing skills using one of five ordinal responses: none, poor, fair, good, or excellent. Three rated their skill as poor, three as fair, and one as good. Participants also rated their expertise with computer tools on a 7-point scale (1=complete beginner, 7=expert). Participants were somewhat experienced with PowerPoint (M = 4.00, SD = 0.82) and very inexperienced with Tablet PCs (M = 1.14, SD = 0.38).

When asked how often they would create animations if they had the time and skill to do so, three responded "at least once a month" and four responded "not sure."

When asked why they wanted to create animations, the purpose varied. One wanted to make animations for web sites, another to design user interfaces, another for pitching ideas to colleagues and clients, and another for religious worship. Two others were interested purely for personal entertainment, and one was unsure. The variety is similar to what I observed in the previous study.

Four participants had next to no experience with any animation tool, but three others had some experience with other animation tools. One of these had 15 hours experience animating with the Keynote presentation tool, one had 30 hours experience making custom animation with PowerPoint, and one had 30 hours experience programming animations with HyperCard. Finally, on a 7-point Likert scale (1=disagree strongly, 7=agree strongly), participants were asked if they were discouraged from creating animations by the time required (M = 5.20, SD =0.84) or by the complexity of animation tools (M = 4.33, SD = 1.63). As before, these responses indicate that speed is participants' primary concern, followed by simplicity.

7.2.4 Environment

All sessions took place in a private room of the Computer Science and Engineering building at the University of Washington. All tasks were completed on a Compaq tc4200 tablet PC that was held up at an angle of approximately 20° to reduce drawing fatigue. I used the version of K-Sketch described in Chapter 5 and the TAB Lite version 2.2. Since the TAB Lite was designed for use with a mouse and keyboard, but to avoid giving one tool an advantage, I provided a mouse and keyboard for all tasks performed with both tools. (The keyboard was on a sliding tray that kept it out of the way when not in use.) For K-Sketch, participants had the option of using a bezel button or a hand held remote control for the Alternate button. Task animations were viewed on a small laptop positioned next to the tablet.

Each of the 28 sessions took about three hours. Most participants did consecutive sessions no less than one day and no more than four days apart. There were two exceptions: Participant 5 did her second K-Sketch session 10 days after her first, and Participant 7 did her first TAB Lite session 22 days after her last K-Sketch session.

7.2.5 Results

As with the previous study, I looked for statistical significance whenever possible, although I knew that I was less likely to find it with only seven participants. I had the same concerns about normality as in the previous study, and I dealt with the problem in the same way. If a Kolmogorov-Smirnov normality test found a significant result for a variable, I analyzed the natural logarithm of that variable or used a non-parametric hypothesis test.

Since only two participants were able to complete all nine tasks with both tools, it was not possible to analyze these results with a simple ANOVA as I did in the previous study. Instead, I performed a mixed-effects model analysis of variance that modeled tool and task as fixed effects and participant as a random effect (because the levels of this factor were drawn randomly from the population). Mixed-effects models are regression-based and correctly handle missing data, as needed for this study. Although mixed-effects models retain larger denominator degrees of freedom, they

		Participants completing	Participants using paper notes	Time to task incident (mean, in minutes)	Full task time (mean, in minutes)	How fast? (mean, 1–7, 1=fastest)	How easy? (mean, 1–7, 1=easiest)	Comfort showing (mean, 1–7)	Comfort creating (mean, 1-7)	Help incidents (total)	Bug incidents (total)	Task incidents (total)
Task A (face)	*KSK*	7	0	3.22	3.22	3.00	2.57	5.29	5.69	0	0	0
	TAB	7	0	4.64	4.64	3.43	3.14	5.32	5.24	1	0	0
Task B (sun)	*KSK*	7	1	**4.90**	**4.90**	3.17	2.86	5.55	5.41	0	0	0
	TAB	7	1	**12.41**	**14.14**	4.17	4.50	4.43	4.22	2	0	2
Task C (gears)	*KSK*	7	1	**5.96**	**6.43**	4.00	4.57	3.66	4.18	3	1	2
	TAB	7	1	**13.49**	**16.86**	4.86	4.86	4.41	4.10	9	5	4
Task D (ant)	*KSK*	7	0	6.64	6.64	4.00	3.71	5.18	5.02	3	1	0
	TAB	7	1	5.45	5.45	3.71	3.14	5.20	5.08	1	0	0
Task E (juggle)	*KSK*	7	1	**11.30**	18.67	5.43	5.43	4.45	3.96	6	3	9
	TAB	7	6	**22.55**	23.42	5.14	5.14	5.36	4.51	0	1	2
Task F (torch)	*KSK*	7	0	**5.01**	**5.21**	3.33	2.33	5.90	5.50	0	3	2
	TAB	6	5	**17.88**	**17.95**	4.50	4.17	4.28	3.76	0	0	1
Task G (plane)	*KSK*	7	0	18.59	19.72	4.80	4.80	6.14	5.17	8	5	4
	TAB	5	4	24.80	24.67	4.40	4.70	6.13	5.14	0	0	1
Task H (faerie)	*KSK*	6	0	**7.83**	7.85	3.00	2.67	6.00	5.33	0	6	0
	TAB	3	2	**14.90**	14.70	3.33	2.67	6.08	5.29	0	0	0
Task I (waves)	*KSK*	5	1	7.64	7.74	2.50	2.50	6.19	6.21	0	0	2
	TAB	2	1	15.50	16.29	3.00	2.50	6.94	6.50	0	0	1

Table 7-15: Laboratory study 2 summary results. Statistically significant results are shown in bold. Means for time to task incident and full task time have been adjusted to account for missing data. Care should be exercised when interpreting other means when less than seven participants completed a task with both tools, as these means only capture the performance of those who did well with both tools.

use wider confidence intervals to compensate, and therefore do not more readily result in statistical significance than traditional ANOVA [56, 102, 136]. Each time I analyzed a variable with a mixed-effects model analysis of variance, I assumed a compound symmetry covariance structure.

In the results that follow, I use an alpha of .05 for most tests of significance. When I need to correct for multiple tests of significance, I usually use a Bonferroni corrected alpha of $.05/n$ (where n is the number of related tests), as with the previous study. When the simple Bonferroni method is too conservative, I use Holm's sequential Bonferroni procedure instead and report the alpha of the last significant test. When significance depends on the correction method used, I report exact p-values.

Table 7-15 shows an overview of this study's results. The following sections present each category of results in greater details. I start with an analysis of task times and follow this with an analysis of subjective measures and comfort level responses. I then examine incidents that occurred during the study, and close with other results. Complete data for this study can be found in Appendix H.

7.2.5.1 Task time measures

Table 7-16 shows the time to the first task incident (if any) for each participant and task, and Table 7-17 shows the full time. The majority of these values are the same, but some (shown in italics in both tables) are quite different. Neither time to task incident nor full time were normally distributed, but the natural logarithm of each variable was, so I analyzed log times as in the previous study.

For time to task incident, a mixed-effects model analysis of variance found significant main effects for both tool ($F_{1,87.14}$=82.38, p<.001) and task ($F_{8,87.06}$=29.83, p<.001), as well as significant interactions between tool and task ($F_{8,87.04}$=5.40, p<.001). I then looked at pair wise effects for each task, correcting for multiple tests of significance using Holm's sequential Bonferroni method with a final alpha of .01. As shown in Table 7-16, the log of time to task incident was found to be significantly lower with K-Sketch than with the TAB Lite for five tasks: B ($F_{1,86.99}$=27.49, p<.001), C ($F_{1,86.99}$=21.21, p<.001), E ($F_{1,86.99}$=15.13, p<.001), F ($F_{1,87.04}$=47.23, p<.001), and H ($F_{1,87.14}$=7.37, p=.008). Two other tasks also had lower times with K-Sketch but missed significance due to correction for multiple tests: Task I barely missed ($F_{1,87.15}$=6.32, p=.014), and Task A missed by a wider margin ($F_{1,86.99}$=4.22, p=.043). No significant difference was detected in the times for Task D ($F_{1,86.99}$=1.24, p>.05) or Task G ($F_{1,87.08}$=2.17, p>.05).

Time to First Task Incident

		P1	P2	P3	P4	P5	P6	P7	Mean[a]
Task A (face)	*KSK*	3.17	5.18	5.75	4.15	1.40	2.95	2.22	3.22
	TAB	3.13	4.93	9.47	11.25	2.20	5.18	2.45	4.64
Task B (sun)	*KSK*	4.08	4.10	8.10	5.00	2.52	7.28	5.42	**4.90**
	TAB	9.62	13.57	*14.13*	21.03	*6.40*	13.78	13.27	**12.41**
Task C (gears)	*KSK*	4.05	8.13	13.65	8.23	2.48	*5.50*	*5.25*	**5.96**
	TAB	5.22	28.28	30.77	*41.70*	6.20	10.97	*6.30*	**13.49**
Task D (ant)	*KSK*	4.77	6.67	5.60	7.32	3.43	15.97	7.95	6.64
	TAB	4.35	4.68	4.78	15.17	2.52	9.43	4.07	5.45
Task E (juggle)	*KSK*	7.43	*10.10*	9.95	23.52	*5.37*	*13.53*	*18.48*	**11.30**
	TAB	11.05	23.83	27.00	36.65	14.70	32.75	*23.60*	**22.55**
Task F (torch)	*KSK*	4.55	7.05	*6.15*	6.93	2.07	5.58	*5.00*	**5.01**
	TAB	18.35	13.90	28.70		8.17	20.37	*16.40*	**17.88**
Task G (plane)	*KSK*	11.15	*11.65*	*15.35*	*40.68*	10.98	30.15	28.57	18.59
	TAB	22.50	*25.82*			9.68	32.05	23.78	24.80
Task H (faerie)	*KSK*	6.15	7.98	15.87	8.95	2.58		10.23	**7.83**
	TAB	14.95				4.78		15.83	**14.90**
Task I (waves)	*KSK*	4.82	9.27	7.45		3.28		*11.72*	7.64
	TAB	10.72				*7.78*			15.50

[a] These means are the inverse log of means computed through a regression-based statistical analysis of log time and may not exactly match a directly computed mean. Standard deviations are not shown, because their units are hard to interpret.

Table 7-16: Laboratory study 2 time to first task incident (in minutes). Significant differences are shown in bold in the far right column. Times that differ from the full time for the task are shown in italic.

Full Time

		P1	P2	P3	P4	P5	P6	P7	Mean[a]
Task A (face)	*KSK*	3.17	5.18	5.75	4.15	1.40	2.95	2.22	3.22
	TAB	3.13	4.93	9.47	11.25	2.20	5.18	2.45	4.64
Task B (sun)	*KSK*	4.08	4.10	8.10	5.00	2.52	7.28	5.42	**4.90**
	TAB	9.62	13.57	*25.28*	21.03	*8.87*	13.78	13.27	**14.14**
Task C (gears)	*KSK*	4.05	8.13	13.65	8.23	2.48	*7.93*	*6.20*	**6.43**
	TAB	5.22	28.28	30.77	*56.45*	6.20	10.97	*22.22*	**16.86**
Task D (ant)	*KSK*	4.77	6.67	5.60	7.32	3.43	15.97	7.95	6.64
	TAB	4.35	4.68	4.78	15.17	2.52	9.43	4.07	5.45
Task E (juggle)	*KSK*	7.43	*28.30*	9.95	23.52	*17.58*	*38.95*	*23.33*	18.67
	TAB	11.05	23.83	27.00	36.65	14.70	32.75	*30.73*	23.42
Task F (torch)	*KSK*	4.55	7.05	*6.70*	6.93	2.07	5.58	*6.02*	**5.21**
	TAB	18.35	13.90	28.70		8.17	20.37	*17.72*	**17.95**
Task G (plane)	*KSK*	11.15	*13.92*	*18.82*	*41.97*	10.98	30.15	28.57	19.72
	TAB	22.50	*27.07*			9.68	32.05	23.78	24.67
Task H (faerie)	*KSK*	6.15	7.98	15.87	8.95	2.58		10.23	7.85
	TAB	14.95				4.78		15.83	14.70
Task I (waves)	*KSK*	4.82	9.27	7.45		3.28		*12.92*	7.74
	TAB	10.72				*8.35*			16.29

[a] These means are the inverse log of means computed through a regression-based statistical analysis of log time and may not exactly match a directly computed mean. Standard deviations are not shown, because their units are hard to interpret.

Table 7-17: Laboratory study 2 full task time (in minutes). Significant differences are shown in bold in the far right column. Task times that differ from the time to first task intervention are shown in italic.

For the log of full time, the results are still in favor of K-Sketch, although not as strong. A mixed-effects model analysis of variance found significant main effects for both tool ($F_{1,87.18}$=65.75, p<.001) and task ($F_{8,87.08}$=31.63, p<.001), as well as significant interactions between tool and task ($F_{8,87.05}$=5.84, p<.001). I then looked at pair wise effects for each task, correcting for multiple tests of significance using Holm's sequential Bonferroni method with a final alpha of approximately .007. As shown in Table 7-17, the log of full time was found to be significantly lower with K-Sketch than with the TAB Lite for three tasks: B ($F_{1,87.00}$=31.45, p<.001), C ($F_{1,87.00}$=26.03, p<.001), and F ($F_{1,87.05}$=39.34, p<.001). Two other tasks also had lower times with K-Sketch but missed significance due to correction for multiple tests: Task H ($F_{1,87.17}$=6.15, p=.015) and Task I ($F_{1,87.19}$=6.16, p=.015). No significant difference was detected in the times for Task A ($F_{1,87.00}$=3.71, p>.05), Task D ($F_{1,87.00}$=1.09, p>.05), Task E ($F_{1,87.00}$=1.44, p>.05), or Task G ($F_{1,87.10}$=1.16, p>.05).

7.2.5.2 Subjective Measures

Subjective measures appear in Table 7-18 (How fast) and Table 7-19 (How easy). Neither measure was normally distributed, and it was therefore inappropriate to analyze these variables with a mixed-effects model analysis of variance. Using a Friedman tests was also inappropriate, however, because so much data was missing. I settled for non-parametric Wilcoxon tests on each subjective measure for each task separately, excluding missing cases test-by-test. I corrected for multiple tests of significance by using a Bonferroni corrected alpha of .05/9 ≈ .006. With this standard, no test showed significant differences for either measure (app p-values > .006). If I had not used a corrected alpha, only three test cases would have been significant:

Task A felt easier with K-Sketch (z=-2.00, N=7, p=.046), and Task F felt faster (z=-2.07, N=6, p=.038) and easier (z=-2.04, N=6, p=.041) with K-Sketch.

How Fast?

		P1	P2	P3	P4	P5	P6	P7	Mean	St.Dev.
Task A (face)	*KSK*	1	4	4	4	1	4	3	3.00	1.41
	TAB	2	6	4	5	2	3	2	3.43	1.62
Task B (sun)	*KSK*	1	5	4	4	2	4	4	3.17	1.33
	TAB	4	-[a]	4	4	4	4	5	4.17	0.41
Task C (gears)	*KSK*	4	4	5	3	3	5	4	4.00	0.82
	TAB	3	6	6	6	3	4	6	4.86	1.46
Task D (ant)	*KSK*	2	6	5	4	3	5	3	4.00	1.41
	TAB	3	4	4	4	2	6	3	3.71	1.25
Task E (juggle)	*KSK*	5	7	3	5	5	7	6	5.43	1.40
	TAB	4	6	5	4	4	6	7	5.14	1.22
Task F (torch)	*KSK*	3	6	3	4	2	2	4	3.33	1.51
	TAB	4	7	5		2	4	5	4.50	1.64
Task G (plane)	*KSK*	4	6	4	6	3	5	6	4.80	1.30
	TAB	5	6			3	4	4	4.40	1.14
Task H (faerie)	*KSK*	3	5	5	4	2		4	3.00	1.00
	TAB	4				1		5	3.33	2.08
Task I (waves)	*KSK*	2	3	2		3		5	2.50	0.71
	TAB	3				3		3	3.00	0.00

[a] Participant 2 omitted this question.

Table 7-18: Laboratory study 2 subjective speed measure (How fast). Responses were in the range 1–7 (1 = fastest). Task means and standard deviations are based on data from participants who completed that task with both tools.

How Easy?

		P1	P2	P3	P4	P5	P6	P7	Mean	St.Dev.
Task A	*KSK*	1	5	3	4	1	2	2	2.57	1.51
(face)	*TAB*	2	6	3	5	2	2	2	3.14	1.68
Task B	*KSK*	1	3	2	4	2	4	4	2.86	1.22
(sun)	*TAB*	5	6½	4	4	4	4	4	4.50	0.96
Task C	*KSK*	6	6	3	4	3	5	5	4.57	1.27
(gears)	*TAB*	3	7	6	6	3	3	6	4.86	1.77
Task D	*KSK*	2	6	2	4	2	6	4	3.71	1.80
(ant)	*TAB*	2	5	2	4	2	5	2	3.14	1.46
Task E	*KSK*	5	7	3	5	4	7	7	5.43	1.62
(juggle)	*TAB*	5	7	6	4	4	5	5	5.14	1.07
Task F	*KSK*	2	5	2	4	1	1	3	2.33	1.51
(torch)	*TAB*	5	5	4		2	4	5	4.17	1.17
Task G	*KSK*	3	6	5	6	3	6	6	4.80	1.64
(plane)	*TAB*	6	6			3	4½	4	4.70	1.30
Task H	*KSK*	3	6	5	4	1		4	2.67	1.53
(faerie)	*TAB*	4				1		3	2.67	1.53
Task I	*KSK*	2	5	2		3		5	2.50	0.71
(waves)	*TAB*	3				2			2.50	0.71

Table 7-19: Laboratory study 2 subjective simplicity measure (How easy). Responses were in the range 1–7 (1 = easiest). Task means and standard deviations are based on data from participants who completed that task with both tools.

7.2.5.3 **Comfort Levels**

As with the previous study, I averaged participants' responses to comfort-level questions across all audiences. Because I handled audience 1 (no one) differently (see Section 7.2.2.3), there were seven audiences for comfort creating and eight for comfort showing.

Comfort Showing Animations to Others

		P1	P2	P3	P4	P5	P6	P7	Mean	St.Dev.
Task A (face)	*KSK*	7.00	3.13	3.75	6.63	6.75	4.75	5.00	5.29	1.54
	TAB	6.13	4.38	3.38	6.63	6.25	6.00	4.50	5.32	1.23
Task B (sun)	*KSK*	6.50	4.63	3.75	6.63	7.00	6.00	4.38	5.55	1.28
	TAB	3.38	3.88	3.13	6.63	4.75	5.00	4.25	4.43	1.18
Task C (gears)	*KSK*	1.13	3.75	3.00	6.50	4.75	3.00	3.50	3.66	1.66
	TAB	2.75	4.63	2.25	6.50	5.75	6.00	3.00	4.41	1.74
Task D (ant)	*KSK*	7.00	4.75	3.63	6.63	6.00	4.00	4.25	5.18	1.35
	TAB	6.88	4.75	3.38	6.63	6.63	4.00	4.13	5.20	1.47
Task E (juggle)	*KSK*	6.13	2.63	4.13	6.50	4.00	4.00	3.75	4.45	1.37
	TAB	7.00	4.25	3.25	6.63	5.88	6.38	4.13	5.36	1.46
Task F (torch)	*KSK*	6.88	5.88	4.00	6.63	7.00	7.00	4.63	5.90	1.31
	TAB	5.25	3.75	3.00		5.88	4.81	3.00	4.28	1.21
Task G (plane)	*KSK*	7.00	6.38	3.50	6.38	6.75	5.94	4.63	6.14	0.94
	TAB	7.00	5.38			6.63	6.88	4.75	6.13	1.00
Task H (faerie)	*KSK*	6.75	3.25	3.13	6.63	6.13		5.13	6.00	0.82
	TAB	7.00				7.00		4.25	6.08	1.59
Task I (waves)	*KSK*	6.63	6.00	4.75		5.75		4.88	6.19	0.62
	TAB	7.00				6.88			6.94	0.09

Table 7-20: Laboratory study 2 comfort showing animations to others (1 = extremely uncomfortable, 7 = extremely comfortable). Responses were averaged across all eight audiences to produce this table. Task means and standard deviations are based on data from participants who completed that task with both tools.

Table 7-20 shows participants' comfort showing their animations to others. Since the average comfort showing was not normally distributed, I analyzed it with Wilcoxon tests as I did with the subjective measures. Using a Bonferroni corrected alpha of $.05/9 \approx .006$, I found no significant differences between tools for any task (all p-values $> .006$). If I had not used a corrected alpha, only two tasks would have shown significant differences: Participants were more comfortable showing the K-Sketch versions of their Task B (z=-2.20, N=7, p=.028) and Task F (z=-2.21, N=6,

p=.027) animations. Results were very similar when I considered each audience separately, and no audience stood out with particularly strong effects.

Table 7-21 shows participants' comfort creating their animations in front of others. This variable appeared to be normally distributed, so I analyzed it with a mixed-effects model analysis of variance. This test found a significant difference between tasks ($F_{8,87.03}$=3.51, p<.001), but no significant difference between tools ($F_{1,87.09}$=2.38, p>.05), and no interactions between tool and task ($F_{8,87.01}$=1.87, p>.05).

Comfort Creating Animations in Front of Others

		P1	P2	P3	P4	P5	P6	P7	Mean	St.Dev.
Task A (face)	*KSK*	6.71	4.43	3.14	7.00	7.00	5.14	6.43	5.69	1.49
	TAB	5.86	3.57	3.00	7.00	5.86	6.00	5.43	5.24	1.43
Task B (sun)	*KSK*	6.57	3.71	3.29	7.00	7.00	6.00	4.29	5.41	1.60
	TAB	3.71	2.43	3.00	7.00	5.14	5.00	3.29	4.22	1.58
Task C (gears)	*KSK*	1.43	3.29	2.57	7.00	5.14	4.29	5.57	4.18	1.91
	TAB	4.86	1.43	2.29	7.00	4.86	6.00	2.29	4.10	2.12
Task D (ant)	*KSK*	6.86	4.00	3.43	7.00	5.00	4.00	4.86	5.02	1.41
	TAB	7.00	2.86	3.57	7.00	5.86	4.00	5.29	5.08	1.65
Task E (juggle)	*KSK*	6.29	2.00	3.71	7.00	3.71	3.00	2.00	3.96	1.97
	TAB	5.57	2.86	2.86	7.00	4.57	6.43	2.29	4.51	1.89
Task F (torch)	*KSK*	6.43	3.86	4.00	7.00	6.43	7.00	5.29	5.50	1.34
	TAB	4.57	2.71	2.57		5.14	5.00	2.57	3.76	1.27
Task G (plane)	*KSK*	6.57	3.14	3.71	7.00	6.71	6.00	3.43	5.17	1.74
	TAB	6.00	2.57			6.14	6.00	5.00	5.14	1.51
Task H (faerie)	*KSK*	6.43	4.71	2.86	7.00	6.00		3.57	5.33	1.54
	TAB	5.86				6.43		3.57	5.29	1.51
Task I (waves)	*KSK*	7.00	4.14	4.43		5.43		3.71	6.21	1.11
	TAB	6.00				7.00			6.50	0.71

Table 7-21: Laboratory study 2 comfort creating animations in front of others (1 = extremely uncomfortable, 7 = extremely comfortable). Responses were averaged across all seven audiences to produce this table. Task means and standard deviations are based on data from participants who completed that task with both tools.

When I considered each audience separately with a Bonferroni corrected alpha of .05/9 ≈ .006, there were only minor differences. Because of the corrected alpha, the significant effect due to task disappeared for Audience 6 (300 students in a class, $F_{8,87.03}$=2.25, p=.031) and Audience 8 (300 professionals in a business presentation, $F_{8,87.02}$=2.77, p=.009). Also, if I had not used the corrected alpha, I would have found significant interactions between tool and task for Audience 3 (10 colleagues in a meeting, $F_{8,87.02}$=2.16, p=.038) and Audience 4 (tutoring one student, $F_{8,87.05}$=2.05, p=.049).

Because this test failed to find a significant main effect due to tool, I did not examine each task for differences between tools on comfort creating animations. To compare this variable more easily against comfort showing, however, I also analyzed it with Wilcoxon tests for each task separately, again using a corrected alpha of .05/9 ≈ .006. As with comfort showing, I found no significant differences between tools on comfort creating for any task (all p-values > .006). If I had not used a corrected alpha, the same two tasks would have shown significant differences: Participants were more comfortable creating Task B (z=-2.20, N=7, p=.028) and Task F (z=-2.21, N=6, p=.027) with K-Sketch. Again, results were very similar when I considered each audience separately, and no audience stood out with particularly strong effects.

7.2.5.4 Incidents

Counts for normal help incidents are shown in Table 7-22. Most participants needed little help, except for Participant 4, who was responsible for 14 of 20 K-Sketch help incidents and 9 of 13 TAB Lite help incidents. The most common cause of help incidents with K-Sketch was difficulty coordinating relative motions (10 of 20 incidents, all for Task G or Task E). The most common cause of help incidents with

Normal Help Incidents

		P1	P2	P3	P4	P5	P6	P7	Sum
Task A (face)	*KSK*	-	-	-	-	-	-	-	0
	TAB	-	-	-	1	-	-	-	1
Task B (sun)	*KSK*	-	-	-	-	-	-	-	0
	TAB	-	-	-	2	-	-	-	2
Task C (gears)	*KSK*	-	-	-	3	-	-	-	3
	TAB	-	2	1	5	-	-	1	9
Task D (ant)	*KSK*	-	-	-	3	-	-	-	3
	TAB	-	-	-	1	-	-	-	1
Task E (juggle)	*KSK*	-	-	-	1	-	4	1	6
	TAB	-	-	-	-	-	-	-	0
Task F (torch)	*KSK*	-	-	-	-	-	-	-	0
	TAB	-	-	-		-	-	-	0
Task G (plane)	*KSK*	-	-	-	7	-	1	-	8
	TAB	-	-			-	-	-	0
Task H (faerie)	*KSK*	-	-	-	-	-		-	0
	TAB	-				-		-	0
Task I (waves)	*KSK*	-	-	-		-		-	0
	TAB	-				-			0

Table 7-22: Laboratory study 2 normal help incident counts. To aid readability, only trials with non-zero incident counts are shown. A dash means the number of incidents is 0, and a blank means the participant did not complete the task. The total number of help incidents for K-Sketch was 20 and the total for the TAB Lite was 13.

Incidents	Reason for receiving help
K-Sketch	
10	Coordinating motions in multiple reference frames (6 from P4)
3	Forgot how to copy and paste objects (all from P4)
3	Forgot how to flip an object horizontally (all from P4)
TAB Lite	
6	General confusion generating in betweens (4 fromP4)
2	Unknown cause of bad in between (all from P4)

Table 7-23: Laboratory study 2 normal help incidents that occurred twice or more.

the TAB Lite was the generation of bad in betweens (8 of 13 incidents, all for Task B or Task C). Table 7-23 lists the most common reasons why participants sought help, noting which ones came from Participant 4.

Table 7-24 shows counts of bug incidents, and Table 7-25 lists the most common bugs encountered. Bugs were much more common with K-Sketch (17 total) than with the TAB Lite (5 total). The most common K-Sketch bug was corruption of the data model when working with relative motions (all four incidents encountered by Participant 3 during Task H). Also, the K-Sketch *Move handle* command caused the handle to make a strange jump on several occasions. K-Sketch would also occasionally give the illusion of recording when none took place, as in the previous study. All bugs with the TAB Lite had to do with its handling of filled regions, which were only used by Participant 4 in Task C.

Bug Incidents

		P1	P2	P3	P4	P5	P6	P7	Sum
Task A (face)	*KSK*	-	-	-	-	-	-	-	0
	TAB	-	-	-	-	-	-	-	0
Task B (sun)	*KSK*	-	-	-	-	-	-	-	0
	TAB	-	-	-	-	-	-	-	0
Task C (gears)	*KSK*	-	-	1	-	-	-	-	1
	TAB	-	-	-	5	-	-	-	5
Task D (ant)	*KSK*	-	-	-	-	-	1	-	1
	TAB	-	-	-	-	-	-	-	0
Task E (juggle)	*KSK*	-	-	-	2	1	-	-	3
	TAB	-	-	1	-	-	-	-	1
Task F (torch)	*KSK*	-	-	1	2	-	-	-	3
	TAB	-	-	-		-	-	-	0
Task G (plane)	*KSK*	-	-	-	3	1	-	1	5
	TAB	-	-			-	-	-	0
Task H (faerie)	*KSK*	-	-	6	-	-		-	6
	TAB	-				-		-	0
Task I (waves)	*KSK*	-	-	-		-		-	0
	TAB	-				-			0

Table 7-24: Laboratory study 2 bug incident counts. To aid readability, only trials with non-zero incident counts are shown. A dash means the number of incidents is 0, and a blank means the participant did not complete the task. The total number of bug incidents for K-Sketch was 19 and the total for the TAB Lite was 6.

Table 7-26 shows counts of task incidents. Due to the complexity of the tasks and my rigid standards for completion, these were much more common in this study than in the previous study. Nearly all incidents took place because participants did not realize that their animation did not meet specifications.

Statistical analysis of these counts was difficult for the same reasons as in the previous study (see Section 7.1.4.4 on page 221). Since a statistical comparison of incident counts between tools seemed even less likely to yield interesting results with this study than the previous one, I did not conduct such an analysis. I did, however, look for correlations between the full task time and the number of incidents. As before, I summed each variable across all tool and task conditions for each participant. For consistency with my previous analysis, I checked for correlations with non-parametric Spearman tests and a Bonferroni-corrected alpha of $.05/3 \approx .017$. Help incidents showed a particularly strong correlation with full task time (ρ=.929, p<.01), while bugs (ρ=.436, p>.05) and task incidents (ρ=.482, p>.05) showed no significant correlation.

Incidents	Bug encountered
K-Sketch	
4	Corrupted reference frames cause confusing motions
3	Get recording feedback without actually recording
3	Object handle jumps after *Move handle* command
2	Heuristics choose bad center point or reference frame for motion
2	Rotation spins in opposite direction after horizontal flip
2	Crash (data recovered)
TAB Lite	
3	Fill color expands into a region that should remain unfilled
2	Filled region is not in betweened

Table 7-25: Laboratory study 2 bug incidents that occurred twice or more.

Task Incidents

		P1	P2	P3	P4	P5	P6	P7	Sum
Task A	*KSK*	-	-	-	-	-	-	-	0
(face)	*TAB*	-	-	-	-	-	-	-	0
Task B	*KSK*	-	-	-	-	-	-	-	0
(sun)	*TAB*	-	-	1	-	1	-	-	2
Task C	*KSK*	-	-	-	-	-	1	1	2
(gears)	*TAB*	-	-	-	2	-	-	2	4
Task D	*KSK*	-	-	-	-	-	-	-	0
(ant)	*TAB*	-	-	-	-	-	-	-	0
Task E	*KSK*	-	2	-	-	2	4	1	9
(juggle)	*TAB*	-	-	-	-	-	-	2	2
Task F	*KSK*	-	-	1	-	-	-	1	2
(torch)	*TAB*	-	-	-		-	-	1	1
Task G	*KSK*	-	1	1	2	-	-	-	4
(plane)	*TAB*	-	1			-	-	-	1
Task H	*KSK*	-	-	-	-	-		-	0
(faerie)	*TAB*	-				-		-	0
Task I	*KSK*	-	-	-		-		2	2
(waves)	*TAB*	-				1		-	1

Table 7-26: Laboratory study 2 task incident counts. To aid readability, only trials with non-zero incident counts are shown. A dash means the number of incidents is 0, and a blank means the participant did not complete the task. The total number of task incidents for K-Sketch was 19 and the total for the TAB Lite was 11.

7.2.5.5 Other Results

I invited participants to comment on K-Sketch and the TAB Lite after each task and at the end of the study. Table 7-27 lists the comments made by the largest number of participants. Nearly all participants commented that they liked K-Sketch's timeline tic marks, particularly for editing the timing of events. Also, most commented that

Participants	Comment
K-Sketch	
likes	
6	Liked timeline tic marks (seeing, moving, or snapping)
4	Liked *Fix last operation* command
4	Liked *copy motion* operation
4	Liked *orient to path* operation
4	Liked capabilities of object handle
3	Liked animating by demonstration
3	Liked timeline
3	Liked *Move handle* command
dislikes	
7	Hard to coordinate motions
TAB Lite	
likes	
6	Liked *swing* operation
4	Liked in between capability
4	Liked onion skin capability
dislikes	
4	Hard to coordinate or keep track of events
4	Hard to in between one object without breaking other in betweens
4	Hard to do "math"
4	Hard to use manipulator
3	Hard to navigate through time

Table 7-27: Laboratory study 2 comments made by three or more participants.

they liked the ability to *fix* operations copy motions with K-Sketch. However, all participants commented at least once that coordinating motions with K-Sketch was difficult.

As for the TAB Lite, most participants commented that they liked the in between tool, and nearly all said that they liked the *swing* operation after they completed Task D (the only task that used it). Most also commented that they liked the TAB Lite's onion skinning capabilities. However, nearly all participants complained at least once that it was hard to coordinate events with the TAB Lite, particularly when trying to preserve existing in betweens. Most also complained about having to do math while creating animations with the TAB Lite.

Another noteworthy result of this study is that five out of seven participants were able to complete all nine experimental tasks during their two three-hour sessions with K-Sketch. Only two participants completed all nine tasks with the TAB Lite in the same time. Out of 63 possible tasks, participants missed only 3 K-Sketch tasks, compared to 12 TAB Lite tasks. Also, many participants found it necessary to sketch notes or diagrams on paper before creating more difficult animations with the TAB Lite. In all, participants made notes or sketches for 7% of completed K-Sketch tasks and 41% of completed TAB Lite tasks.

Finally, in the interest of continuing to improve the K-Sketch interface, Table 7-28 lists the most common types of difficulty participants experienced when using K-Sketch (as noted by the experimenters). Some of this difficulty relates to the object handle. Four participants grabbed the wrong region of the object handle on several occasions. All participants had trouble manipulating objects in a reference frame other than the world's frame. As explained in Section 5.1.4.2 and Figure 5-11 on page 128, such manipulations require the user to manipulate a ghost of the object that

Participants	Type of K-Sketch difficulty
7	Hard to demonstrate motions outside world reference frame (ghosts)
6	Accidentally place motions at wrong time
5	Expect timeline tic marks to shrink proportionally
4	Grab wrong region of object handle
4	Accidentally overwrite motions
3	Forget to fix an operation
3	Select wrong item in fix dialog
3	Trouble selecting objects

Table 7-28: Laboratory study 2 types of difficulty with K-Sketch experienced by three or more participants.

stays in a fixed reference frame during demonstration. For example, to spin the wheel of a car that is moving across the screen, the user must rotate a ghost of the wheel in one place while the car moves away. In this study, I observed that all users instinctively did exactly the opposite when performing such an operation for the first time: they followed the moving object with their pen rather than manipulating the ghost. Many had trouble controlling this instinct even after they had learned the proper way to perform such operations.

Participants experienced far less trouble timing events with K-Sketch than they did with the TAB Lite, but Table 7-28 shows that these problems were still present. Problems often began with participants inserting a motion at the wrong time or overwriting a motion accidentally. When adding relative motions, participants sometimes forgot to fix an operation immediately after performing it, which meant that their opportunity to fix it was lost. If participants did not repair their animations quickly, their attempts to recover often filled the timeline with small fragments of motion, making recovery progressively more difficult. This is why the full task time is much longer than the time to task incident for some trials, particularly with Task E (33–*The trickster mummy*). Participants often began experiencing these problems upon editing an animation.

7.2.6 Discussion

When the many results of this study are brought together, they present a detailed picture of how my interface optimization technique and K-Sketch interface design make an expressive animation interface for novices. Participants completed most tasks 2–3 times faster with K-Sketch, allowing them to complete 95% of the K-Sketch trials vs. 81% of the TAB Lite trials. Participants also made very positive comments. "Every one of the tasks was easier with K-Sketch," according to Participant 1. Participant 7 said, "My experience using K-Sketch was better—more pleasant and *way* more efficient." Participant 4, a complete novice at animation, was also grateful for K-Sketch, saying, "K-Sketch felt more intuitive. It made animation for a 'first-timer' more accessible quicker."

7.2.6.1 Evidence of Expressivity

Recall from Section 7.2.2.4 on page 240 that I predicted the outcome of each task using the animation operations used by the task and those provided by each tool. The statistical results for both task time measures support these hypotheses well (except for Task E, explained below). This gives weight to my claim that interface optimization made K-Sketch a more expressive tool. Other results also support this claim, though the inability to conduct reliable statistical analysis makes the support less clear. Here, I discuss the results for each task individually and explain how they match my predictions.

A. 39–*Cobalt:* This was a straightforward task for both tools. As I predicted, no difference between tools was found on any measure.

B. 15–*Man watching sun:* I predicted that the ability to translate along curved paths would give K-Sketch an advantage. The results showed an even greater difference than I expected. Participants struggled with TAB Lite in betweening, because the man moved in discreet jumps while the sun moved smoothly.

C. 66–*Gear reduction:* Drawing these gears was challenging in both tools (see Figure 7-3 on page 235), which resulted in small differences between tools on some measures. As I predicted, however, the inability to rotate more than 180° with the TAB Lite's in between tool made participants take significantly longer to complete the task.

D. 25–*Ant walking:* As I predicted, operations in both K-Sketch and the TAB Lite made this task easy with both tools. No significant differences were found in any measure.

E. 33–*The trickster mummy:* The juggler in this animation throws balls on curved paths, and I predicted that this would again give K-Sketch an edge over the TAB Lite. This advantage is significant, but it can be observed only in the time to first task incident. Timing events in this task was challenging (see Figure 7-5), and when participants began to fix errors, problems with both tools made K-Sketch's advantage disappear.

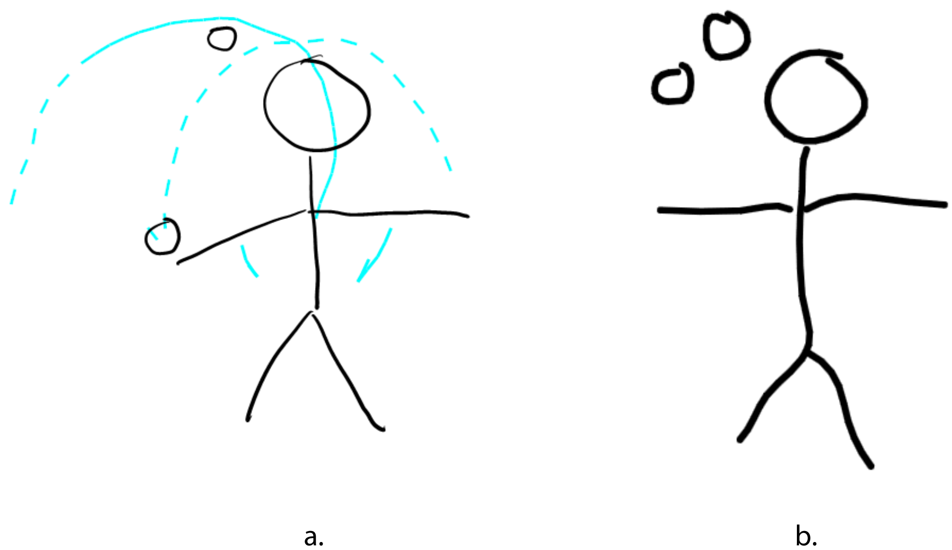


Figure 7-5: Laboratory study 2 Task E (33–*The trickster mummy*) animations produced by participant 6. *a:* The participant animated this juggler in 14 minutes with K-Sketch, but the sequence of events was not correct. Over the course of 4 task interventions, the timeline became progressively more fragmented, confusing the user and requiring 24 additional minutes before the final version shown here was complete. (The odd positioning of the upper ball's motion path is due to a bad center point for the ball that the participant did not correct). *b:* Later, this participant learned how to create storyboards, helping him to plan events more carefully. This allowed him to produce the TAB Lite version shown here in 32 minutes with no task interventions.

F. 26–*Lady with torch tumbling:* As I predicted, this task showed the strongest differences on all measures, because K-Sketch provided three important operations that the TAB Lite lacked (see Figure 7-4 on page 236).

G. 46–*Plane story:* This was a long task, but both tools were equally equipped to handle it. As I predicted, there were no significant differences in any measure.

H. 30–*Faerie adventures* : I expected zooming the camera to be easier with the TAB Lite's *interpolate*, but the motion of the faerie could be done in only three steps with K-Sketch's *move relative* and *set timing*. This gave K-Sketch a significant advantage, but it can be observed only in time to task incident, as the difference for full time slipped just below significant levels. (Since this analysis was regression-based, and since little data was available for task H, the significance of this task depends on the data for other tasks. Task E showed a much higher full time than time to task incident, and this could have caused slight fluctuations that resulted in Task H showing no significant differences for full time, even though significant differences were found for time to task incident.) Other differences were minor.

I. 51–*Detection of distant planets:* In this case, I predicted that *copy motion* and curved paths would give K-Sketch a significant advantage. Though little data was available for this task, the two participants who completed it with both tools needed half the time with K-Sketch. Using a regression-based statistical analysis, these differences in both time measures barely missed significance.

In summary, users were able to complete 8 of 9 tasks (all but Task D) faster with K-Sketch than with the TAB Lite. Time to task incident was significantly lower for K-Sketch in 5 of 9 tasks (one less than I predicted), and full time was significantly lower for K-Sketch in 3 of 9 tasks (three less than I predicted). When the significance did not match my prediction, it was usually a borderline case. Task E's full time was the exception, and this was caused by user difficulty when editing hierarchies of reference frames, a problem that can be corrected.

These results match my predictions well, and as I explained in Section 7.2.2.4 on page 240, this is evidence that K-Sketch is a more expressive tool than the TAB Lite. This is also evidence that interface optimization is a valid approach for designing interfaces, since this method helped me choose the animation operations that made K-Sketch so expressive. Furthermore, my ability to predict the outcomes of this study demonstrate that meaningful performance predictions can be made by characterizing tasks by the operations they use and tools by the operations they provide.

7.2.6.2 Support for Creative Improvisation

K-Sketch was designed to support creative improvisation with animation. Although both laboratory studies presented in this chapter revealed significant benefits to animating with K-Sketch, the controlled nature of these studies prevented participants from improvising. Participants did, however, notice this potential in K-Sketch. As Participant 4 put it, "[K-Sketch] really gives some real spontaneity, some real instantaneousness to the animation."

Though some participants did experience difficulty when editing animations with K-Sketch (as discussed later), most participants said that editing was one of K-Sketch's strengths. This freed participants from having to plan, allowing them to work quickly and patch things up as they went. Participant 5 described it this way, "When I have to do things fast I just kind of do it by trial and error instead of actually thinking it through, and so that doesn't really work when I've made a small error that I'll need to fix… four or five steps down the road." The *Fix last operation* dialog box played a big part in giving participants this sense of freedom. Whenever they detected any problem, most participants immediately clicked the *fix* button, even though I explained that this button only fixed reference frame issues or paste commands. This shows that most participants were not constructing a precise mental model of the animation as they proceeded, but rather devoting their energy to their creative task.

Participants also found K-Sketch's timeline liberating, because it gave them an overview of the entire animation in one place and simplified navigation through time. In the words of Participant 3, "…There was a certain smoothness that related to being able to see the animation all at once and then going to certain points within the animation to make changes or additions. This often helped me keep my focus on the entire animation rather than worrying about bits and pieces of it." In contrast, the TAB Lite had no overview of the animation, and participants frequently erased key frames by accident as they worked. At one point, this caused Participant 7 to explode in frustration, "That's stupid that you can make mistakes like that!"

Managing complex animations with the TAB Lite was so difficult that I had to teach participants storyboarding techniques during training phases. Few participants would have completed more than six of the nine tasks in the TAB Lite without this training. Participants were never forced to create storyboards, but most needed them to complete longer tasks. Since participants were simply copying animations, storyboarding was not particularly onerous in this study, but the need for so much preparation makes creative improvisation difficult.

The need for storyboards had the additional side effect of requiring participants to make mathematical computations. Participants could create each frame of the storyboard in the TAB Lite, but to make objects move at the right speed, they had to insert an appropriate number of frames between them. Because I required participants to keep the speed of moving objects "within 50% of what they saw," many participants found themselves doing math to fix the problem. This inspired many negative comments from Participant 1 ("Did I tell you I don't really like math?") and Participant 2 ("This is very math intensive! I'm not sure I signed up for a math test, okay?"), among others. Because K-Sketch uses demonstration, it takes advantage of users' intuitive sense of time and frees them from the burden of doing math. In all of these ways, K-Sketch provides greater support for creative improvisation than the TAB Lite.

7.2.6.3 Areas for Improvement

This study also revealed several areas for improvement in K-Sketch. Chief among them is the need to improve the editing experience. While K-Sketch's current editing capabilities do surpass those of the other tools studied in this chapter, it is possible for a few mistakes to snowball into a morass of fragmented motions and a confusing hierarchy of reference frames. There are several strategies for addressing these problems, as I discuss in Section 9.1 on page 314.

The difficulties participants experienced with the object handle may be a sign that the graphic design of this handle should be reconsidered. The fact that participants instinctively demonstrated all motions in the world reference frame also indicates the need for a re-design. Participants had trouble, manipulating a ghost in a fixed reference frame while the actual object moved. This is not surprising in retrospect, because the current design splits users' attention between two screen locations during demonstration, a time when focus is critical.

Finally, it is important to note that most participants made positive comments about the TAB Lite's *interpolate* operation. Two participants preferred the TAB Lite, because *interpolate* made motions more smooth (Participant 6) and because the mental model for interpolation was more similar to that of tools like Flash (Participant 5). It is true that the *interpolate* operation makes certain animations easier, and many users have a real need to transition their animations into a more formal form. For these reasons, it may make sense to include the *interpolate* operation in future versions of K-Sketch, if it can be done in a simple way. Section 9.1 briefly discusses a strategy for doing so.

In this chapter, K-Sketch has proven itself in head-to-head competition with two commercial tools: PowerPoint and the TAB Lite. PowerPoint's custom animation features were designed for presenters with little or no animation skill making formal animations. This focus on precision was PowerPoint's Achilles' heel in the first laboratory study, because participants were unable to resist the temptation to tweak small details in their animations. The second study removed the effects of informality and compared K-Sketch to the TAB Lite, a key frame animation tool designed for novices. Because of K-Sketch's carefully chosen animation operations, it outperformed or matched the TAB Lite in all tasks. These experimental results give strong evidence that K-Sketch is meeting its goal of balancing speed, simplicity, and expressivity to make a paper-like interface.

Both studies also provide evidence that K-Sketch's simple approach to animation supports creative improvisation like no other tool. The next chapter concludes the summative evaluation of this project with quantitative data from real use of K-Sketch as well as two case studies that show what happens when novices improvise animations with K-Sketch.

8 Real-world Evaluations

The laboratory studies presented in the previous chapter demonstrate K-Sketch's speed, simplicity, and expressivity in controlled settings. These are effective evaluations, but they assume that the field studies of Chapter 3 and interface optimization analysis of Chapter 4 are a correct reflection of how people would use K-Sketch in the real world. I validated these assumptions in the final phases of K-Sketch's summative evaluation by observing K-Sketch users "in the wild." The evaluations presented here were conducted under conditions that were less tightly controlled but more ecologically valid. They provide further evidence that K-Sketch helps people create a wide variety of animations quickly and to use animation in new ways.

This chapter begins in Section 8.1 with an analysis of usage logs and animation files collected from K-Sketch users. This analysis shows how the proportions of animation operations in my coded library of scenarios lines up with similar values collected from actual usage. This provides further evidence that I made sound assumptions during my interface optimization analysis and designed a system that matches users' needs. Deeper analysis of animation files also reveals further evidence of K-Sketch's speed, simplicity, and expressivity.

Section 8.2 closes the chapter with two case studies showing how K-Sketch is used to accomplish real work. The first explores a user interface design team's use of K-Sketch to prototype dynamic interface feedback. The second summarizes the experiences of a high school science teacher who instructed his students to create animations as a learning exercise. Both of these case studies show that K-Sketch is already helping people to work and learn in new ways.

8.1 Analysis of Usage Logs and Animation Files

K-Sketch has been available for download from http://www.k-sketch.org since February of 2008. Its availability was made public through announcements during conference presentations and invited talks. E-mail announcements were also sent to people who participated in field studies and others who expressed interest after reading an article on K-Sketch in *New Scientist* magazine [24]. In the first eight months after its release, K-Sketch was downloaded more than 1000 times and used in many projects. This user base is large enough that I can validate the estimates I made when coding my library of usage scenarios and look for further evidence of K-Sketch's speed, simplicity, and expressivity.

I begin this section by explaining how I collected usage logs and animation files. I then explain how this data compares to the library of scenarios that I collected in my field studies (see Chapter 3) and then coded for interface optimization (see Chapter 4). Then, I describe a deeper analysis I conducted on a subset of the animation files.

8.1.1 Collecting Usage Logs and Animation Files

K-Sketch is capable of recording usage logs that count the number of times a user performs any operation with any control. The K-Sketch installation program asked users if they would allow these anonymous logs to be collected and returned automatically. If they agreed, logs were posted to a server about once every two *sessions*, where a session was defined as a string of commands that ended when the user created a new animation or closed K-Sketch. A session, therefore, corresponds roughly to a period of K-Sketch usage resulting in a single animation. The correspondence is not perfect, however, because a bug prevented K-Sketch from ending sessions when the user loaded an existing animation for editing.

The analysis of this chapter is based on logs collected between February 24th and August 3rd 2008. I removed logs generated by development and debugging, as well as logs from users who simply started K-Sketch but never created animations. This left 138 log files covering 262 sessions from 54 users who were spread out between the United States, Europe, and Asia.

While these logs were being collected, I also collected a steady stream of animations created with K-Sketch. I collected some animations during open house events that presented K-Sketch to small groups of children. I collected others from

Category	Animations	Users[a]	Description
Without session data			
case study: school	101	89	high school science learning exercise (see Section 8.2.2)
open house 1	13	12	created by children after K-Sketch demonstration
design students	5	2	HCI/design students prototyping interfaces
women's seminar	4	4	created by undergraduate women after K-Sketch demonstration
others	11	8	assorted animations created by novices
Subtotal	*134*	*115*	
With session data			
child animation study	47	21	middle school study investigating animation as creative medium
case study: designer	31	1	HCI student prototyping interfaces (see Section 8.2.1)
lab study 2 doodles	5	2	doodles created after laboratory study 2 (see Section 7.2)
conference presentation	2	2	one-minute presentations at an HCI conference
open house 2	9	9	created by children after K-Sketch demonstration
Subtotal	*94*	*35*	
Total	*228*	*150*	

[a] Some animations (particularly in schools) were created by pairs of children working together. For the purpose of this analysis, a pair of children is considered to be one user.

Table 8-1: Files analyzed in this chapter. All files were considered when counting the number of operations used once or more (see Table 8-2), because this analysis was based solely on the content of the files. All other analyses required *session data* (a time-stamped list of commands executed to create an animation), which was available only for those animations in the bottom half of the table.

colleagues who began to use K-Sketch for various projects. Some users who had downloaded K-Sketch also posted their animations on the K-Sketch bulletin board. I also began to support others who expressed an interest in using K-Sketch for their work. (I give a detailed report of two such users' experiences in the case studies of Section 8.2.) I saved all of these files for further analysis.

I performed the analysis presented here on a collection of 228 animations from 150 different users. A summary of these user groups is presented in Table 8-1, and a small sample of these animations is shown in Figures 8–1, 8–2, and 8–3. The table and figures reflect a variety of subjects and user goals similar to the variety found in my field studies.

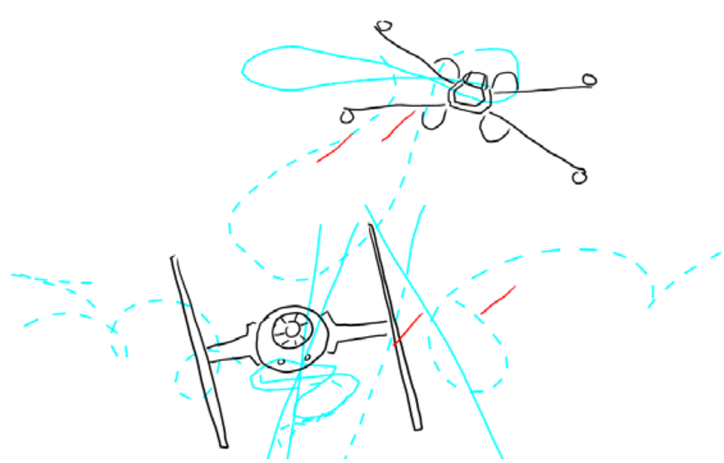


Figure 8-1: Star wars animation by an early K-Sketch user (9 seconds long with 89 ink strokes and 68 motions). No session data was recorded. The user created this animation to entertain himself and friends. Both ships move erratically while the x-wing (top) shoots and eventually destroys the tie fighter (bottom). The user reported creating this animation 10 minutes after encountering K-Sketch for the first time. View this animation online at http://www.k-sketch.org/dissertation/fig8-1.

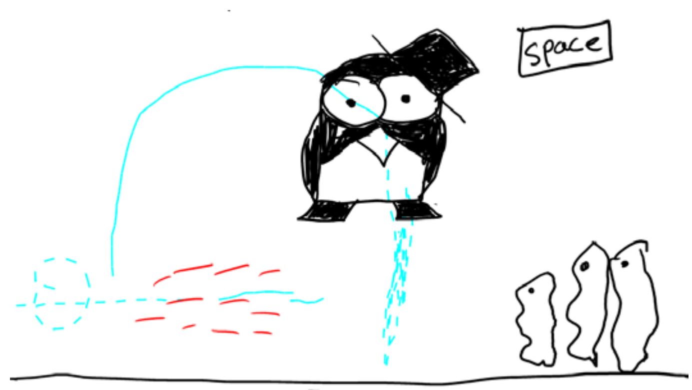


Figure 8-2: Video game animation created during the child animation study (37 seconds long with 93 ink strokes and 40 motions). Session data shows that this animation was created in 62 seconds using 80 commands, with the penguin drawing borrowed from a previous animation. This animation was created to demonstrate the action in a proposed video game. The penguin jumps over shots fired at him from the three blobs on the right, and later shoots all three blobs. View this animation online at http://www.k-sketch.org/dissertation/fig8-2.

Figure 8-3: Ubiquitous computing prototype animation created by an HCI graduate student at another university (33 seconds long with 210 ink strokes and 32 motions). No session data was available, but the user reported creating this animation in about 30 minutes. This animation was created to prototype a mobile fax application. The figure talks on his mobile phone, walks into a building, presses a button on his phone, and a fax emerges from the printer. View this animation online at http://www.k-sketch.org/dissertation/fig8-3.

As the table shows, a subset of these animations (94 animations from 35 users) also included *session data:* a time-stamped list of user actions like those used to generate logs. These animations provided the richest data, making it possible to answer detailed questions that could not be answered by analyzing animation files alone. Session data, for example, reveals how long it took to create an animation, and how many operations the user had to perform. This kind of information is necessary for most of the analysis here, but I include files without session data when possible.

The analysis presented here sometimes includes another set of animation files that I produced with K-Sketch to document my library of scenarios (see Appendix B). These animations realize a subset of the animations in the original library as accurately as possible. Because I produced these animations myself, I do not lump them together with the other animations in Table 8-1, nor do I analyze them in Section 8.1.3. I do analyze these animations in Section 8.1.2, however, when I compare the estimates in my coded library of scenarios to real world usage.

8.1.2 Comparing Against the Coded Scenario Library

The interface optimization of Chapter 4 relied on a coded library of scenarios. This coding broke each animation into features to be represented and then broke each feature into approaches for representing that feature with a particular set of animation operations. As I mentioned in Section 4.4 on page 103, it would be helpful to perform some validation of the coding procedure before optimizing to ensure that the coding accurately reflects reality. Though I did not validate my coding before optimization, I do so now by comparing it to real-world data. This will also demonstrate the effectiveness of my interface optimization technique.

A side-by-side comparison of the coded library with usage logs or animation files is difficult to produce for several reasons. First, the coded library included eight animation operations that K-Sketch did not provide: *repeat motion*, *define cels*, *morph*, *physically simulate*, *interpolate*, *move forward/back*, *deform*, and *move limb*. Since none of these operations could be used with K-Sketch, their presence would complicate this analysis and add little value. I therefore ignored these eight animation operations when analyzing the coded library, even when an approach required one. In such cases, a user would have to make do with the available operations, but I did not correct for this in my analysis by re-coding the scenario. A second difficulty is that no data source (not even the coded library) contains enough information to capture every occasion when an operation is used. In the tables below, I count the instances of animation operations that I am able to count and note potential inaccuracies.

A third difficulty is that the coded library sometimes lists many possible approaches to animating a feature. Each feature has a preferred (fastest) approach, but it is not clear that users would always use this approach. Because of this, I examine the coded library two different ways in the following two sections. First, I count the number of scenarios that use each operation in one or more legal[5] approaches. Then, I count the frequency of each operation (i.e., the number of times it is used) in preferred approaches only. Neither analysis validates the entire coding, but both provide a useful cross-reference.

Animations using an operation once or more

	Scenario lib. (predictions)	Scenario lib. subset[a] (predictions)	Scenario lib. subset[a] (actual)	Usage logs	Animation files
Animations	72	31	31	262	228
Users	38	11	1	54	163
Translate	76%	77%	32%	37%	61%
Scale	40%	32%	16%	28%	14%
Rotate	42%	42%	13%	25%	17%
Set timing	31%	39%	45%	n/a[b]	73%
Move relative	35%	42%	29%	8%	14%
Appear	56%	45%	65%	n/a[b]	87%
Disappear	60%	48%	68%	42%	70%
Trace	13%	3%	6%	30%	32%
Copy motion	14%	6%	n/a[c]	0%	n/a[c]
Orient to path	18%	19%	45%	25%	28%

[a] The subset of scenarios analyzed here is the same as the subset listed in Appendix B.
[b] Logs did not collect data on the *set timing* or *appear* animation operations.
[c] Static analysis of files could not distinguish copied motions from originals.

Table 8-2: Percentage of animations using an operation once or more. Scenario library predictions were computed from the coding used in interface optimization, while the actual values were computed from animations I produced for Appendix B. Usage logs were gathered anonymously from users who downloaded and used K-Sketch. Animation files were gathered from others who used K-Sketch, including participants in the cases studies presented in this chapter.

[5] As in Chapter 4, I ignored all approaches that took more than four extra steps.

8.1.2.1 Animations Using Each Operation Once or More

Table 8-2 counts the animations in each data set that use an operation once or more. The first data column shows the values I computed from the original coded library, while the second shows values computed from a subset of the coded library corresponding to the animations in Appendix B. This second column is included for easier comparison with the third column, which was computed from static analysis of the reproduced scenario animations appearing in Appendix B. I used the same static analysis on the complete set of 228 files I collected from K-Sketch users to produce the fifth column. Note that static analysis of files could not detect the presence of the *copy motion* operation, because the file format did not differentiate between normal motions and motions that were copies of other motions.

The fourth column shows the percentage of logs that contain each operation (recall that each log reflected one or two animations). Values for *set timing* and *appear* could not be computed, because the logs did not capture this data. The *trace* operations could not be detected either, but I assumed the operation was used if the *Record drawings* button was pressed. The value for *move relative* in this column may be lower than it should be, because this operation was only available for 68% of the logged sessions, and it omits instances when a new reference frame was inserted without the fix dialog. There is also a small chance that the value for *disappear* in this column is too high, because it was impossible to distinguish disappear operations from normal erasures.

8.1.2.2 Animation Operation Frequency

In my second analysis, I counted the number of times users performed each operation in each data set. The percentages show how often each operation occurred out of the total set of operations for that data set. (*Set timing* operations were not added to the total, because this operation was treated as a modifier for *translate, scale, rotate,* or *orient to path* operations.) Table 8-3 shows this frequency data.

Animation operation frequency

	Scenario lib. (predictions)	Scenario lib. subset[a] (predictions)	Scenario lib. subset[a] (actual)	Usage logs	Animation files
Animations	72	31	31	262	94
Users	38	11	1	54	35
Translate	29%	32%	32%	24%	26%
Scale	11%	10%	7%	6%	4%
Rotate	9%	12%	7%	6%	5%
Set timing	7%	12%	14%	n/a[b]	24%
Move relative	8%	11%	1%	1%	1%
Appear	16%	14%	24%	n/a[b]	30%
Disappear	17%	15%	24%	47%	20%
Trace	4%	0%	2%	6%	4%
Copy motion	4%	0%	2%	0%	2%
Orient to path	4%	5%	1%	10%	7%

[a] The subset of scenarios analyzed here is the same as the subset listed in Appendix B.
[b] Logs did not collect data on the *set timing* or *appear* animation operations.

Table 8-3: Animation operation frequency. Not all rows add to 100%, because of rounding error and because *set timing* was considered a modifier for *translate, scale, rotate,* or *orient to path.* Scenario library predictions were computed from the coding used in interface optimization, while the actual values were computed from animations I produced for Appendix B. Usage logs were gathered anonymously from users who downloaded and used K-Sketch. Animation files were gathered from others who used K-Sketch, including participants in the cases studies presented in this chapter.

I computed values for the original coded library (first column) and the subset of the library (second column) by considering only preferred approaches to each animation feature. If an animation operation appeared in a preferred approach, I counted it as one operation. I did not scale this count by the number of times each operation would be used in the approach, because the coded library did not contain such fine-grain data. This is a potential source of inaccuracy.

The third and fifth columns of this table were produced from reproduced scenario animations and the reduced set of 94 user files with session data. I omitted files without session data, because session data captured the frequency of each user operation more accurately than the static analysis of files. To keep these numbers as close as possible to those computed from the original coded library, I did not count undone operations. As before, the value for *move relative* may be lower than it should be, because this analysis counts uses of the fix dialog only and leaves out reference frames that were automatically inserted. On the other hand, this value might have been inflated slightly if the user clicked *Undo* after fixing, because I did not correct for this.

The fourth data column shows data collected from usage logs. The difficulties I experienced while generating this data are similar to those I experienced with the previous table. The logs did not give enough data to compute values for *set timing* and *appear*. The value for *trace* is actually the frequency that the *Record drawings* button is pressed. As before, the value for *move relative* may be too low because it leaves out reference frames inserted without the fix dialog. Finally, the value for *disappear* may be too high because it counts normal erasures as disappearances.

8.1.2.3 Discussion

Table 8-2 shows that actual K-Sketch use is close to predicted use, within a factor of two in most cases. There are notable discrepancies. The actual use of *move relative* may be lower than expected because of inaccurate counting, or it may be a sign of a deeper problem, such as difficulty using this operation. The usage logs did not reveal any use of *copy motion.* This may be due to the fact that the tutorial document did not explain how to use this operation. On the other hand, the actual use of *set timing* appears to be higher than expected, probably because K-Sketch automatically includes this with every recorded movement of the object handle. *Orient to path* was also used in more animations than predicted by the coded library, possibly because many users found this feature particularly fun to use. Recall that my interface optimization analysis did not support including this operation, but I included it because users liked the operation in early tests. This analysis of logs supports that design choice. Heavier than expected use of *orient to path* may also explain why the use of *translate* and *rotate* is lower than expected.

The meaning of Table 8-3 is less clear. The operations *appear* and *disappear* are used more frequently than predicted, while most others are used less frequently. A closer look at the re-implemented library of scenarios indicates that this is probably due to the fact that the coded library says little about how often an operation is used. Animations with many scenes or many objects animated frame-by-frame take more operations than others (primarily due to frequent use of *appear* and *disappear*), and therefore count for more in the analysis of files. Table 8-3 would be more useful, therefore, if the coded library took the frequency of operations into account, or possibly if the analysis normalized each animation's contribution to the total percentage.

While the numbers generated from real-world data do not exactly match the numbers predicted by my coded library of scenarios, they are close enough to show that the coded library was an excellent design guide. The design decisions prescribed by interface optimization are therefore trustworthy, although coarse distinctions appear to be more trustworthy than fine-grain distinctions. For example, *translate* and *appear* were clearly more important than *trace* and *copy motion,* but the relative importance of *translate* and *appear* is not as clear. In future evaluations of interface optimization, I will attempt to generate more accurate comparisons by including more accurate operation counts in coded scenario libraries and by designing usage logging systems that make similar counts.

8.1.3 *Deeper Analysis of Files*

After using this real-world data to evaluate my interface optimization technique, I looked for evidence that K-Sketch was meeting its design goals: speed, simplicity, and expressivity. I needed detailed information for this analysis, so I excluded usage logs and files without session data. This left the 94 files with session data listed in Table 8-1. In the sections that follow, I discuss each design goal individually and close with general discussion.

8.1.3.1 Speed Analysis

Since session data records a time-stamped list of user actions, I was able to construct a detailed picture of how long it took users to create animations. In clock time, most sessions took less than 20 minutes, but seven sessions (about 7%) took more than a day. Closer analysis of the logs revealed that the length of many sessions was due to long delays between user actions, during which the user may have been thinking about the animation or doing other work. To correct for this, I examined session length both in clock time and with pauses of more than one minute removed.

When I examined the longest sessions, I found that some users frequently opened existing animations as a starting point for new animations. This was most common in animations from the UI Designer described in Section 8.2.1 and animations from users making conference presentations. Unfortunately, such use triggered the K-Sketch bug (mentioned previously) that did not reset a session when the user opened

File analysis: speed

Time to create animations (in minutes)

	Mean	Median
including pauses:	985.33	15.05
not including pauses:	9.95	6.28

Time in minutes separated by animation length

Length	% in range	Median time (with pauses)	Median time (no pauses)
Less than 7 seconds	36%	16.32	4.72
7–15 seconds	36%	12.90	6.97
More than 15 seconds	28%	17.18	15.77

Time in minutes separated by animation size (# motions + # strokes/5)

Size	% in range	Median time (with pauses)	Median time (no pauses)
Less than 10 items	32%	6.13	3.73
10–20 items	28%	10.82	5.62
20–30 items	16%	21.08	8.43
30–40 items	10%	14.03	10.27
40–50 items	4%	33.27	15.72
50–60 items	4%	29.20	29.07
60–70 items	2%	17.18	15.77
70–80 items	1%	26.55	25.35
More than 80 items	3%	31.32	19.00

Table 8-4: Speed-related information collected from analysis of files. This table shows the median time to create the 94 animation files with session data, separating by length of the animation and by the size (number of ink strokes or motions). The table shows both the full time to create animations and the time with pauses of one minute or more removed. Medians are reported rather than means in these tables, because the mean time with pauses was distorted by a few animations that were created over many days. In size computations, five ink strokes were considered to be one *item*. This made the effect of ink strokes and motions about equal, because the mean number of strokes was five times the mean number of motions.

existing animation for editing. Consequently, some animations include session data from other animations. I was not able to remove this extra data reliably.

In spite of these problems, the data showed that users were able to create animations quickly. Mean and median times are shown at the top of Table 8-4. The mean time with pauses included is distorted by the few animations that took more than a day to create. I use medians instead of means in the remainder of my speed analysis to avoid this distortion.

I suspected that the time to create animations might depend on each animation's length and *size* (i.e., the number of objects and motions it contains). Consequently, Table 8-4 also separates the time to create animations by these metrics. When separating by animation length, the time without pauses increases as expected for longer animations. Separating by size required me to choose a size metric that accounted for both objects and motions. I chose to measure size in *items,* with each motion counting as one item and five ink strokes counting as one item. Since the average number of strokes was five times the average number of motions, this meant that drawings and motions contributed about equally to the size metric. Table 8-4 shows that time to create increases with animation size up to a point, but levels off at about 50 items.

8.1.3.2 Simplicity Analysis

Signs of K-Sketch's simplicity were harder to extract from this data, but I did find the numbers shown in Table 8-5. I first characterized the amount of work necessary to produce an animation by computing the mean edits per item. This uses the same size metric in the previous section, with one motion counting as one item and five ink strokes counting as one item. "Edits" include all drawing and animating operations,

File analysis: simplicity

Time to create animations	
Mean edits per item:	4.9
Animations with speed changes:	19
Mean speed changes[a]:	2.0

a Mean computed only for the 19 animations with speed changes.

Table 8-5: Simplicity-related information collected from analysis of files, with edits per item and speed changes. When computing the mean edits per item, all editing actions are considered, including undone operations, modifications, and ink strokes. The number of items includes motions and strokes as before, with five ink strokes considered one item.

whether or not they were modified or undone in the final version. If users completed all of their animations in one pass with no modification, this mean would be three edits per item[6]. The actual value is 4.9, which indicates that most animations require little modification after they are created.

[6] A simple way to compute this value is to consider that the mean number of strokes is five times the mean number of motions. Assuming five ink edits per graphical item and one motion edit per motion item, the mean edits per item would be equal to (5 + 1)/2 = 3.

The ease with which users demonstrate motions is also a sign of simplicity. For this reason, I also examined my data looking for signs of user difficulty when demonstrating motions. The speed control was the best indicator of this. I gave users this control to slow down the animation when demonstrating a motion at full speed was too difficult. Session data showed this control was used in only 19 animations (20%). Considering those 19 animations separately, the average number of speed changes was 2.0.

8.1.3.3 Expressivity Analysis

To evaluate K-Sketch's expressivity, I examined these animations for signs of variety. The variety of uses listed in Table 8-1 is one such sign. Other measures appear in Table 8-6. Most of these measures have wide ranges. In many cases, the means are much larger than the medians, and the standard deviations are much larger than the means. Both are signs that a measure's distribution has a long tail: most animations have small values and those with large values are spread out over a wide range. Reference frame depth is the one measure that showed little variety.

Figure 8-4 shows one of the extreme cases found in this table. The animation has over 70 rain drops that fall simultaneously. The user who created this animation took advantage of the *copy motion* operation to create this animation quickly. Figure 8-5 shows one of the animations with the highest number of reference frames for a single

File analysis: expressivity

Measure	Mean	Median	Std. dev.	Min	Max
Animation length (seconds)	13.2	9.3	12.36	0.5	67
Strokes	60.5	45.5	58.35	2	319
Motions	12.3	6	21.21	0	159
Grouped objects	4.5	2	9.74	0	79
Objects moving simultaneously	2.2	1	8.05	0	78
Reference frame depth	1.1	1	0.50	0	3

Table 8-6: Expressivity-related information collected from analysis of files. For most measures other than reference frame depth, the mean is greater than the median. The standard deviation is also greater than the mean in many cases, and the maximum value is quite high. These are signs that a measure's distribution has a long tail (i.e., the animations with large values are spread across a wide range).

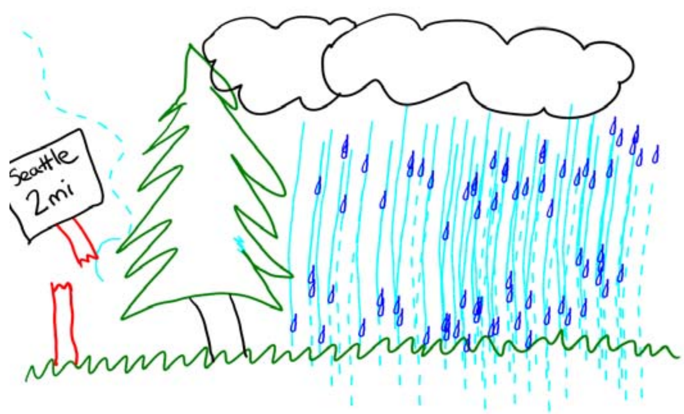


Figure 8-4: Rain storm animation created by Laboratory study 2 Participant 1 (6 seconds long with 102 ink strokes and 159 motions) (see Section 7.2). Session data shows that this animation was created in 58 seconds using 105 commands. This animation was created for fun at the end of Laboratory study 2. Rain drops fall while the wind blows a tree and breaks a road sign. The user created the rain drops with heavy use of copied graphics and motions, causing this animation to have 78 simultaneously moving parts, more than any other in this evaluation. View this animation online at http://www.k-sketch.org/dissertation/fig8-4.

object. The faces move around in one reference frame and shrink in two others (one for the faces and one for the faces and a rectangular frame).

Many measures in Table 8-6 show a minimum value of zero, because three animations contained ink strokes only and no motions. One of these was a template for generating other animations. In another, the user set out to make an animation, but found that a static picture would suffice. The third was a simple doodle. The number of grouped objects may also be a bit lower than it should, because the count excludes drawings that do nothing but appear or disappear.

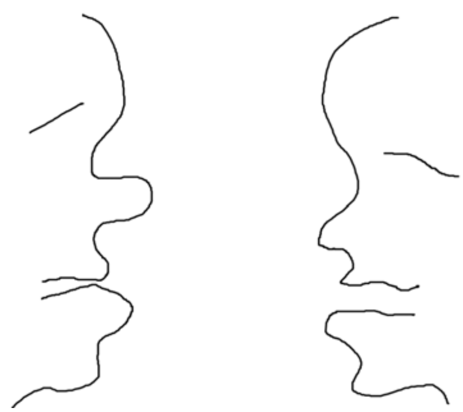


Figure 8-5: Short presentation created by a conference attendee (30 seconds long with 233 ink strokes and 80 motions). This animation was modified from an earlier animation for which no session data was available. The available session data showed that the modifications took 31 minutes (17 minutes plus two pauses of 1 and 13 minutes) with 84 commands. This animation was created by a computer science graduate student for a short conference presentation. In this scene, the two faces move their mouths and heads and shrink to occupy a smaller part of the frame. The faces have three reference frames: two for the faces themselves (one *translate* and one *scale*), and another *scale* for the faces and a blue frame that appears later. View this animation online at http://www.k-sketch.org/dissertation/fig8-5.

8.1.3.4 Discussion

This real-world evaluation complements my laboratory evaluations and shows further signs of K-Sketch's speed, simplicity, and expressivity. The time users needed to create these animations was comparable to the time needed by laboratory participants. This is a good sign, since the animations analyzed here tended to be more complex than those in my studies. I expected completion times for experimental tasks to be longer than completion times for real-world tasks, because users would not need to interpret complex task instructions in the real world. The fact that completion times level off at about 50 items is also a sign that K-Sketch scales surprisingly well to more complex animations.

Although this analysis provides few simplicity measures, the results are favorable for K-Sketch. The low edits per item is a sign that users do not often need to modify their animations extensively. Thus, users tend to create animations in one pass and avoid devoting mental energy to modifications. The small number of speed changes is also a sign of simplicity, because it shows that the speed of moving objects was rarely a source of difficulty. Users experience this difficulty in only 20% of the animations analyzed. When users did experience difficulty, they usually needed to change the animation speed only twice (once before demonstrating the difficult motion and once after demonstrating it).

Finally, the variety of uses I observed and the range of values for metrics such as animation length, number of strokes and motions, and number of moving objects is further evidence of K-Sketch's expressivity. As I hoped, users feel comfortable reaching for K-Sketch when they want to sketch out any moving picture. This frees them to explore ideas without having to find a special tool for each type of animation they can imagine.

8.2 Case Studies

The data analysis in the first part of this chapter highlights specific qualities of the K-Sketch interface and my interface optimization technique. The remainder of this chapter, however, addresses the broader and more challenging question of how K-Sketch helps people use animation in new ways. As evidence of this, I present two case studies in which K-Sketch enabled novices to use animation in ways they had not previously thought possible. In the first, a user interface researcher prototypes new interfaces with sketched animations. In the second, a high school science teacher introduces sketched animation as a learning exercise to a large number of students. In both cases, the presence of sketched animation fundamentally changes work and learning practices, producing dramatic improvements.

8.2.1 User Interface Design Case Study

My first case study observed a user interface research team designing a zero-button mouse for people with motor-control impairments. The primary participant in this case study was a female graduate student studying human-computer interaction in an information school. Under the supervision of her advisor, she wanted to design a new interaction technique that would allow people with limited motor control to perform target selection tasks without having to press mouse buttons.

This research team planned to use *goal crossing* [4, 156] as a substitute for button clicks. In goal crossing techniques, users signal their desire to select an object by moving the mouse pointer over a threshold (i.e., a goal line), rather than clicking on a confined area (e.g., a button). To create a zero-button mouse that would allow users to select a variety of targets in a direct manipulation interface, these researchers needed a way to distinguish incidental crosses that happen while moving the mouse from crosses intended to select a target. They set out to design a method for making this distinction. Many of their ideas involved prompts that would appear after a user's

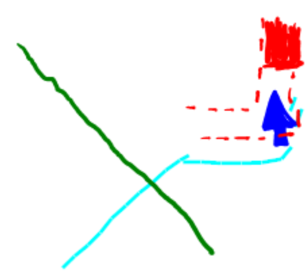


Figure 8-6: Hover Widget [63] crossing animation created during the UI design case study (11 seconds long with 32 ink strokes and 4 motions). Session data showed that this animation was created in 11 minutes (4 minutes plus a 7 minute pause) using 50 operations. After the cursor crosses the line, a track appears. The cursor then moves through the track, and when it reaches the square at the end, the word "cross" appears. View this animation online at http://www.k-sketch.org/dissertation/fig8-6.

mouse crossed certain targets: crossing the secondary targets in this prompt would indicate a selection action (see Figure 8-6). Their main design problem, then, was to design an intuitive post-crossing prompt that avoided occluding other screen objects.

I begin this section by describing this research team's design process and how K-Sketch fit into it. I then present the results of introducing K-Sketch into this team's process. I close with a discussion of these results.

8.2.1.1 Design Process

When I encountered this design team, their usual process was to sketch ideas on paper or on a whiteboard. Because of the dynamic nature of their designs, however, they found it difficult to communicate designs effectively to one another. Looking for a better design process, the research advisor prototyped a post-crossing prompt with a PowerPoint animation. The animation seemed to communicate his idea more effectively, but it was difficult and time-consuming to produce. Furthermore, the student would bear the primary responsibility for creating any animations, and she balked at the idea of producing so many animations in PowerPoint.

This research team wanted K-Sketch to help them quickly animate a large number of design ideas. I supplied the student designer with an HP tc1100 tablet PC, running Windows XP Tablet PC edition and the version of K-Sketch presented in Chapter 5. I gave this student a 15 minute introduction to K-Sketch and pointed her to the online user's guide (see Appendix G).

8.2.1.2 Results

Three weeks after acquiring K-Sketch, this student reported that her conversations with her advisor had improved. "Usually we begin our discussion on the board or on paper. It's usually confusing, but when we create animations, it focuses our discussion." During that time, she had prototyped 25 different designs with K-Sketch, a number that was inconceivable before. The designs flowed easily after a little practice. "Once I got the hang of it, it was pretty easy to use."

The figures in this section show examples of the design prototypes this student produced. Figure 8-6 shows a design inspired by Hover Widgets [63]. This promising design minimizes occlusion by hiding the post-crossing prompt once the

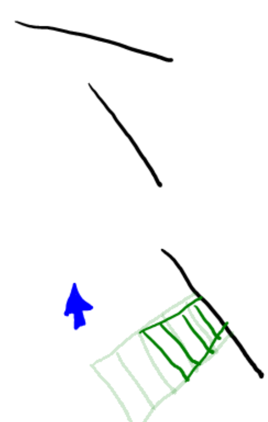


Figure 8-7: Trapezoidal feedback animation created during the UI design case study (3 seconds long with 38 ink strokes and 17 motions). Session data showed that this animation was created in 7½ minutes using 59 operations. A trapezoidal crossing region reaches out of the line when the mouse is near it and shrinks as the mouse moves away. View this animation online at http://www.k-sketch.org/dissertation/fig8-7.

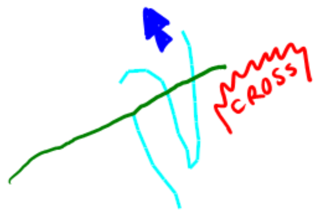


Figure 8-8: Zigzag crossing animation created during the UI design case study (5 seconds long with 10 ink strokes and 3 motions). Session data showed that this animation was created in 2½ minutes using 15 operations. The cursor moves across the line in a zigzag pattern. When it crosses the line a third time, the word "cross" appears. View this animation online at http://www.k-sketch.org/dissertation/fig8-8.

user has learned to perform the necessary gesture on an invisible track. Figure 8-7 shows a trapezoidal crossing prompt that reaches out for the mouse as it approaches and shrinks as it moves away. This is the animation done originally in PowerPoint. The student reported that the K-Sketch version took less time and omitted unnecessary details. Finally, Figure 8-8 shows a crossing technique that uses no post-crossing prompt, relying instead on a sequence of events to signal a selection action.

Most of this design student's comments were positive, but she did have some criticisms. She found motion paths distracting and turned them off while she was working. Also, she frequently needed to insert pauses at the end of animations, but the timeline tic marks did not allow her to do so without stretching the last motion. Since K-Sketch lumps many events into a single tic mark, this problem surfaces whenever a motion stops at the very end of the animation.

This student also had a number of requests. Most of these requests concerned the evolution of her most promising designs into more polished animations. Fast prototyping was her greatest need in the first phase of her project, but in later phases she needed to create more formal prototypes. She therefore wanted to replace groups of strokes with images or pre-defined objects. She also wanted more pen colors, a gradient fill tool, and a more precise eraser, so that she could make more polished animations. Finally, she requested the ability to add annotations that would not be visible when she played her animation. Because K-Sketch lacked this ability, she instead took paper notes on each animation and found it cumbersome to link these paper notes to the animation files. She believed that a separate page of handwritten notes (or a separate scene) would have been sufficient.

8.2.1.3 Discussion

K-Sketch enabled this user interface design team to evaluate more designs more quickly than they could before. Animation proved to be an effective communication medium, and no other animation tool fit this student's needs. As she explains, "I'm glad you were here. Otherwise I'd be doing this in Flash, and it would take me forever…Even PowerPoint is too polished, and you can't do things free hand." While animated prototypes were possible before, the time and energy needed to create them made them impractical in the early stages of design before K-Sketch was available.

This is an excellent example of how K-Sketch enables people to use animation in new ways.

The difficulty this student experienced when attempting to insert pauses at the end of the animation is not surprising. I mentioned this on page 143 (in Section 5.1.9) as one of the reasons for re-designing K-Sketch's timeline feedback. The need to evolve rough animations into more polished versions is also a request I expected to encounter. I discuss future work to address this problem in Section 9.1 on page 314.

8.2.2 Science Learning Case Study

My second case study observed over a hundred students creating K-Sketch animations as learning exercises. This study took place at a secondary school (high school) in Great Britain with about 1500 students between 12 and 19 years of age from a variety of financial backgrounds. Students were not very diverse ethnically, though they were many students who used English as a second language.

K-Sketch was introduced into this school by a science teacher who wanted to help his students create and describe mental models for scientific ideas. This teacher believed that drawing pictures to visualize scientific phenomena was particularly important for students with a lower literacy level. His own words explain his teaching philosophy succinctly:

> I am hoping to use k-sketch to allow pupils to create 'models' of their ideas, for example, in answering the question how does the Van de Graaff generator make a

> person's hair stand on end? To reply to this question in a written form will require the pupils to sequence the pictures they see in their head. Then by using detailed annotations the pupils can often write a much more detailed account and show a lot of progress. K-sketch should allow the pupils to really engage with this all important modeling phase, as they will have to sequence carefully when they decide how to animate. This should then lead to improved understanding and improved questioning in areas of weakness.

This teacher did not believe that students would get the same benefits from drawing static pictures. "Pupils might get 'snap shots' of the model, enough to draw a simple cartoon, but they might not know how they link. Cause and effect can then be either ignored of misunderstood." He felt that animation was important, because it helped students establish causal links. At first, he intended for his students to use Sketchy, a simple frame-by-frame animator [59]. He was delighted to discover that K-Sketch would allow his students to create a wider variety of animations more quickly.

As with the previous case study, I begin by explaining how K-Sketch fits into these students' learning process. I then describe the results of these K-Sketch learning exercises. I close with a discussion of these results.

8.2.2.1 Learning Process

This science teacher introduced K-Sketch to many classes for a variety of learning tasks. In all cases, the concepts being studied were introduced before K-Sketch exercises took place. Sometimes, K-Sketch exercises took place after concepts were discussed for just a few class periods. In these cases, the teacher was careful to show students static diagrams only, and no animations. In other cases, K-Sketch exercises took place during topic review, just before an exam.

When students encountered K-Sketch for the first time, the teacher led them through the tutorial in the K-Sketch User's Guide (see Appendix G). When the time came for a learning exercise, the teacher would pose a question, such as "Why is the lattice enthalpy of MgO greater than the lattice enthalpy of $MgCl_2$?" and ask students to answer it with an animation. The time students had to produce these animations varied from 30 minutes in one class period to 2½ hours spread out over a week. The teacher recorded his observations as the students worked, then collected students' solutions, and finally recorded his observations of student solutions. Not all students saved their work, however, so not all data was available for analysis.

Students created their animations on one of 16 Samsung Q1 Ultra Tablet PCs that this teacher acquired for the purpose of running K-Sketch. This model of tablet has a seven inch touch-sensitive screen with a tiny qwerty keyboard on either side of it. Each machine was running Windows XP Tablet PC edition and the version of K-

Sketch presented in Chapter 5. Because only 16 machines were available, larger groups of students sometimes had to work in pairs or an occasional triple.

8.2.2.2 Results

The teacher recorded his students' experience with K-Sketch for three months. His records document eight K-Sketch tasks spread out between five classes totaling 106 students. This is not an exhaustive count of all learning exercises conducted during this period, but it is enough to capture the scope of this project. Table 8-7 describes the classes that used K-Sketch, while Table 8-8 lists the learning tasks that were documented.

As Table 8-8 shows, these learning tasks covered a range of topics. The purpose of Task 1 (lab safety rules) was primarily to get more practice with K-Sketch, but it produced creative solutions such as the one in Figure 8-9. Figure 8-10 and Figure 8-11 show more typical examples covering topics primarily in chemistry and physics. I collected at total of 101 animations that ranged in length from three seconds to

Class	Average age	Ability profile	Boys	Girls
A	15	low	6	6
B	15	high	19	12
C	15	medium	13	9
D	15	medium-high	28	7
E	17	high	4	2

Table 8-7: Classes for which K-Sketch usage data was collected in the science learning case study. To simplify discussion, I identify each class by the letters in the left column. All other information came from the science teacher who ran this study.

Task	Class	Animations collected	Description
1	A	10	Describe a laboratory safety rule (K-Sketch practice)
2	A	4	Describe the events in the big bang theory.
3	B	15	Describe how crude oil is formed
4	B	7	Describe how catalytic cracking forms hydrocarbons
5	B	17	Teach other students about properties of radiation
6	C	12	*(same as above)*
7	D	23	*(same as above)*
8	E	6	Explain how ion properties affect lattice enthalpy

Table 8-8: Tasks for which K-Sketch usage data was collected in the science learning case study. The teacher's primary goal in the first task was to give students experience with K-Sketch. In later tasks, his primary goal was to teach the concepts described. The number of animations given is the number collected, but some students neglected to save their animations. In addition to the 94 animations listed in this table, seven other animations from scattered tasks were collected, bringing the total to 101.

Figure 8-9: Task 1 (lab rules) animation created during the science learning case study (18 seconds long with 108 ink strokes and 12 motions). This animation demonstrates the importance of tying hair back before approaching a Bunsen burner. An unfortunate girl approaches a Bunsen burner with a flickering flame. Her hair catches fire and she runs around until it all burns away. View this animation online at http://www.k-sketch.org/dissertation/fig8-9.

seven minutes. Longer animations tended to have many scenes. Animations had between 5 and 1319 ink strokes with between 3 and 851 motions (except for one that had zero motions). These animations did not record session data, so I was not able to collect detailed information about the time to create these animations.

During or after these K-Sketch learning exercises, this teacher observed three significant benefits. First, he observed that some students showed evidence of better recall (particularly with Task 2: the big bang theory). Second, many students produced animations that revealed fine detail about their misconceptions (particularly with Task 4: catalytic cracking shown in Figure 8-10, and with Task 8: lattice enthalpy). With a better understanding of student misconceptions, the teacher was better equipped to plan his later lessons. Finally, the teacher noticed that a particular group of students with attendance and behavior problems (Class C), was more engaged in learning while using K-Sketch. "Overall the level of engagement with this class was good. Considering their reputation for poor behaviour and vandalism they used the equipment and software impeccably. They felt proud of their accomplishments by the end of the lesson."

The teacher made several other observations about student animations. While most students were able to draw clearly and formulate sequences of ideas, some had trouble grasping the concept of communicating with animation. For example, some students did not add appropriate labels for sequences of events, particularly at the

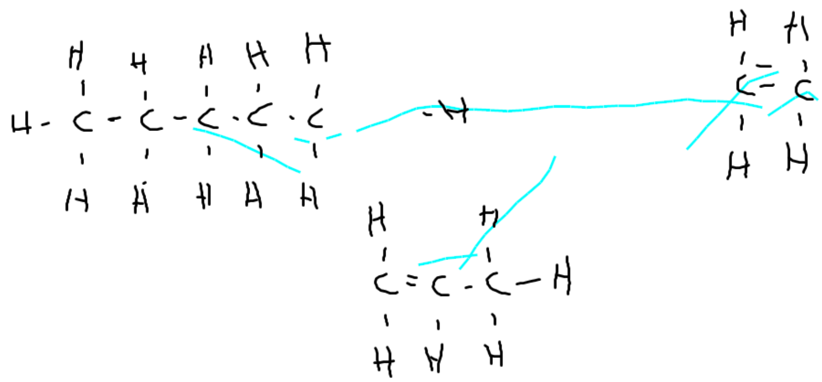
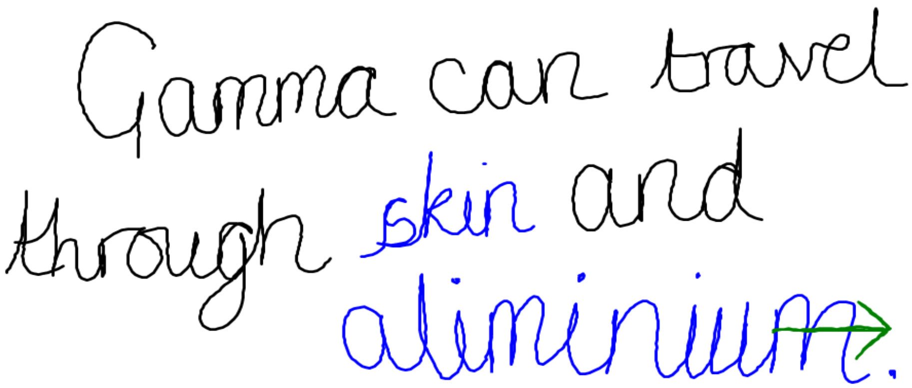
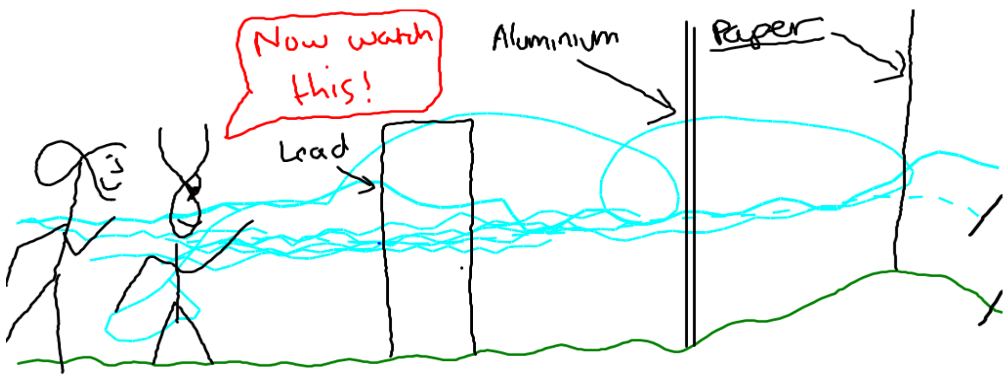


Figure 8-10: Task 4 (catalytic cracking) animation created during the science learning case study (15 seconds long with 124 ink strokes and 11 motions). This animation shows a large hydrocarbon separating into three parts, with atoms and bonds trading places. View this animation online at http://www.k-sketch.org/dissertation/fig8-10.

Figure 8-11: Task 3 (formation of crude oil) animation created during the science learning case study (30 seconds long with 96 ink strokes and 28 motions). In this animation, a fish swims around, dies, and turns into oil. Then, an oil drill is built above the water to harvest the oil. View this animation online at http://www.k-sketch.org/dissertation/fig8-11.



a.



b.

Figure 8-12: Two Task 5 (radiation) animations created during the science learning case study. *a:* This animation resembled a PowerPoint presentation, showing a sequence of panels with words and pictures describing properties of gamma radiation (23 seconds long with 266 ink strokes and 63 motions). *b:* This animation was a more creative solution to the same problem. An alpha particle-man attempts to walk through paper, but is repelled. Then, a beta particle-man succeeds in walking through paper. Finally, a gamma particle-man walks through lead, aluminum, and paper. (58 seconds long with 220 ink strokes and 43 motions.) View these animations online at http://www.k-sketch.org/dissertation/fig8-12a and http://www.k-sketch.org/dissertation/fig8-12b.

beginning or the end of event sequences. Other students suffered from what this teacher called “PowerPoint syndrome.” Their animations resembled PowerPoint presentations with sequences of words and drawings (see Figure 8-12a), even though the teacher was hoping for more creative use of animation (see Figure 8-12b). Finally, some students made excessive use of the *trace* operation for drawing text. This was fun for the students, but it produced poor presentations.

8.2.2.3 Discussion

K-Sketch enabled the students in this case study to think and learn in new ways. Comments from the science teacher who organized these learning activities showed that K-Sketch made new teaching techniques possible: “K-sketch’s simple user-friendly interface and stylus means all pupils can access their ideas in a more effective way than a normal piece of paper.” Students who participated in these activities recalled information more easily, and the animations they created helped their teacher understand their misconceptions. K-Sketch also proved to be an excellent way to engage problem students.

The problems students experienced when creating K-Sketch animations show that they quickly reached a high level of proficiency with the tool. Teacher comments on student animations showed that most of their problems concerned the subject matter under study or stylistic issues, such as when to add text labels. As I hoped, the K-Sketch interface faded into the background, allowing students to focus more of their attention on these more important problems. Some students did have difficulty thinking about animation as anything other than a sequence of presentation slides, but their teacher believed this to be merely a bad habit: “They have been preconditioned to think of information in terms of bullet points on slides and some of their work resembles this.” With practice, this habit could be replaced with better ones.

This science teacher’s experience was so positive that he expanded his use of K-Sketch the following school year, adding 90 more students. He also trained two other teachers at his school to use his K-Sketch teaching methods. Because the response to this project was so positive, he presented his methods at two county-wide teaching consortia. Thus, the new teaching methods that K-Sketch makes possible are already beginning to spread.

The final evaluations in this chapter show that the benefits of K-Sketch are not confined to laboratory experiments. The software is already in use across the world. Usage logs and animation files I gathered from K-Sketch users show that they are using K-Sketch much as I predicted they would. The balance of animation operations in actual usage resembles the predictions I made during my interface optimization analysis, providing further validation of this technique. The usage patterns I observed in animation files with session data show more signs that K-Sketch is reaching its low-level goals of supporting speed, simplicity, and expressivity. Finally, the case studies presented here show that K-Sketch is reaching its high-level goal of enabling new uses for animation.

9 Future Work

This dissertation has presented a complete picture of K-Sketch and established that my goals and design methods are sound, but there is ample room for future work. This chapter gives an overview of possible extensions to the K-Sketch project. The first two sections discuss extensions mentioned in earlier chapters: enhancements to K-Sketch itself (see Section 9.1) and a generalization of the interface optimization technique (see Section 9.2). Following this, I describe two novel uses of informal animation enabled by K-Sketch. Section 9.3 describes a wizard of oz prototyping system that could use K-Sketch to simulate dynamic interface behavior on the fly during user tests. Section 9.4 explains how K-Sketch's animation techniques can help novices create simple games and simulations. Finally, Section 9.5 discusses how K-Sketch heralds new possibilities for managing ideas throughout their life cycle.

9.1 K-Sketch Enhancements

The summative evaluation of Chapters 7 and 8 pointed to a need to improve some aspects of K-Sketch's animation editing experience. As many of the improvements mentioned here could add complexity, each should be weighed carefully against the benefit it provides before including it in the interface. Here, however, I simply list possible improvements and leave such careful discernment for future work.

One of the most obvious problems uncovered by laboratory study 2 was that participants expected all objects to be manipulated in the world reference frame. Future versions of K-Sketch should work this way, even when manipulating objects that are not in the world reference frame. There was also evidence that users could get lost in large numbers of reference frames, particularly when the timeline became fragmented with many short motions. This confusion could be reduced by limiting the number of reference frames available for each object to three: one for translation, one for scaling, and one for rotation. More reference frames could still be added when users combine two objects to create a new object. This simplification would eliminate much of the confusion I observed in user studies, and it does not prevent users from creating any animation I encountered in this research.

Another noteworthy outcome of laboratory study 2 is that all participants liked the *interpolate* animation operation in the TAB Lite. Two actually preferred the TAB Lite to K-Sketch because *interpolate* made some types of motion easier. Even though *interpolate* is seldom needed in these cases, it may make sense for K-Sketch to include this operation as an accommodation to some novices. This could be done by changing the standard object manipulation behavior to insert *interpolate* operations instead of instantaneous changes. Instantaneous changes could still be simulated by copying and pasting objects at a moment in time or by providing a special mode. This is a major change and would require a careful study of the tradeoffs as novices learn.

K-Sketch's timeline also showed room for improvement, particularly in the appearance and behavior of timeline tic marks. Some mechanism is needed for selecting a subset of events at a particular time. In particular, the tic marks should allow the end of one motion to be manipulated separately from the beginning of another motion. Also, one of the two modes for moving tic marks should stretch the timeline proportionally, as many users expected. Finally, it may make sense to change K-Sketch's behavior when overwriting motions so that it scales the new motion to match the duration of the old motion. This would reduce the fragmentation of motion that confused many users, and it seems to match their expectations.

In addition to fragmentation in the timeline, objects sometimes became fragmented when users could not select them accurately. Selection problems were frequent enough that it makes sense to revisit K-Sketch's selection behavior. Though K-Sketch currently groups objects implicitly, there is no way for users to access these groups. Future versions of K-Sketch will select all of an implicitly grouped object if any part of it is selected. Some mechanism is still needed for regrouping and choosing between groups when strokes are in multiple groups. Some gesture or delay inserted into a selection loop may be sufficient for cycling through possibilities.

These changes would make K-Sketch even more accessible to novices. With a few more design iterations, the K-Sketch interface may reach a steady state. K-Sketch could then evolve into a default animation interface that could be inserted into many other applications.

9.2 Improving Interface Optimization

With K-Sketch, Interface optimization has proven itself as a reliable tool for designing interfaces that balance speed, simplicity, and expressivity. Another area of future work already discussed in Section 4.4 is the need to address several limitations with this technique. The most significant limitation is that the interface optimization program presented in this dissertation takes intolerably long to produce a result when there are more than 40 operations. Also, the way in which scenarios are coded could be improved to more accurately reflect the complexity of operations and the time required to complete a scenario. Steps have already been taken to address these problems. A recent revision of interface optimization found a new approximation algorithm that handled 84 operations in 2 hours [33]. This revision also added time estimates to coded scenarios [33]. Future versions will also add an index of difficulty to operations so that complexity can be captured more accurately.

Some design teams may also wish to cross-check their coded library of scenarios to ensure that the coding technique is appropriate and estimated values are as accurate as possible. Section 4.4 discussed possible methods for addressing both of these problems. If only half of a library's scenarios are used to develop a coding manual (including the definitions for interface operations), then the quality of the manual can be judged by the number of modifications needed to code the second half of the library. Accuracy of the coding can be verified by having another person use the manual to code 25% of the scenarios and comparing results. Both of these techniques have been used effectively in social science research [77].

Finally, coding a library of scenarios is a laborious process that can take several iterations, and designers who use this technique would benefit from a better defined coding process that can be easily taught. Also, future projects would benefit from tools that manage the coding process and automate certain tasks. Operations may be split and merged as the coding manual evolves, and a tool that manages the process could help a designer keep track of previously coded scenarios that must be updated. Coding tools could also allow designers to define macros that capture sequences of interface operations for reuse in multiple scenarios. These macros would improve the consistency of the coding and document assumptions made when coding the library.

9.3 Wizard of Oz Simulations of Dynamic Interface Behavior

K-Sketch makes animation so simple and fast that it enables new uses of animation. Because K-Sketch's user interface is also quite simple, it can easily be merged with other applications to add animation capabilities. One application that would benefit from sketched animation is SketchWizard, a tool that enables designers to create wizard of oz prototypes of ink-based applications [43].

With SketchWizard, designers use the interface shown in Figure 9-1 to create interface prototypes that users can access on a remote computer. When users manipulate these interfaces, the designer can simulate the response of the interface in real time by editing the user's view. Studies showed that users are able to interact with SketchWizard prototypes as if they were functional programs, even with delays of 30 seconds ore more while the designer updates their view [43].

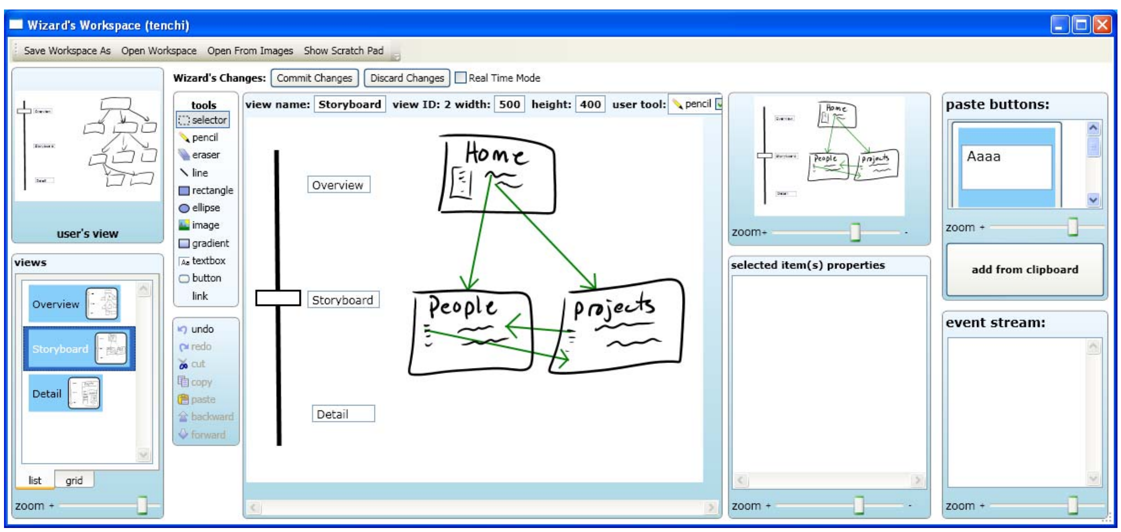


Figure 9-1: SketchWizard simulating semantic zooming in a web site design application. SketchWizard allows designers to prototype applications by simulating interface behavior in real time with graphical edits. However, designers cannot prototype dynamic behavior, such as an animated transition between zoom levels. Merging K-Sketch into SketchWizard would allow designers to prototype such animated behavior as well.

SketchWizard is somewhat limited, however, in the range of behavior that it can simulate. Consider the semantic zooming behavior in Figure 9-1. In this simulated interface (center of Figure 9-1), the slider on the left controls the web site map on the right. Higher levels of magnification not only zoom in on web pages, but also increase the level of detail shown. With the current version of SketchWizard, the transition between levels of detail would be instantaneous. If SketchWizard were enhanced with K-Sketch's animation interface, however, a designer could quickly prototype an animated transition between levels of detail, even in the middle of a user test. Another interface that could be prototyped in this way is the trapezoidal feedback from the UI designer case study (see Figure 8-7 on page 1). If this designer had K-Sketch and SketchWizard together, she could have gathered user feedback on her designs with only a little extra effort.

9.4 End-user Prototyping of Games and Simulations

Animation sketching with K-Sketch could also be the starting point for functional prototypes of simple games and simulations. The creation of games and simulations currently requires programming skill, even when using tools for novices such as Alice [34, 85] and Scratch [107]. It may be possible, however, to build an entire game development environment around the K-Sketch animation interface. Starting a prototype with a sketched animation could give end users with no programming skill the anchors they need for adding behavioral details. It would also give precedence to crafting the concrete visual and dynamic aspects of a game, which could facilitate the flow of ideas and help end users unlock creative potential in this area.

Field studies and interface optimization have already begun for this project [33]. The animation in Figure 9-2 was created by a child prototyping a game, as was the animation in Figure 8-2 on page 281. Both were collected during the child animation study mentioned in Table 8-1 on page 279, in which 21 children used sketched animation to make their game ideas tangible. Other field work collected a scenario library of 14 games and 13 simulations, from which a set of 84 operations were defined. This project is yet another example of how K-Sketch will make animation useful in ways never dreamt of previously.

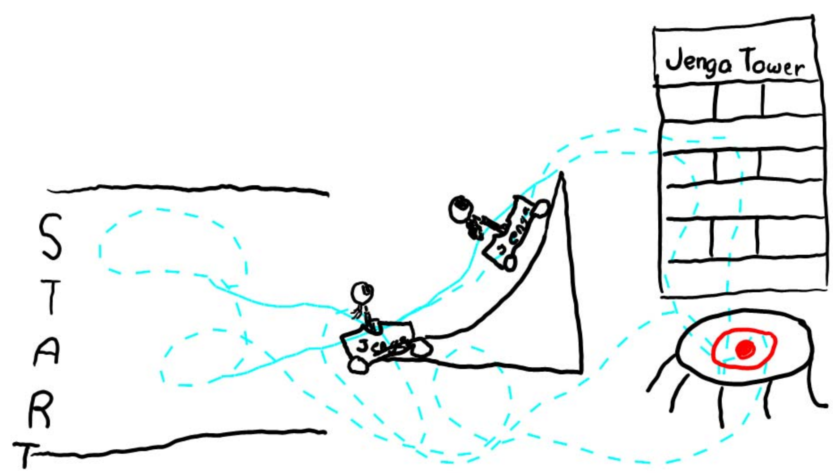


Figure 9-2: A driving game prototype created with K-Sketch. Sketched animations such as this one could be a starting point for a functioning game prototype. View this animation online at http://www.k-sketch.org/dissertation/fig9-2.

9.5 Managing Ideas Throughout Their Life Cycle

As a final example of future work, consider that sketching in an integral part of the design process for every medium, but few software tools can make use of rough sketches. An artificial wall currently exists between users' sketches and the finished artifacts produced from them. This boundary is difficult to cross, and it causes ideas to become fragmented and unmanageable.

For example, imagine that a 3D animator sketches a plane in a notebook (or in an electronic notebook like NotePals [42], as in Figure 9-3a). Then she imports this sketch into a 2D animation tool (e.g., Flash) to refine some parts and add some movement. Later, she wishes to show her clients a 3D mock-up, so she reconstructs her sketch in a complex 3D modeling tool, as in Figure 9-3b. She may modify her plane model further in the 3D tool, but it is hard to bring this modified version back into the 2D animation tool or the notebook where it can be browsed alongside other designs. Over time, her design ideas get spread out between many files, and consolidating them can become a time consuming process.

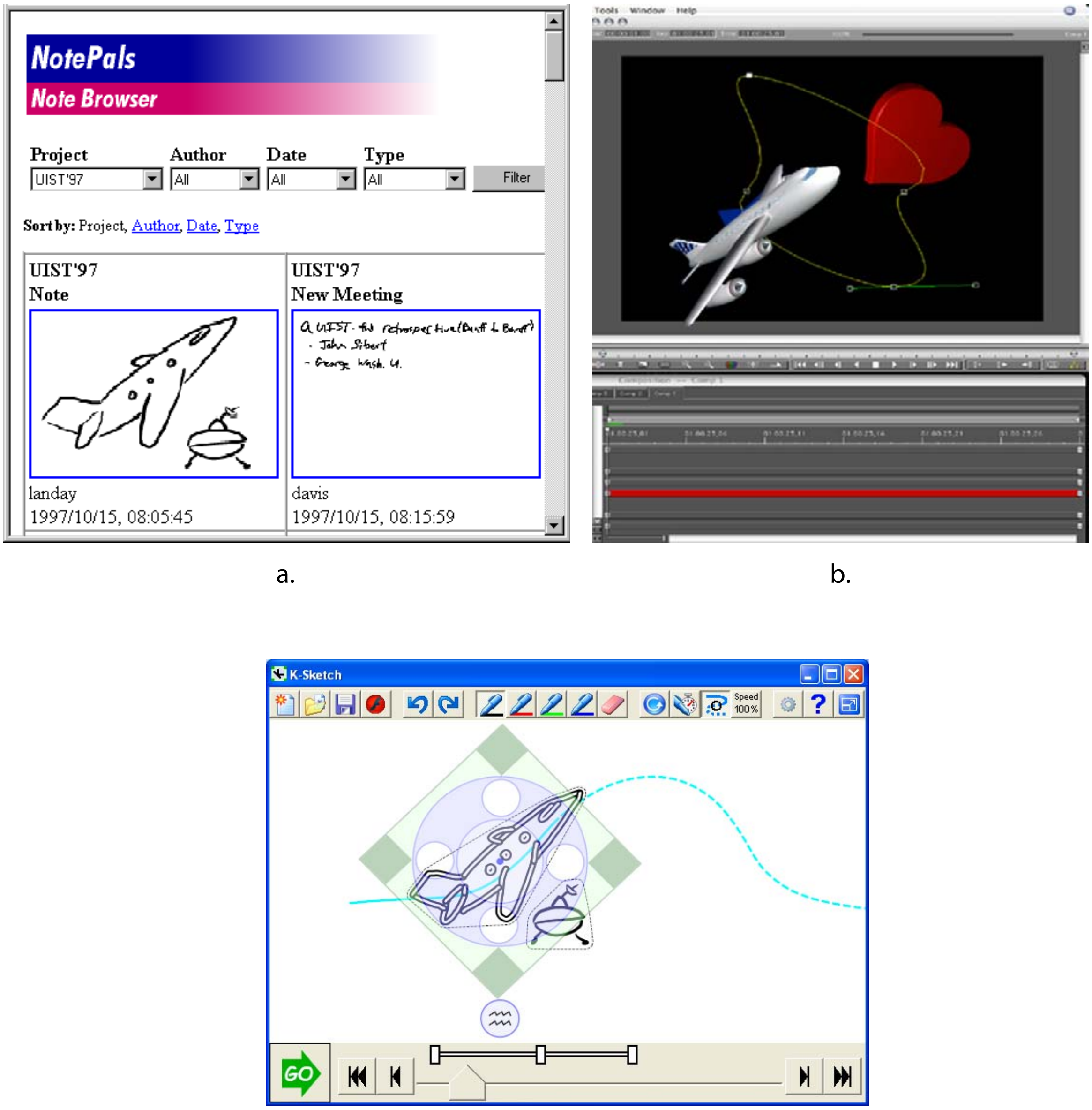


Figure 9-3: An illustration of how K-Sketch blurs the boundary between rough sketches and polished documents. *a:* A sketch of a plane in an electronic notebook. *b:* A 3D animation of a plane based on the original sketch. *c:* A K-Sketch animation that adds rough animation behavior to the original sketch.

K-Sketch blurs the boundary between rough sketches and the animations produced from them and points to a better approach to managing documents. With K-Sketch, an animator can move seamlessly from brainstorming character ideas to working out character motions and back again (see Figure 9-3c). In the future, electronic notebooks may allow users to add many types of semantics to their sketches, blurring the boundary between sketches and artifacts in many media. After semantics have been added, these notebooks may also help users transition their sketches into more polished documents for final presentation. Furthermore, such electronic notebooks could help users manage their notes and documents, maintaining connections between different versions of an idea in different levels of detail. For computer users who get lost in the many computer files they produce, an idea management system of this kind could mean a large leap in productivity.

These are a few examples of continuing research enabled by this dissertation research. K-Sketch itself will continue to evolve, as will the interface optimization technique. Even so, The K-Sketch interface can already be integrated into other applications to enable new uses for animation. Finally, the way K-Sketch merges rough sketches and semantics may inspire similar systems that blur the boundary between rough sketches and polished works, helping users to manage large collections of interconnected ideas.

10 Conclusion

In this dissertation I have argued that numerous people have a desire to create animation but no practical means for doing so. The ubiquity of animation in our culture conditions us to think in moving pictures and also inspires us to communicate in this medium. Current animation tools, however, are not up to the challenge of supporting widespread use of animation. Some tools allow users to work quickly, but they are narrow in focus. Similarly, users desiring a simple tool can reach for a flip-book style animation program, but creating animations frame by frame is too slow. Finally, general purpose animation tools can be used to express a variety of animations, but their complex interfaces make them inaccessible to most users. The vast majority of computer users are novices when it comes to animation, and novice animators need a tool that balances these three needs: speed, simplicity, and expressivity.

The central thesis of this dissertation has been that **an informal, sketch-based animation system designed through analysis of usage scenarios will allow novices to create a wide variety of animations quickly and easily and to use animation in new ways.** K-Sketch is the system that realizes this vision. Informality and sketching are key elements to this vision, because they help users avoid focusing on unnecessary details that distract them from their more important goals. Analysis of usage scenarios with interface optimization is another key element, because the conflicting requirements of speed, simplicity, and expressivity required me to strike a delicate balance between these requirements. The final system does balance these needs, and in doing so it enables new uses of animation, such as fast prototyping of interfaces and powerful new science learning exercises.

10.1 Contributions

Let us revisit the major contributions of this dissertation.

1. *Concepts and Techniques for designing user interfaces in general and for animation in particular.*

I began this research with field studies into the needs of animators and would-be animators. Through this process, I collected a library of 72 detailed usage scenarios for an informal animation tool. These scenarios captured users' background and goals, as well as the abstract operations that might be used to accomplish those goals. The scenario library and the 18 animation operations will serve as guides to future research and development in informal animation sketching (see Chapter 3).

I then developed a new analysis technique, interface optimization, that helped me choose a smaller subset of 10 animation operations that still allowed users to create a large number of scenarios quickly, as described in Chapter 4. The technique begins with breaking down scenarios into essential features and then breaking down features into various approaches to representing those features with a given set of operations. This produces a coded library of scenarios that can be processed to produce visualizations revealing the speed, simplicity, and expressivity tradeoffs associated with each operation. The technique and the processing programs are easily generalizable to other domains, giving all interface designers a powerful new way to discover such design tradeoffs.

2. *Artifacts that provide immediate benefit to novice animators and guide future research and development of animation tools.*

K-Sketch's user interface design, described in Chapter 5, combines powerful interaction techniques in a new way. Animation is modeled as a sequence of editing steps over time. The use of demonstrated motion allows many small operations to be combined into one fluid motion. K-Sketch's simplified timeline avoids showing detailed information for each moving object, instead showing an iconic view of edit history that highlights events related to the currently selected object. K-Sketch also supports motions in multiple frames of reference without requiring users to navigate through a confusing hierarchy of reference frames. Instead, the system uses heuristics to determine the reference frame of new motions and gives users a suggestive interface for choosing the correct reference frame when it makes the wrong choice.

Like the library of scenarios, these design ideas also serve as a guide to future research and development in informal animation sketching.

An implementation of this K-Sketch design has been available for public download since February of 2008. This implementation was downloaded more than 1000 times in the first eight months after it was made available. Reports describe a range of uses for K-Sketch animations, including conference presentations, science learning exercises, interface prototyping, and good clean fun (see Chapter 8).

3. Study results that show the benefits of using an informal animation tool.

The power of K-Sketch's approach to animation was evident both in the laboratory and in the field. Laboratory experiments compared K-Sketch to two commercial tools. The first laboratory experiment compared against Microsoft PowerPoint and focused on K-Sketch's speed and simplicity, as described in the first half of Chapter 7. The sixteen participants in this study completed their animations an average of three times faster with K-Sketch, and they reported that it felt much faster. These speed improvements were due in large part to K-Sketch's informality, as many participants' felt compelled to perfect their PowerPoint animations, even when told not to. Participants also needed less practice time with K-Sketch, reported much less cognitive load, and reported that K-Sketch felt much easier to use. These simplicity improvements were due largely to K-Sketch's simpler animation model and timeline. Ironically, participants were no less comfortable sharing their rough K-

Sketch animations with others, and they were more comfortable creating them while being watched by others. This is a sign that K-Sketch may enable spontaneous collaboration with sketched animation.

The second laboratory study, described in the second half of Chapter 7, focused on expressivity and compared K-Sketch to The TAB Lite. This tool was simpler than PowerPoint, and it could be used in an informal way, allowing me to compare two animation interfaces (K-Sketch and the TAB Lite) that had roughly equivalent speed and simplicity. The two tools supported very different animation operations, however, making it possible to focus my comparison on expressivity. Seven participants completed up to nine tasks with both tools. Six of these tasks were better supported by K-Sketch's set of operations, and participants completed them two to three times faster. Other tasks were completed in about the same time with both tools. These results give weight to my claim that K-Sketch is the more expressive tool. Users also needed to do less planning on paper and less mathematical calculations when creating animations with K-Sketch.

Finally, observations of K-Sketch use in the real world, described in Chapter 8, show that these benefits exist outside the laboratory. An analysis of animation files produced by users found more signs of K-Sketch's speed, simplicity, and expressivity. Furthermore, two case studies with novice animators demonstrated that K-Sketch is enabling new uses of animation. In the first case study, a graduate student interface designer with no animation skills was able to produce 25 animated prototypes for a new interaction technique within three weeks of encountering K-Sketch. She commented that the animations helped focus her discussions with her advisor, and that she never would have completed so many animations so fast with another tool. In the second case study, over a hundred teenage science students created highly detailed animations in multiple learning exercises. Students displayed better recall than they did before encountering K-Sketch, and their animations effectively revealed their misconceptions to their teacher.

10.2 Toward Ubiquitous Animation

The ultimate goal of this research is to make animation ubiquitous, so that every computer user will feel comfortable creating an animation whenever the need arises. Just as the word processor became ubiquitous and its interface became second nature to most computer users, it is my hope that K-Sketch will help make animation second nature as well. For this to happen, users must learn how to formulate ideas in both space and time. K-Sketch facilitates such learning in the same way that sketching helps people learn how to draw: it removes unnecessary details and focuses their attention on what they are trying to create. And when users become so proficient that they can produce highly polished animations, K-Sketch will still be there for sketching out quick prototypes. One day, it may even be possible to seamlessly move from sketched animations to more formal animations, removing even more unnecessary burdens and freeing new animators to realize their full potential.

Bibliography

1 Abowd GD, Atkeson CG, Brotherton J, Enqvist T, Gulley P, LeMon J. Investigating the capture, integration and access problem of ubiquitous computing in an educational setting. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 440-47. Los Angeles, CA, April 1998.

2 Abowd GD, Atkeson CG, Feinstein A, Hmelo C, Kooper R, Long S, Sawhney N, Tani M. Teaching and learning as multimedia authoring: the classroom 2000 project. *Proceedings of the ACM International Conference on Multimedia*, pp 187-98. Boston, MA, November 1996.

3 Accot J, Zhai S. Beyond Fitts' law: models for trajectory-based HCI tasks. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 295-302. Atlanta, GA, March 1997.

4 Accot J, Zhai S. More than dotting the i's – foundations for crossing-based interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 73-80. Minneapolis, MN, April 2002.

5 Adobe Systems Incorporated, *AfterEffects*. http://www.adobe.com/aftereffects/

6 Adobe Systems Incorporated, *Flash*. http://www.adobe.com/flash/

7 AgentSheets Inc., *Agent Sheets*. http://www.agentsheets.com/

8 Alvarado C, Davis R. Resolving ambiguities to create a natural sketch based interface. *Proceedings of the International Joint Conference on Artificial Intelligence*, pp 1365-71. Seattle, WA, August 2001.

9 Alvarado CJ, *A Natural Sketching Environment: Bringing the Computer into Early Stages of Mechanical Design*, Unpublished master's thesis, EECS Department,

Massachusetts Institute of Technology, Cambridge, MA, May 2000. http://rationale.csail.mit.edu/publications/Alvarado2000Natural.ps

10 Anderson R, Anderson R, Davis P, Linnell N, Prince C, Razmov V, Videon F. Classroom Presenter: enhancing interactive education with digital ink. *IEEE Computer,* 40(9):56-61, 2007.

11 Anderson RJ, Hoyer C, Wolfman SA, Anderson R. A study of digital ink in lecture presentation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 567-74. Vienna, Austria, April 2004.

12 Aniboom, *aniBoom: The Home of Animation*, 2008. http://www.aniboom.com/

13 Animation School Review, *Animation School Review*, 2006. http://www.animationschoolreview.com/

14 Apitz G, Guimbretière F. CrossY: a crossing-based drawing application. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 3-12. Santa Fe, NM 2004.

15 Baecker R, *Interactive Computer-Mediated Animation*, Unpublished doctoral dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA, June 1969.

16 Baecker R. Picture-driven animation. *Proceedings of the AFIPS Spring Joint Computer Conference* (volume 34), pp 273-88. Boston, MA, May 1969.

17 Baecker R. A conversational extensible system for the animation of shaded images. *Proceedings of the Third Annual Conference on Computer Graphics and Interactive Techniques*, pp 32-39. Philadelphia, PA, July 1976.

18 Baecker R, Sorting out sorting: a case study of software visualization for teaching computer science, In *Software Visualization: Programming as a Multimedia Experience,* pp 369-81. Cambridge, MA: MIT Press, 1998.

19 Bailey BP, Konstan JA. Are informal tools better?: comparing DEMAIS, pencil and paper, and authorware for early multimedia design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 313-20. Ft. Lauderdale, FL, April 2003.

20 Bailey BP, Konstan JA, Carlis JV. DEMAIS: designing multimedia applications with interactive storyboards. *Proceedings of the Ninth ACM International Conference on Multimedia*, pp 241-50. Ottawa, ON, Canada, September 2001.

21 Bederson BB, Grosjean J, Meyer J. Toolkit design for interactive structured graphics. *IEEE Transactions on Software Engineering,* 30(8):535-46, 2004.

22 Bertamini M, Proffitt DR. Hierarchical motion organization in random dot configurations. *Journal of Experimental Psychology: Human Perception and Performance,* 26(4):1371-86, 2000.

23 Bier EA, Stone MC, Pier K, Buxton W, DeRose TD. Toolglass and magic lenses: the see-through interface. *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques*, pp 73-80. Anaheim, CA, August 1993.

24 Biever C. Easy animation to get novices jumping for joy, *New Scientist* (2524): p 26, November 5, 2005.

25 Bone P, *Pivot Stickfigure Animator*. http://www.geocities.com/peter_bone_uk/pivot.html

26 Brotherton JA, Abowd GD. Lessons learned from eClass: assessing automated capture and access in the classroom. *ACM Transactions on Computer-Human Interaction,* 11(2):121-55, 2004.

27 Brown MH, Sedgewick R. A system for algorithm animation. *ACM SIGGRAPH Computer Graphics,* 18(3):177-86, 1984.

28 Burke KA, Greenbowe TJ, Windschitl MA. Developing and using conceptual computer animations for chemistry instruction. *Journal of Chemical Education,* 75(12):1658-61, 1998.

29 Buxton B, *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, 2007.

30 Card SK, Moran TP, The Model Human Processor – An Engineering Model of Human Performance, In Boff KR, Kaufman L, Thomas JP (eds), *Handbook of Perception and Human Performance* (volume two), pp 45:1-35. New York: Wiley, 1986.

31 Carnegie Mellon University, *Alice*. http://www.alice.org/

32 Christian W, Belloni M, *Physlet Physics: Interactive Illustrations, Explorations and Problems for Introductory Physics*. Upper Saddle River, NJ: Prentice Hall, 2004.

33 Colwell B, Davis RC, Landay JA, A study of early stage game design and prototyping. Technical report UW-CSE-08-10-03, Computer Science and Engineering Department, University of Washington, Seattle, WA, 2008. http://www.cs.washington.edu/education/grad/UW-CSE-08-10-03.PDF

34 Conway M, Audia S, Burnette T, Cosgrove D, Christiansen K. Alice: lessons learned from building a 3D system for novices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 486-93. The Hague, The Netherlands, April 2000.

35 Cross N, *Designerly Ways of Knowing*. London: Springer, 2006.

36 Csikszentmihalyi M, *Flow : The Psychology of Optimal Experience* (First edition). New York: Harper and Row, 1990.

37 Cypher A, Smith DC. KidSim: end user programming of simulations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 27-34. Denver, CO 1995.

38 Damm CH, Hansen KM, Thomsen M. Tool support for cooperative object-oriented design: gesture based modeling on an electronic whiteboard. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 518 - 25. The Hague, The Netherlands, April 2000.

39 Dancy M, *Investigating Animations for Assessment with an Animated Version of the Force Concept Inventory*, Unpublished doctoral dissertation, Department of Physics, North Carolina State University, Raleigh, NC, 2000.

40 Davis J, Agrawala M, Chuang E, Popović Z, Salesin D. A sketching interface for articulated figure animation. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 320-28. San Diego, CA, July 2003.

41 Davis R. Sketch understanding in design: overview of work at the MIT AI lab. *Proceedings of the AAAI Spring Symposium on Sketch Understanding*, pp 24-31. Palo Alto, CA, March 2002.

42 Davis RC, Landay JA, Chen V, Huang J, Lee RB, Li FC, Lin J, Morrey CB, Schleimer B, Price MN, Schilit BN. NotePals: lightweight note sharing by the group, for the group. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 338-45. Pittsburgh, PA, May 1999.

43 Davis RC, Saponas TS, Shilman M, Landay JA. SketchWizard: Wizard of Oz prototyping of pen-based user interfaces. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 119-28. Newport, RI 2007.

44 Davis SB, Moar M. The amateur creator. *Proceedings of the ACM Conference on Creativity and Cognition*, pp 158-65. London, UK, April 2005.

45 Dekel U, Herbsleb JD. Notation and representation in collaborative object-oriented design: an observational study. *ACM SIGPLAN Notices,* 42(10):261-80, 2007.

46 Design Simulation Technologies Inc., *Working Model 2D*. http://www.design-simulation.com/WM2D

47 Di Fiore F, Van Reeth F. A multi-level sketching tool for 'pencil-and-paper' animation. *Proceedings of the AAAI Spring Symposium on Sketch Understanding*, pp 32-36. Palo Alto, CA, March 2002.

48 Digital Video s.r.l., *The TAB Lite*. http://www.tablite.com/

49 Dontcheva M, Yngve G, Popović Z. Layered acting for character animation. *ACM Transactions on Graphics,* 22(3):409-16, 2003.

50 Edwards WK, Igarashi T, LaMarca A, Mynatt ED. A temporal model for multi-level undo and redo. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 31-40. San Diego, CA, November 2000.

51 Edwards WK, Mynatt ED. Timewarp: techniques for autonomous collaboration. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 218-25. Atlanta, GA, March 1997.

52 Fekete J-D, Bizouarn É, Cournarie É, Thierry G, Taillefer F. TicTacToon: a paperless system for professional 2D animation. *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques*, pp 79-90. Los Angeles, CA, August 1995.

53 Ferguson ES, *Engineering and the Mind's Eye* (First edition). Cambridge, MA: MIT Press, 1992.

54 Fitzmaurice G, Khan A, Pieké R, Buxton B, Kurtenbach G. Tracking menus. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 71-79. Vancouver, BC, Canada, November 2003.

55 Francik E, Rudman SE, Cooper D, Levine S. Putting innovation to work: adoption strategies for multimedia communication systems. *Communications of the ACM,* 34(12):52-63, 1991.

56 Frederick BN, Fixed-, random-, and mixed-effects ANOVA models: A user-friendly guide for increasing the generalizability of ANOVA results, In

Thompson B (ed) *Advances in Social Science Methodology,* pp 111-22. Stamford, CT: JAI Press, 1999.

57 Gleicher M. A graphics toolkit based on differential constraints. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 109-20. Atlanta, GA, November 1993.

58 Goel V, *Sketches of Thought*. Cambridge, MA: MIT Press, 1995.

59 GoKnow Learning Inc., *Sketchy*. http://www.goknow.com/Products/Sketchy/

60 Goodstein DL, The California Institute of Technology, The Southern California Consortium for Community College Television, *The Mechanical Universe – and Beyond*. Annenberg/CPB Project, 1985-1987.

61 Goulekas KE, *Visual Effects in a Digital World: A Comprehensive Glossary of over 7,000 Visual Effects Terms*. San Diego, CA: Morgan Kaufmann, 2001.

62 Gross MD, Do EY. Ambiguous intentions: a paper-like interface for creative design. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 183-92. Seattle, WA, November 1996.

63 Grossman T, Hinckley K, Baudisch P, Agrawala M, Balakrishnan R. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. *Proceedings of the SIGCHI conference on human factors in computing systems*, pp 861-70. Montréal, Québec, Canada, April 2006.

64 Guiard Y. Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *Journal of Motor Behavior,* 19(4):486-517, 1987.

65 Hart SG, Staveland LE, Development of the NASA-TLX (task load index): results of empirical and theoretical research, In Hancock PA, Meshkati N (eds), *Human Mental Workload,* pp 139-77. North-Holland: Elsevier, 1988.

66 Hegarty M, Kriz S, Cate C. The roles of mental animations and external animations in understanding mechanical systems. *Cognition and Instruction,* 21(4):325-60, 2003.

67 Hegarty M, Mayer S, Kriz S, Keehner M. The role of gestures in mental animation. *Spatial Cognition and Computation,* 5(4):333 - 56, 2005.

68 Heider F, Simmel M. An experimental study of apparent behavior. *American Journal of Psychology,* 57(2):243-59, 1944.

69 Hinckley K, Baudisch P, Ramos G, Guimbretiere F. Design and analysis of delimiters for selection-action pen gesture phrases in Scriboli. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 451-60. Portland, OR, April 2005.

70 Honda M, Igarashi T, Tanaka H, Sakai S. Integrated manipulation: context-aware manipulation of 2D diagrams. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 159-60. Asheville, NC, November 1999.

71 Hong JI, Landay JA. SATIN: a toolkit for informal ink-based applications. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 63-72. San Diego, CA, August 2000.

72 Hundhausen CD, Douglas SA, Stasko JT. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing,* 13(3):259-90, 2002.

73 Igarashi T, Edwards WK, LaMarca A, Mynatt ED. An architecture for pen-based interaction on electronic whiteboards. *Proceedings of the Working Conference on Advanced Visual Interfaces*, pp 68-75. Palermo, Italy, May 2000.

74 Igarashi T, Hughes JF. A suggestive interface for 3D drawing. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 173-81. Orlando, FL, November 2001.

75 Igarashi T, Moscovich T, Hughes JF. As-rigid-as-possible shape manipulation. *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques*, pp 1134-41. Los Angeles, CA, July 2005.

76 Jarrett R, Su P, *Building Tablet PC Applications*. Redmond, WA: Microsoft Press, 2003.

77 Kahn PH, *The Human Relationship With Nature: Development and Culture*. Cambridge, MA: MIT Press, 1999.

78 Kaiser MK, Proffitt DR. The development of sensitivity to causally relevant dynamic information. *Child Development,* 55(4):1614-24, 1984.

79 Kaiser MK, Proffitt DR, Anderson K. Judgments of natural and anomalous trajectories in the presence and absence of motion. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 11(4):795-803, 1985.

80 Kaiser MK, Proffitt DR, Wheland SM, Hecht H. Judgments of natural and anomalous trajectories in the presence and absence of motion. *Journal of Experimental Psychology: Human Perception and Performance,* 18(3):669-90, 1992.

81 Kam M, Wang J, Iles A, Tse E, Chiu J, Glaser D, Tarshish O, Canny J. Livenotes: a system for cooperative and augmented note-taking in lectures. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 531-40. Portland, OR, April 2005.

82 Kara LB, Gennari L, Stahovich TF. A sketch-based interface for the design and analysis of simple vibratory mechanical systems. *Proceedings of the American Society of Mechanical Engineers International Design Engineering Technical Conferences (ASME / DETC'04)* (volume one), pp 1047-57. Salt Lake City, UT, September 2004.

83 Karsenty S, Landay JA, Weikart C. Inferring graphical constraints with Rockit. *Proceedings of the Conference on People and Computers VII (HCI)*, pp 137-53. York, United Kingdom, September 1992.

84 Kato Y, Shibayama E, Takahashi S. Effect lines for specifying animation effects. *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*, pp 27-34. Rome, Italy, September 2004.

85 Kelleher C, Pausch R, Kiesler S. Storytelling Alice motivates middle school girls to learn computer programming. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 1455-64. San Jose, CA, April 2007.

86 Kerren A, Stasko JT, Algorithm Animation - Introduction, In Diehl S (ed) *Revised Lectures on Software Visualization, Lecture Notes in Computer Science* (volume 2269), pp 1-15. London: Springer-Verlag, 2002.

87 Klemmer SR, Thomsen M, Phelps-Goodman E, Lee R, Landay JA. Where do web sites come from?: capturing and interacting with design history. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 1-8. Minneapolis, MN, April 2002.

88 Kohavi R, Henne RM, Sommerfield D. Practical guide to controlled experiments on the web: listen to your customers not to the HiPPO. *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 959-67. San Jose, CA, August 2007.

89 Kramer A. Translucent patches—dissolving windows. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 121-30. Marina del Rey, CA, November 1994.

90 Kriz S, Hegarty M. Top-down and bottom-up influences on learning from animations. *International Journal of Human-Computer Studies,* 65(11):911-30, 2007.

91 Kurlander D, Feiner S. A history-based macro by example system. *Proceedings of the ACM symposium on User Interface Software and Technology*, pp 99-106. Monterey, CA, November 1992.

92 Kurtenbach G, Buxton W. The limits of expert performance using hierarchical marking menus. *Proceedings of the INTERACT and SIGCHI Conference on Human Factors in Computing Systems*, pp 482 - 87. Amsterdam, The Netherlands, April 1993.

93 Kurtenbach G, Buxton W. User learning and performance with marking menus. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 258-64. Boston, MA, April 1994.

94 Landay JA, *Interactive Sketching for the Early Stages of User Interface Design*, Unpublished doctoral dissertation, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, December 1996. http://reports-archive.adm.cs.cmu.edu/anon/anon/usr/ftp/1996/CMU-CS-96-201.pdf

95 Landay JA, Myers BA. Sketching interfaces: toward more human interface design. *IEEE Computer,* 34(3):56-64, 2001.

96 Lank E, Saund E, May L. Sloppy selection: providing an accurate interpretation of imprecise selection gestures. *Proceedings of the AAAI Fall Symposium on Making Pen-Based Interaction Intelligent and Natural*, pp 106-12. Arlington, VA, October 2004.

97 LaViola JJ, Zeleznik RC. MathPad^2: a system for the creation and exploration of mathematical sketches. *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques*, pp 432-40. Los Angeles, CA, August 2004.

98 Li Y, Hinckley K, Guan Z, Landay JA. Experimental analysis of mode switching techniques in pen-based user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 461-70. Portland, OR, April 2005.

99 Li Y, Hong JI, Landay JA. Topiary: a tool for prototyping location-enhanced applications. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 217-26. Santa Fe, NM, October 2004.

100 Lilipip Inc., *Lilipip!: Feeding Curiosity Daily*, 2008. http://www.lilipip.com

101 Lin J, Newman MW, Hong JI, Landay JA. DENIM: finding a tighter fit between tools and practice for web site design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 510-17. The Hague, The Netherlands, April 2000.

102 Littell RC, Milliken GA, Stroup WW, Wolfinger RD, *SAS System for Mixed Models*. Cary, NC: SAS Institute, Inc., 1996.

103 Lowe RK. Animation and learning: Value for money? *Proceedings of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE) Conference*, pp 558-61. Perth, Australia, December 2004.

104 Mankoff J, Abowd GD, Hudson SE. OOPS: a toolkit supporting mediation techniques for resolving ambiguity in recognition-based interfaces. *Computers and Graphics,* 24(6):819-34, 2000.

105 Michotte A, *The Perception of Causality*. Translated by Miles TR, Miles E. London: Methuen, 1963.

106 Microsoft Corporation, *PowerPoint*. http://www.microsoft.com/powerpoint/

107 MIT Media Lab, *Scratch*. http://scratch.mit.edu/

108 Moran TP, Chiu P, Harrison S, Kurtenbach G, Minneman S, van Melle W. Evolutionary engagement in an ongoing collaborative work process: a case study. *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pp 150-59. Boston, MA 1996.

109 Moran TP, Chiu P, van Melle W, Kurtenbach G. Implicit structure for pen-based systems within a freeform interaction paradigm. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 487-94. Denver, CO, May 1995.

110 Moran TP, Palen L, Harrison S, Chiu P, Kimber D, Minneman S, van Melle W, Zellweger P. "I'll get that off the audio": a case study of salvaging multimedia meeting records. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 202-09. Atlanta, GA, March 1997.

111 Moran TP, van Melle W, Chiu P. Spatial interpretation of domain objects integrated into a freeform electronic whiteboard. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 175-84. San Francisco, CA, November 1998.

112 Moran TP, van Melle W, Chiu P. Tailorable domain objects as meeting tools for an electronic whiteboard. *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pp 295-304. Seattle, WA, November 1998.

113 Moscovich T, *Animation Sketching: An Approach to Accessible Animation*, Unpublished master's thesis, Computer Science Department, Brown University, Providence, RI, 2001. http://www.cs.brown.edu/publications/theses/masters/2001/tm.pdf

114 Moscovich T, Hughes JF, Animation sketching: an approach to accessible animation. Technical report CS-04-03, Computer Science Department, Brown University, Providence, RI, 2004. ftp://ftp.cs.brown.edu/pub/techreports/04/cs04-03.pdf

115 Myers BA, Kosbie DS. Reusable hierarchical command objects. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 260-67. Vancouver, BC, Canada, April 1996.

116 Mynatt ED, Igarashi T, Edwards WK, LaMarca A. Flatland: new dimensions in office whiteboards. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp 346-53. Pittsburgh, PA, May 1999.

117 Newman MW, Lin J, Hong JI, Landay JA. DENIM: an informal web site design tool inspired by observations of practice. *Human-Computer Interaction,* 18(3):259-324, 2003.

118 Newman W, Wellner P. A desk supporting computer-based interaction with paper documents. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 587-92. Monterey, CA, May 1992.

119 Norman DA, Draper SW, (eds). *User Centered System Design: New Perspectives on Human-computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1986.

120 Palmiter S, Elkerton J. An evaluation of animated demonstrations of learning computer-based tasks. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 257-63. New Orleans, LA, April 1991.

121 Pickering JM, Bhuphaibool DS, LaViola JJ, Pollard NS, The coach's playbook. Technical report CS-99-08, Computer Science Department, Brown University, Providence, RI, 1999. ftp://ftp.cs.brown.edu/pub/techreports/99/cs99-08.pdf

122 Price MN, Schilit BN, Golovchinsky G. XLibris: the active reading machine. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 22-23. Los Angeles, CA, April 1998.

123 Rebetez C, Sangin M, Bétrancourt M, Dillenbourg P. Learning from animation enabled by collaboration. *Instructional Science*, (in press).

124 Repenning A, *Agentsheets: A Tool for Building Domain-Oriented Dynamic, Visual Environments*, Unpublished doctoral dissertation, Department of Computer Science, University of Colorado, Boulder, CO, December 1993. http://www.cs.colorado.edu/~ralex/papers/PDF/Repenning-PhD.pdf

125 Repenning A, Perrone C, Programming by analogous examples, In Lieberman H (ed) *Your Wish is My Command: Programming by Example,* pp 351-69. San Francisco, CA: Morgan Kaufmann, 2001.

126 Resnick M. All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. *Proceedings of the SIGCHI Conference on Creativity and Cognition*, pp 1-6. Washington, DC, June 2007.

127 Rettig M. Prototyping for tiny fingers. *Communications of the ACM,* 37(4):21-27, 1994.

128 Rhyne JR, Wolf CG. Tools for supporting the collaborative process. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 161-70. Monterey, CA, December 1992.

129 Rieber LP, *Computers, Graphics, and Learning*. Madison, WI: Brown and Benchmark, 1994.

130 Rogers B. Living Ink: implementation of a prototype sketching language for real time authoring of animated line drawings. *Proceedings of the Eurographics Workshop on Sketch-based Interfaces and Modeling*, pp 115-22. Vienna, Austria, September 2006.

131 Russell JW, Kozma RB, Jones T, Wykoff J, Marx N, Davis J. Use of simultaneous-synchronized macroscopic, microscopic, and symbolic representations to enhance the teaching and learning of chemical concepts. *Journal of Chemical Education,* 74(3):330-34, 1997.

132 Sangin M, Molinari G, Dillenbourg P, Rebetez C, Bétrancourt M. Collaborative learning with animated pictures: the role of verbalizations. *Proceedings of International Conference on Learning Sciences*, pp 667-73. Bloomington, IN, June 2006.

133 Sangin M, Rebetez C, Bétrancourt M, Dillenbourg P. The effects of animations on verbal interaction in computer supported collaborative learning. *Journal of Computer Assisted Learning*, (in press).

134 Saund E, Lank E. Stylus input and editing without prior selection of mode. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 213-16. Vancouver, BC, Canada, November 2003.

135 Saund E, Moran TP. A perceptually-supported sketch editor. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 175-84. Marina del Rey, CA, November 1994.

136 Schuster C, von Eye A. The relationship of ANOVA models with random effects and repeated measurement designs. *Journal of Adolescent Research,* 16(2):205-20, 2001.

137 Shipman FM, Hsieh H. Navigable history: a reader's view of writer's time. *New Review of Hypermedia and Multimedia,* 6(1):147-67, 2000.

138 Shneiderman B. Creating creativity: user interfaces for supporting innovation. *ACM Transactions on Computer-Human Interaction,* 7(1):114-38, 2000.

139 Stagecast Software Inc., *Stagecast Creator*. http://www.stagecast.com/creator.html

140 Stasko JT. Tango: a framework and system for algorithm animation. *IEEE Computer,* 23(9):27-39, 1990.

141 Sutherland IE, *Sketchpad, A Man-Machine Graphical Communication System*, Doctoral dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA, January 1963. http://hdl.handle.net/1721.1/14979

142 Sutherland IE. Sketchpad: a man-machine graphical communication system. *Proceedings of the AFIPS Spring Joint Computer Conference*, pp 329-46. Detroit, MI, May 1963.

143 Takahashi S, Kato Y, Shibayama E. A new static depiction and input technique for 2D animation. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing*, pp 296-98. Dallas, TX, September 2005.

144 Tatar D, Roschelle J, Vahey P, Penuel WR. Handhelds go to school: lessons learned. *IEEE Computer,* 36(9):30-37, 2003.

145 Terra SCL, Metoyer RA. Performance timing for keyframe animation. *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 253-58. Grenoble, France, August 2004.

146 Terry M, Mynatt ED. Side views: persistent, on-demand previews for open-ended tasks. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 71-80. Paris, France, October 2002.

147 Thorne M, Burke D, van de Panne M. Motion doodles: an interface for sketching character motion. *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques*, pp 424-31. Los Angeles, CA, August 2004.

148 Toon Boom Animation Inc., *Animation-ish*. http://www.animationish.com/

149 Toon Boom Animation Inc., *Toon Boom Digital Pro*, 2008. http://www.toonboom.com/products/digitalpro/

150 Tversky B, Morrison JB, Betrancourt M. Animation: can it facilitate? *International Journal of Human-Computer Studies,* 57(4):247-62, 2002.

151 Vronay D, Wang S. Designing a compelling user interface for morphing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 143-49. Vienna, Austria, April 2004.

152 Wellner P. The DigitalDesk calculator: tangible manipulation on a desk top display. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 27-33. Hilton Head, SC, November 1991.

153 Wellner P. Interacting with paper on the DigitalDesk. *Communications of the ACM,* 36(7):87-96, 1993.

154 Whittaker S, Hyland P, Wiley M. FILOCHAT: handwritten notes provide access to recorded conversations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 271-77. Boston, MA, April 1994.

155 Wilcox LD, Schilit BN, Sawhney N. Dynomite: a dynamically organized ink and audio notebook. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 186-93. Atlanta, GA, March 1997.

156 Wobbrock JO, Gajos KZ. Goal crossing with mice and trackballs for people with motor impairments: performance, submovements, and design directions. *ACM Transactions on Accessible Computing (TACCESS),* 1(1):4:1-4:37, 2008.

157 Wolber D, Pavlov: where PBD meets Macromedia's Director, In Lieberman H (ed) *Your Wish is My Command: Programming by Example,* pp 345-50. San Francisco: CA: Morgan Kaufmann, 2001.

158 Wolf CG, Rhyne JR, Ellozy HA. The paper-like interface. *Proceedings of the International Conference on Human-Computer Interaction (Designing and Using Human-Computer Interfaces and Knowledge Based Systems)*, pp 494-501. Boston, MA, September 1989.

159 Wolfram Research Inc., *Mathematica*. http://www.wolfram.com/mathematica

160 Yang E, Andre T, Greenbowe TJ, Tibell L. Spatial ability and the impact of visualization/animation on learning electrochemistry. *International Journal of Science Education,* 25(3):329 - 49, 2003.

161 YouTube LLC, *YouTube: Broadcast Yourself*, 2008. www.youtube.com

162 Zeleznik R, Miller T. Fluid inking: augmenting the medium of free-form inking with gestures. *Proceedings of Graphics Interface*, pp 155-62. Quebec City, QC, Canada, June 2006.

163 Zhao S, Balakrishnan R. Simple vs. compound mark hierarchical marking menus. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 33-42. Santa Fe, NM, October 2004.

164 Zongker DE, Salesin DH. On creating animated presentations. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 298-308. San Diego, CA, July 2003.

APPENDIX

A Field Study Data

This appendix contains a sampling of the raw data I collected during the field studies described in Chapter 3. The data that is not shown here consists mainly of images that either add little value or that the animator did not give permission to show. I begin with animator interviews and follow this with non-animator interviews.

Each interview begins with a short summary. This is followed by notes I took on my initial contact with the animator (or a short description of how contact was made, if no notes were kept). I then present responses to the major questions asked during the interview. In some cases, these responses were reconstructed from other interview data, either because the interview took place before I had finalized the interview format (e.g., Animator 7), or because my contact with the animator was too brief and unstructured (e.g., Animator 8). Finally, I present a sample of the images and artifacts I collected during the interview.

A.1 Animator 1

- Interview date: 2/5/2004
- Occupation: artist, animator
- Primary animation media: clay, key framed drawings
- Scenarios: (prototyping character tests and moving frames of animatics)

    1–Simple potbelly fall (artist, prototype)

    2–Simple potbelly crash (artist, prototype)

    3–King leafy entrance (artist, prototype)

    4–Complex potbelly fall (artist, prototype)

5–Potbelly heads out (artist, prototype)

6–Potbelly collision (artist, prototype)

7–Hitchhiker (artist, prototype)

8–Sea animals (artist, prototype)

A.1.1 Notes on Initial Contact

1/30/2004 Met at Northwest Animator's Social

2/3/2004 Following notes written based on contact

Flash Animations - interactive projects

Working on Puppet interview character design for Boys & Girls clubs

Process

- Brainstorming
- Storyboarding
- Animatics

Seahorses

- Sketches of drawings
- Animatic of drawings w/ limited animation
- Show that animatic

A.1.2 Interview Questions

1) How long have you been working as an animator?

Artist 20 yrs.

Animator 10 yrs.

2) Could you describe the steps in your work process? Give greater detail on the early parts of the process and the parts that involve sketching.

Sketching Conceptualization is a big part. Stir up the mental soup.

Use anything she's got available

Doodles. Loose.

Make most changes up front, because animation is so time intensive

Refined character development / storyboarding

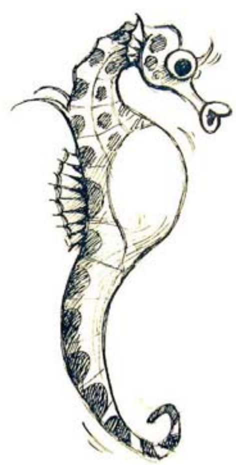
Animatic (w/o voice actors)

Final Art, timing sheet, exposure sheet

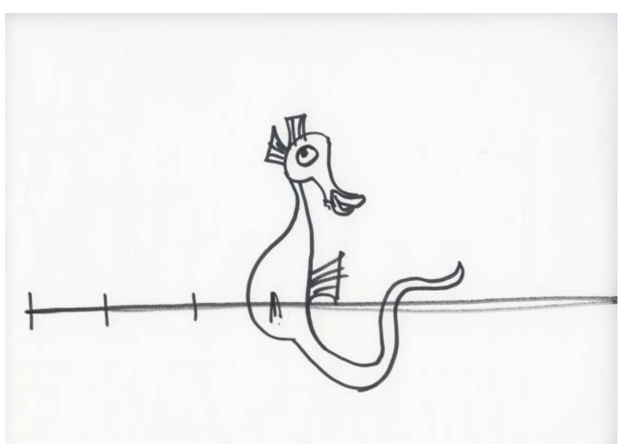
3) What hardware or software tools do you use to do your work?

Sketch Pad, pencils, scanner, Word, Photoshop, Premiere, After Effects, Final Cut Pro, Media Cleaner, DV, digital still camera, sound forge or *pro tools for sound, Flash – another output option.

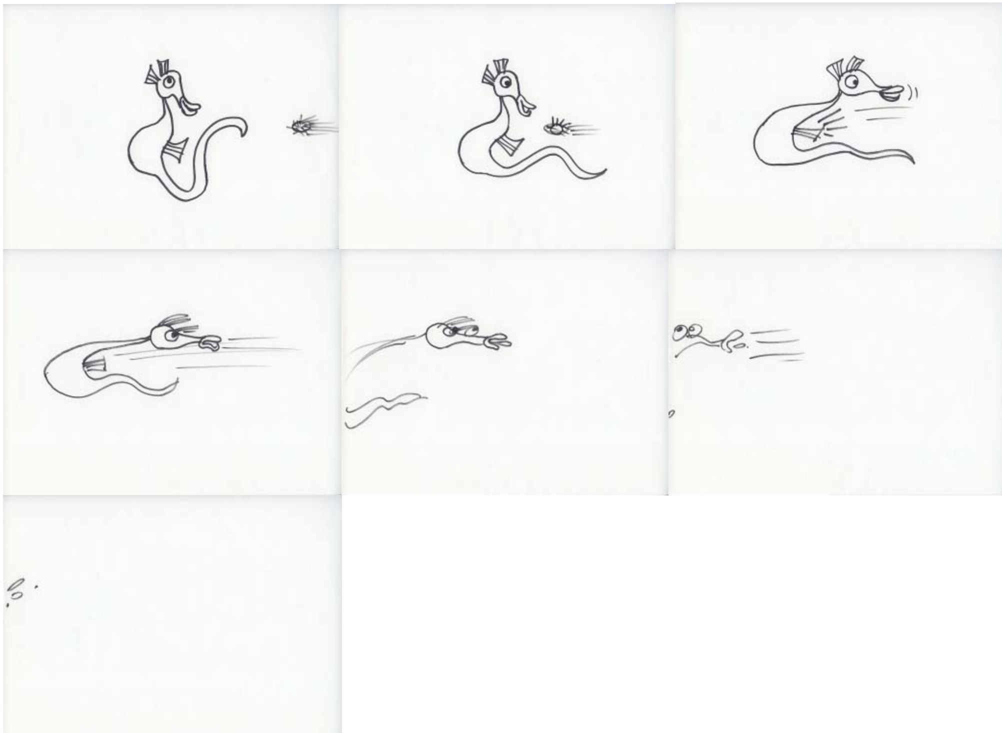
A.1.3 Images and Artifacts



An early character sketch and the final character

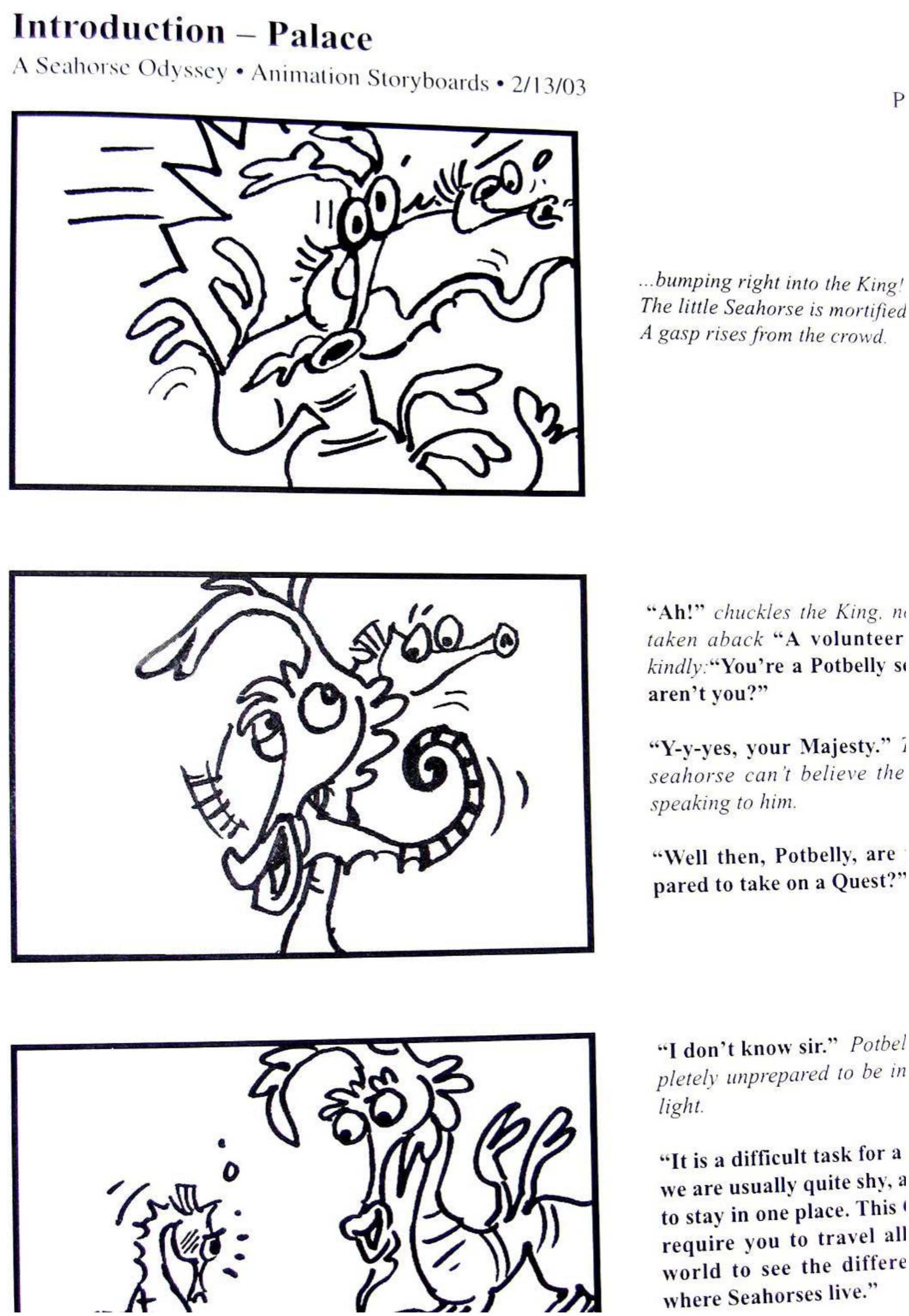


A plan for an animated character sketch

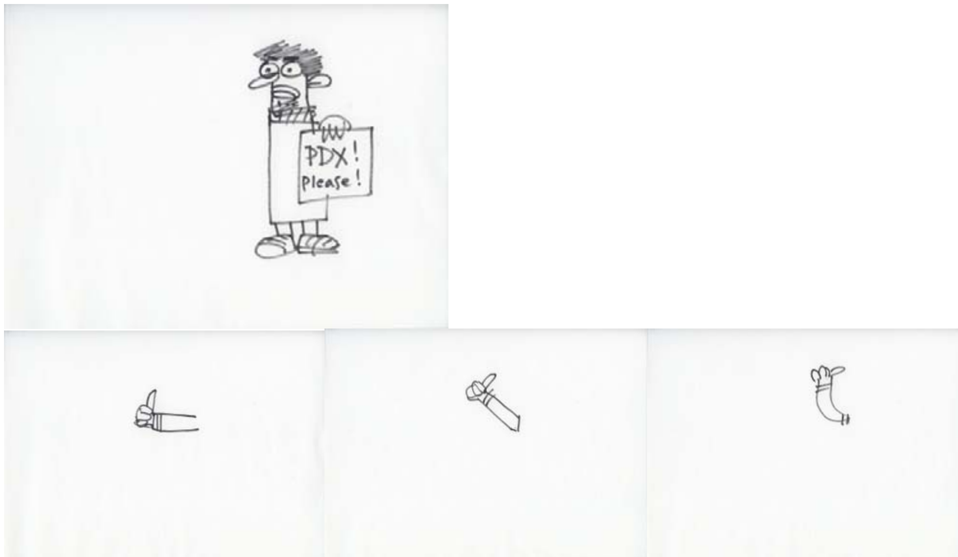


Frames in a moving character sketch (source of 6–*Potbelly collision*)

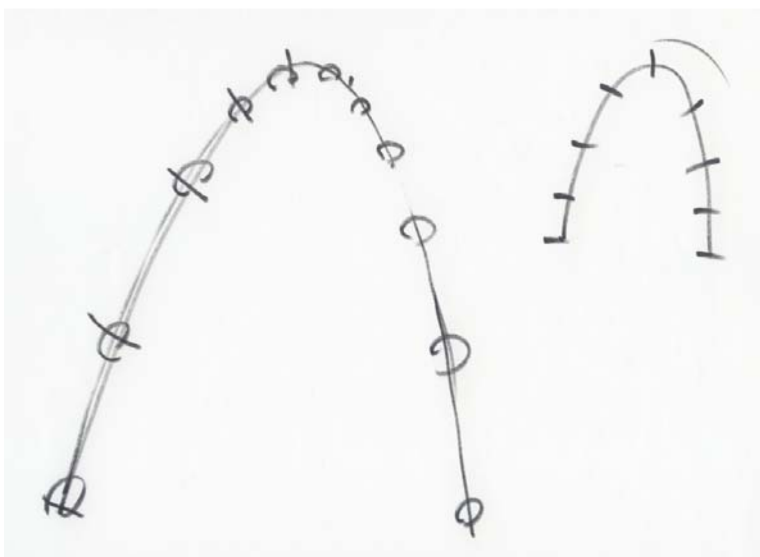
Frames that were discarded when creating the previous moving character sketch



A Seahorse Odyssey storyboard (first frame is source of 2–*Simple potbelly crash*)



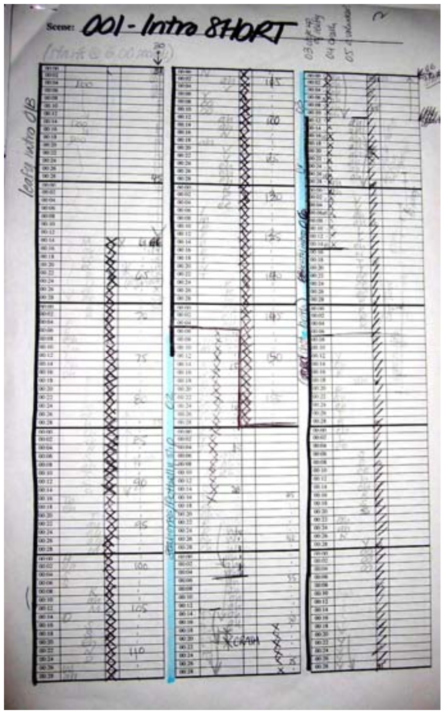
An animated character sketch (source of 7–*Hitchhiker*)

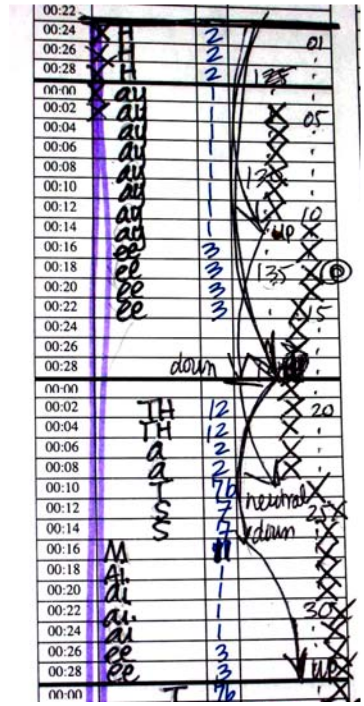


A drawing that illustrates timing and spacing

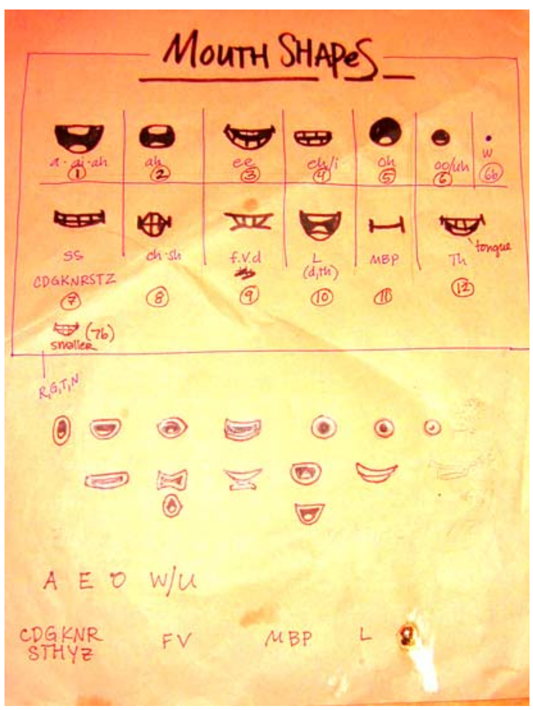
Rough static character sketches

Polished static character sketches





Timing sheet and a close-up view of a different sheet



Mouth shapes reference sheet

A.2 Animator 2

- Interview date: 2/8/2004
- Occupation: animator
- Primary animation media: cel animation
- Scenarios: (prototyping character tests and moving frames of animatics)

  9–Security card slide (artist, prototype)

  10–Perpetrator entrance (artist, prototype)

  11–Perpetrator turning (artist, prototype)

  12–Sorcerer crystal ball (artist, prototype)

  13–Car steering (artist, prototype)

  14–Car on country road (artist, prototype)

  15–Man watching sun (artist, prototype)

  16–Thermometer with more (artist, prototype)

  17–Translate rotate scale (artist, prototype)

  18–Car bobbing (artist, prototype)

  19–Word collision (artist, prototype)

A.2.1 Notes on Initial Contact

1/30/2004 Met at Northwest Animator's Social

2/3/2004 Following notes written based on contact

Play with stories in back of head

"layout" red & blue drawings

Save trouble of exporting to animatic

Liked Disney animator software (old)

Simple morphing

A.2.2 Interview Questions

1) How long have you been working as an animator?

25 years

2) Could you describe the steps in your work process? Give greater detail on the early parts of the process and the parts that involve sketching.

- Character sketch
- Storyboards
- Layouts (for each scene) – working out small details
- Extremes, keys, in-betweens
- Timing sheets

- Computer

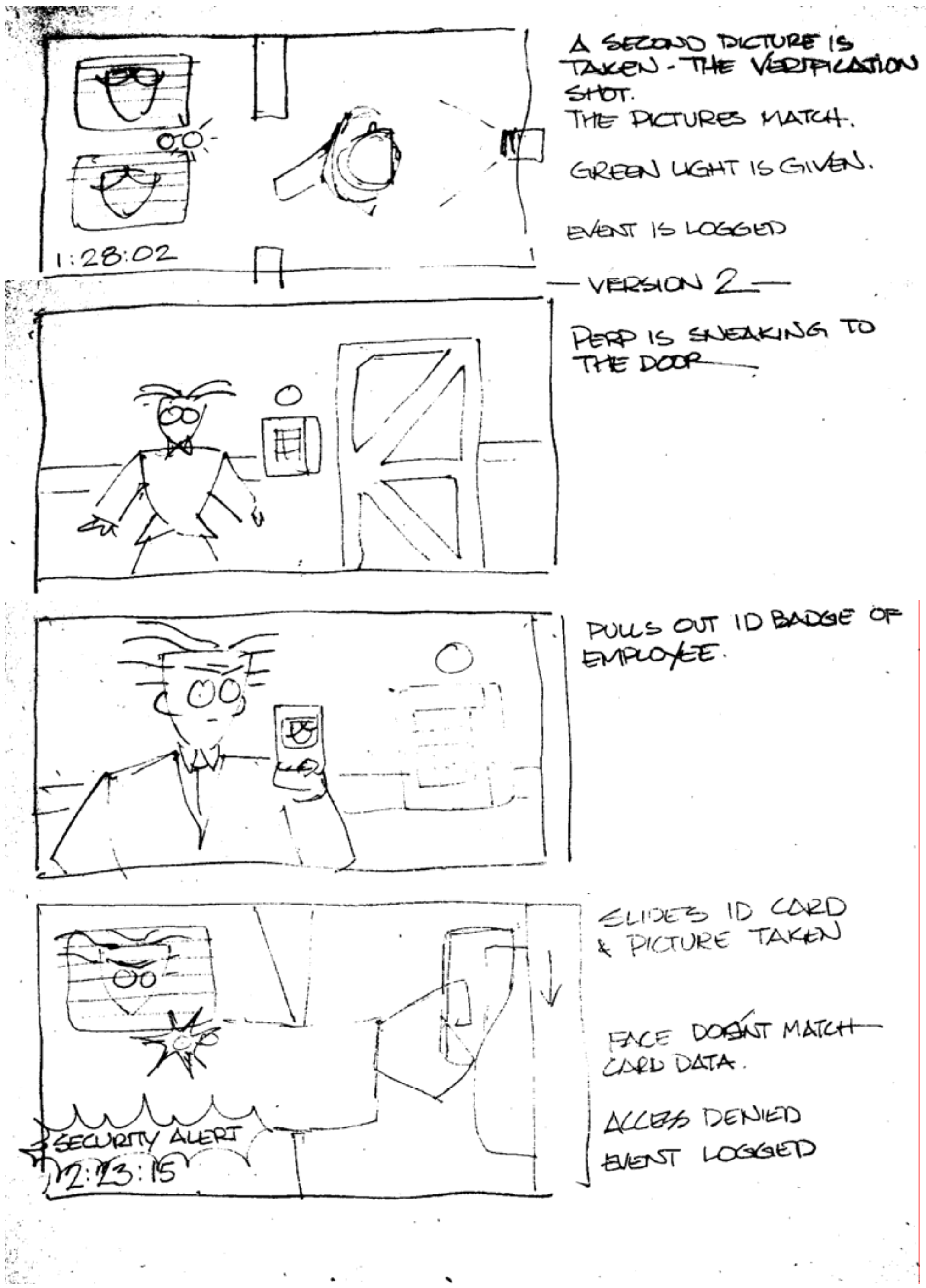
3) What hardware or software tools do you use to do your work?

inking board, animation disk, pencils (Ticonderogas) (red and blue), celluloid (acetate) (sketch on with grease pencils), paint, 16 mm film camera, full coat recorders, cel punches, editing tables

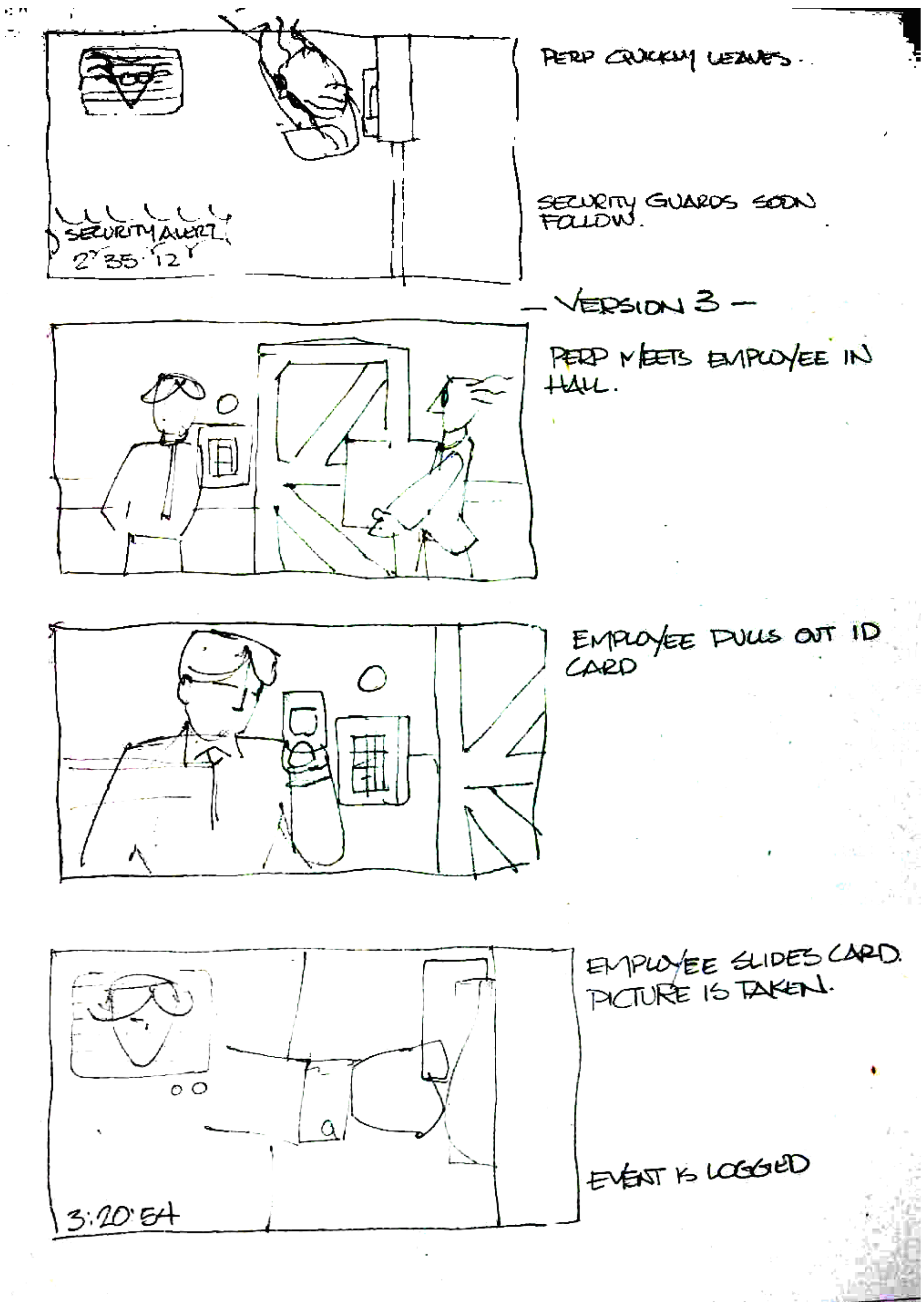
ACME cartoon color, chromacolor

PC, scanner, Wacom Tablet, Photoshop, Painter, AfterEffects, Premiere, Sound Forge, Flash, video input, Bravado card (video capture), The Animation Studio (Disney)

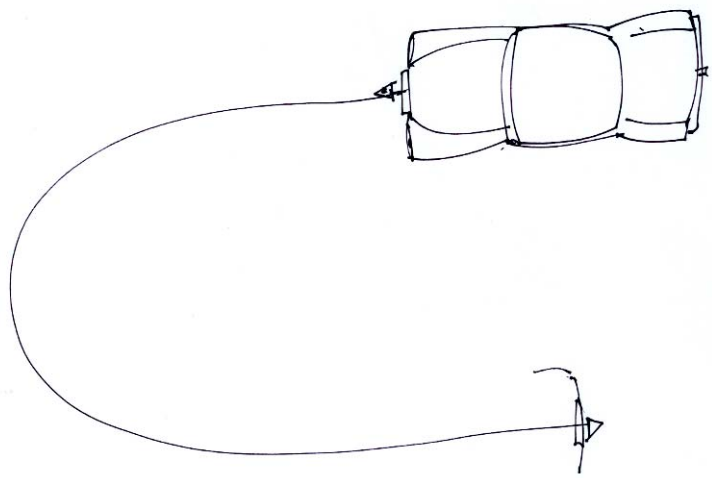
A.2.3 Images and Artifacts



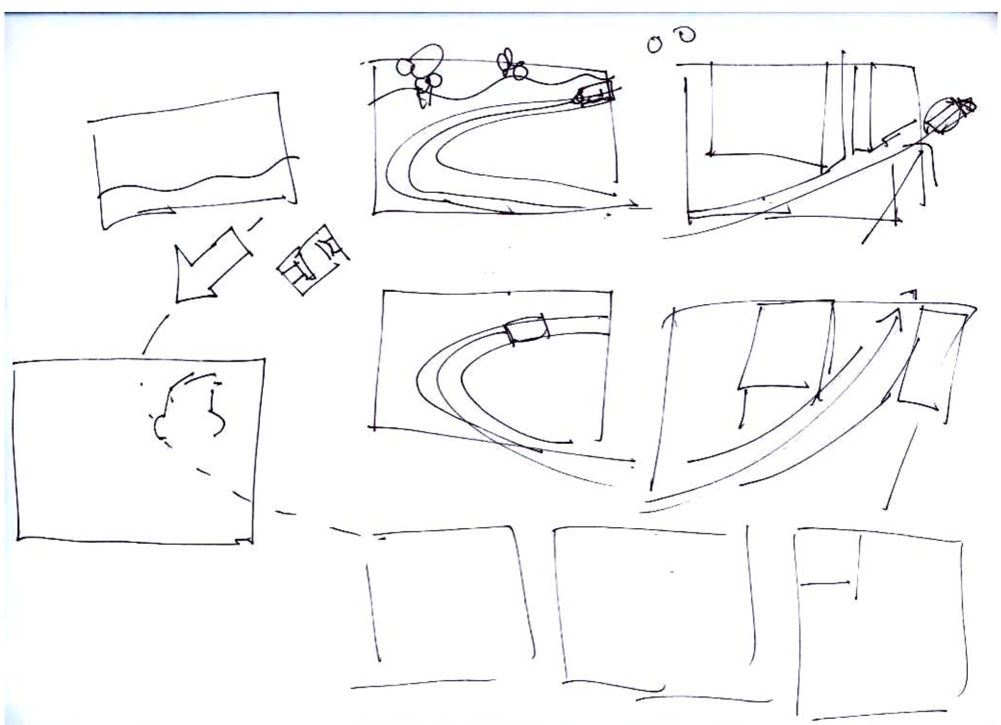
Security System storyboards (bottom frame is source of 9–*Security card slide*)



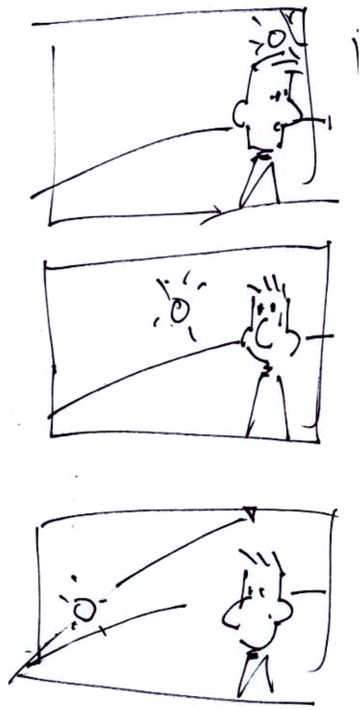
Security System storyboards (top frame is source of 11–*Perpetrator turning*, bottom is source of 10–*Perpetrator entrance*)



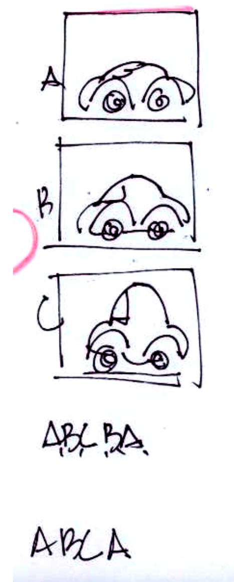
Source of 13–*Car steering*



Source of 14–*Car on country road*

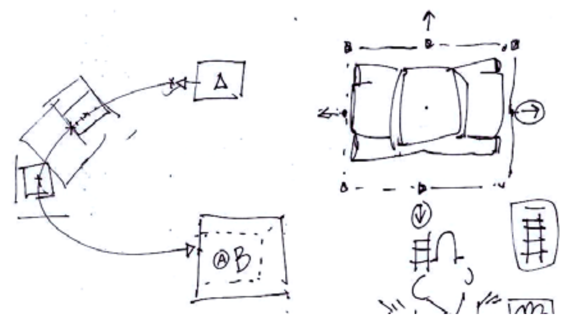


Source of 15–*Man watching sun*

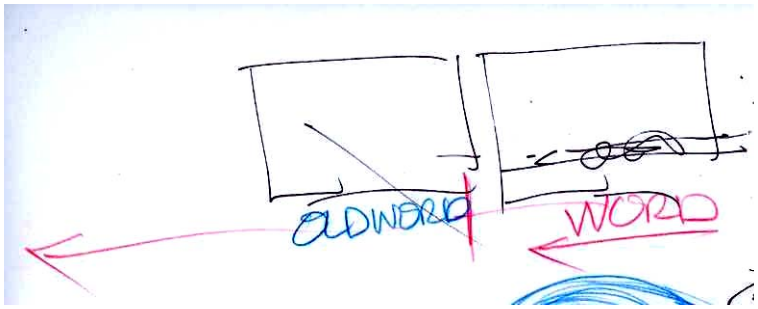


Source of 18–*Car bobbing*

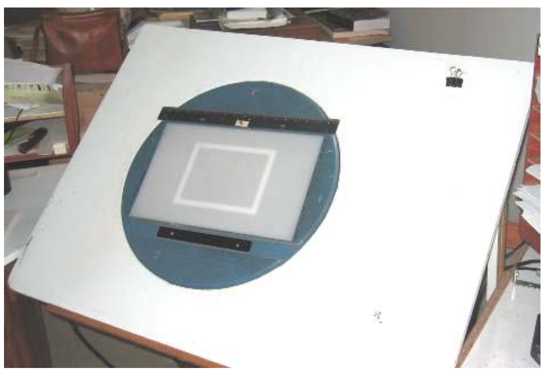
Source of 16–*Thermometer with more*



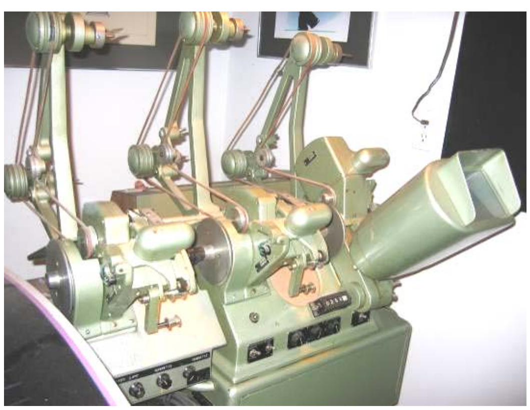
Source of 17–*Translate rotate scale*



Source of 19–*Word collision*

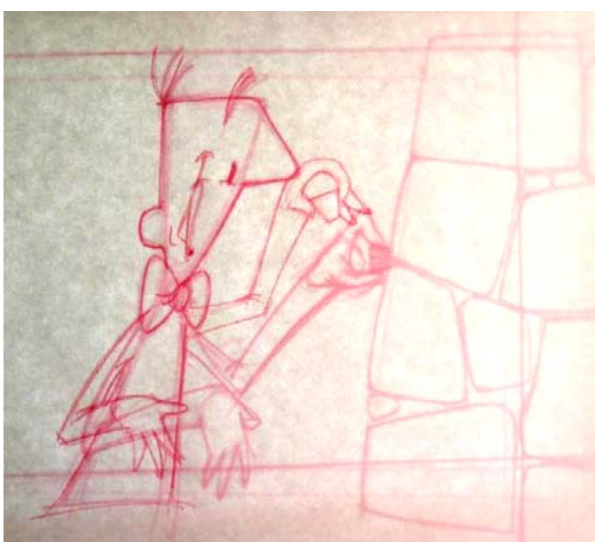

Animation disk

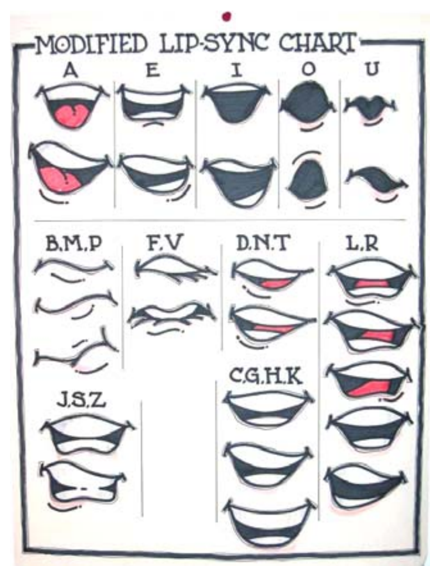

Animation stand


Movieola editing machine

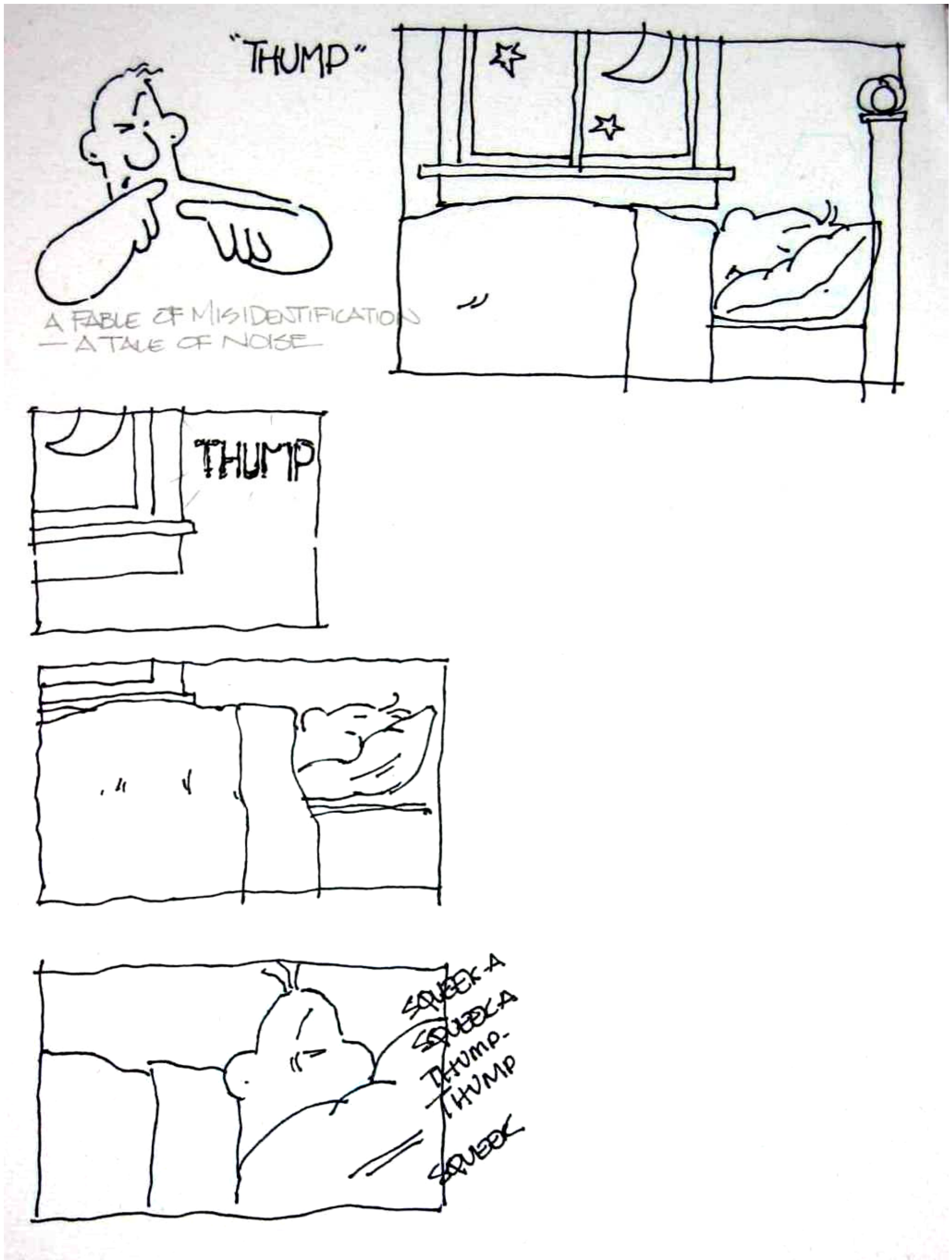

Squawk box and sync block

Background painting with a cel overlay on an animation disk

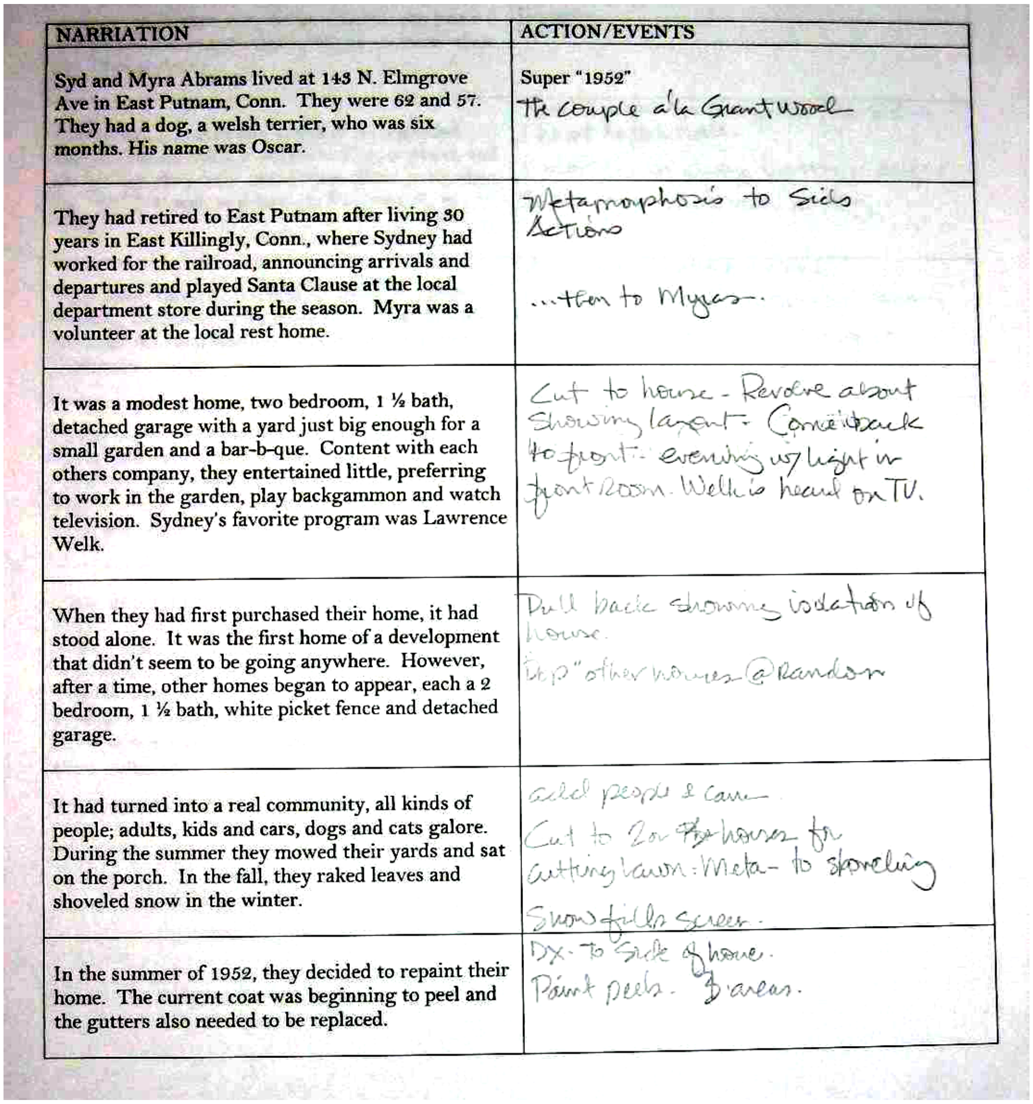

Character test with two frames overlaid



Mouth shapes reference sheet

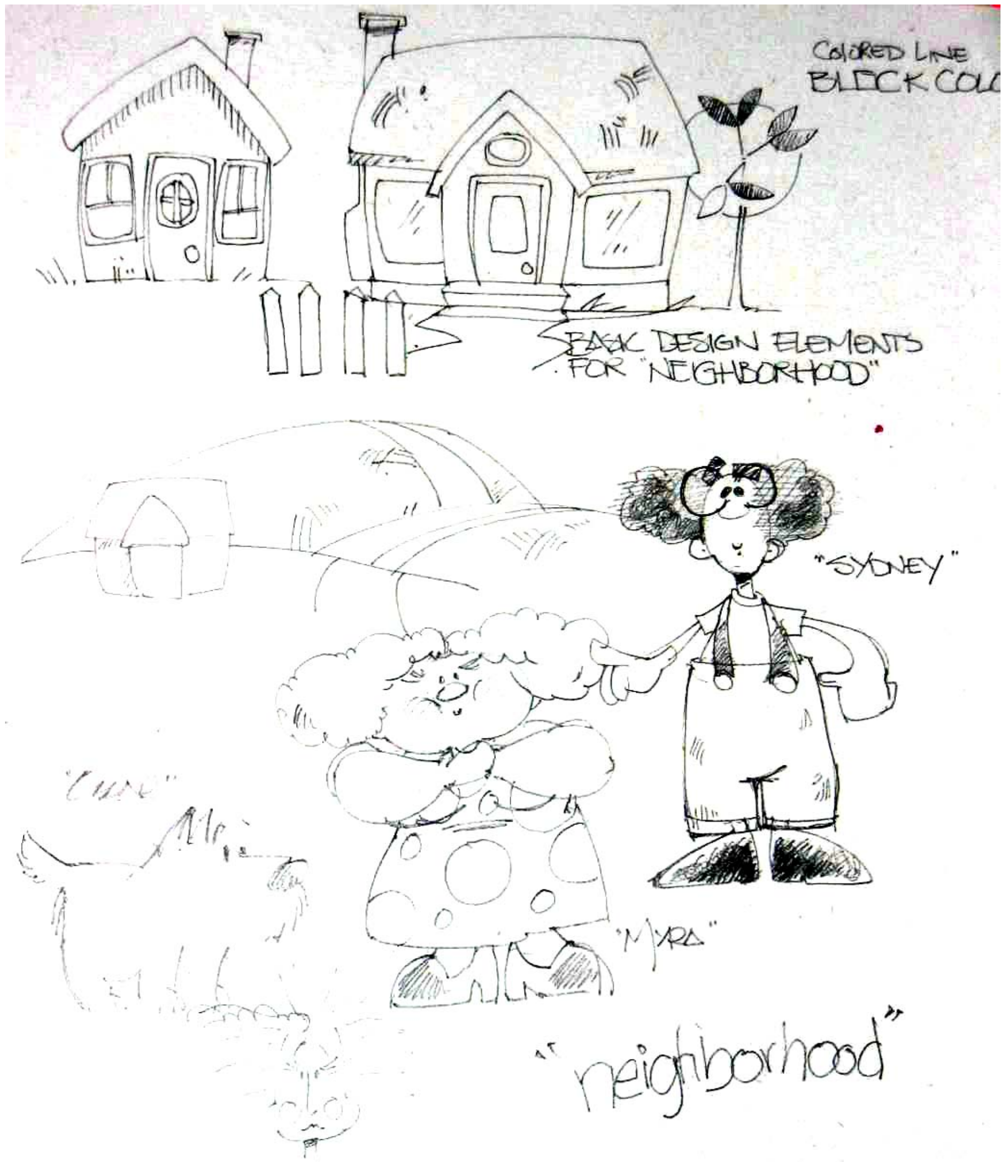


Fable storyboard



NARRATION	ACTION/EVENTS
Syd and Myra Abrams lived at 143 N. Elmgrove Ave in East Putnam, Conn. They were 62 and 57. They had a dog, a welsh terrier, who was six months. His name was Oscar.	Super "1952" The couple à la Grant Wood
They had retired to East Putnam after living 30 years in East Killingly, Conn., where Sydney had worked for the railroad, announcing arrivals and departures and played Santa Clause at the local department store during the season. Myra was a volunteer at the local rest home.	Metamorphosis to Sid's Actions ...then to Myra's.
It was a modest home, two bedroom, 1 ½ bath, detached garage with a yard just big enough for a small garden and a bar-b-que. Content with each others company, they entertained little, preferring to work in the garden, play backgammon and watch television. Sydney's favorite program was Lawrence Welk.	Cut to house - Revolve about showing layout - Come back to front - evening w/ light in front Room. Welk is heard on TV.
When they had first purchased their home, it had stood alone. It was the first home of a development that didn't seem to be going anywhere. However, after a time, other homes began to appear, each a 2 bedroom, 1 ½ bath, white picket fence and detached garage.	Pull back showing isolation of house. Dip "other houses @ Random
It had turned into a real community, all kinds of people; adults, kids and cars, dogs and cats galore. During the summer they mowed their yards and sat on the porch. In the fall, they raked leaves and shoveled snow in the winter.	add people & cars. Cut to 2 or 3 houses for cutting lawn: Meta- to shoveling. Snow fills screen.
In the summer of 1952, they decided to repaint their home. The current coat was beginning to peel and the gutters also needed to be replaced.	DX - To side of house. Paint peels. 3 areas.

Neighborhood script



Neighborhood sketches

A.3 Animator 3

- Interview date: 2/11/2004
- Occupation: animator, animation teacher
- Primary animation media: key framed drawings, cel animation

- Scenarios: (prototyping character tests and timing experiments, learning tool that matches students' intuition)

  20–Simple seagull (artist, prototype)

  21–Bird swimming (artist, prototype)

  22–Bird flying (artist, prototype)

  23–Leaf falling (artist, prototype)

  24–46 From video tapes of children's animations (with one created by the animator, 40–Hello from instructor)

A.3.1 Notes on Initial Contact

1/15/2004 Contacted Via e-mail, saw animation workshop advertisement

1/30/2004 Saw at Northwest Animator's social

2/3/2004 Following notes written based on contact

- 5 Short films Alphids (Puffins, etc) for Seattle aquariums.
- Make "Animatics" w/ storyboard - Video Lunchbox
- Leaf Falling, block of wood, piece of string
- Can you draw and have drawing animated?

Sketch out motions on my bird project. "And then the bird will go like this."

Liked Painter, particularly the onion skin feature

A.3.2 Interview Questions

1) How long have you been working as an animator?

13 years

2) Could you describe the steps in your work process? Give greater detail on the early parts of the process and the parts that involve sketching.

Animator's own process:

1) sketch – collect styles, collect images & video
2) drawings become more final

3) sketching out phases of motion, test out timing
4) storyboards – many… re-done
5) simple animation
6) full-page storyboard

(items 4–6: Images important)

7) animatics w/audio – in iMovie
8) pieces get replaced
    - scan/replace bits of animatic.
    - take to a friend (Animator 1)

Process during one-week course for kids

1) develop character
    - sketching character and background
2) build character out of clay
3) "act out story"
4) storyboard

3) What hardware or software tools do you use to do your work?

Traditional tools for animator's process: pencils, charcoal, oil pastel, smudge sticks, ink, paint, notebook, 4" x 6" index cards, homemade peg bar (for managing overlaid sheets).

Computer tools for animator's process (Mac and PC): scanner, digital camera, video camera, Video Lunchbox, Photoshop, AfterEffects, Final Cut Pro, Pro Tools (sound), Flash (uses occasionally but doesn't like much, has good vector-based drawing tools, not suited to charcoal), Painter, iMovie (for quick and dirty sketch of video editing).

Tools in one-week course for kids: clay, pencil, paper, foam, paper pads for flipbooks, Video Lunchbox

A.4 Animator 4

- Interview date: 2/20/2004

- Occupation: computer science graduate student
- Primary animation media: Slithy (animation description language)
- Scenarios:

  No scenarios (participant does not prototype)

A.4.1 Notes on Initial Contact

Met through computer science connections

A.4.2 Interview Questions

1) How long have you been working as an animator?

Created three conference presentations with Slithy, working on fourth.

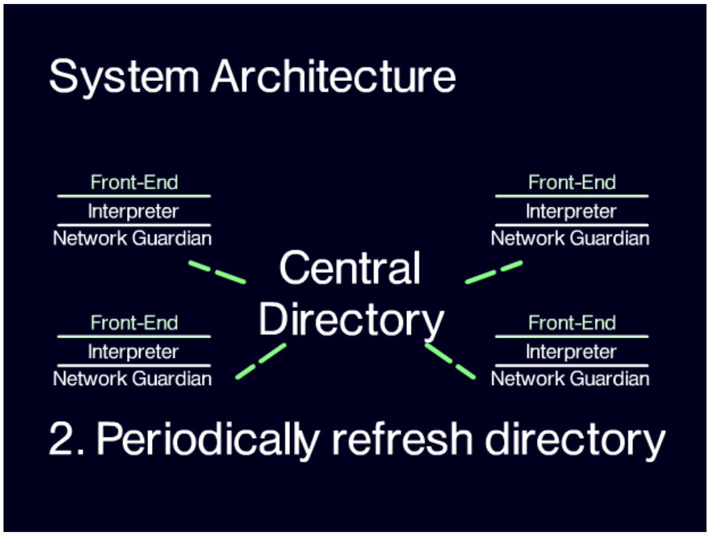
2) Could you describe the steps in your work process? Give greater detail on the early parts of the process and the parts that involve sketching.

- Focused on aspects he wants audience to take away (thinking alone)
  - Think bottom up; do the graph first.
- Sometimes a storyboard
- Code
- Likes to think there is no iteration, but there is some tweaking.
- Throw away animation? No, made it work.
- Don't know how to create a presentation with someone else.

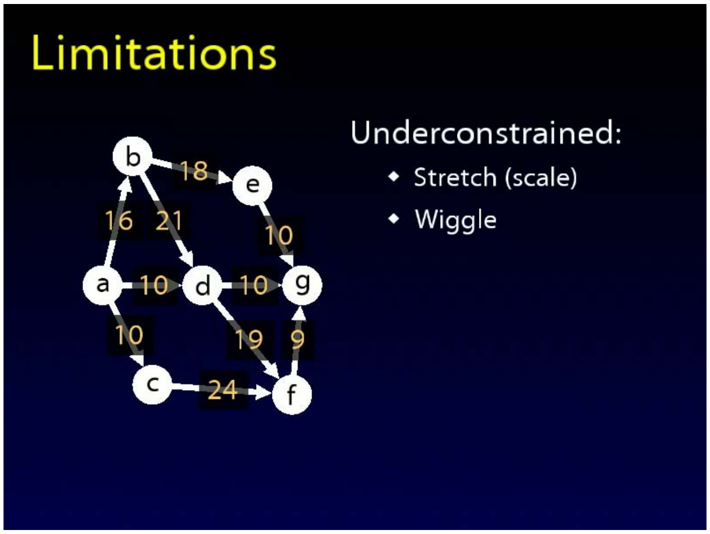
3) What hardware or software tools do you use to do your work?

Slithy (on Mac, sometimes Linux), xfig, jgraph, neato (AT&T graph visualization program)

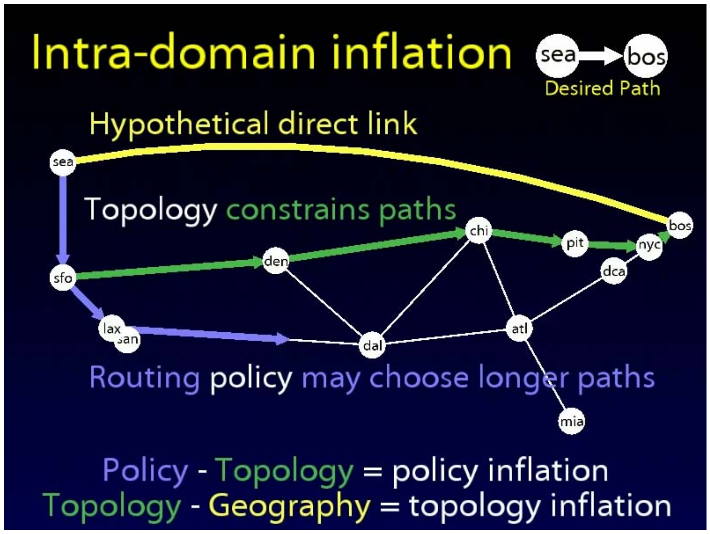
A.4.3 Images and Artifacts



Slithy animation: most common motion is blocks of words that translate and scale, black marks in green lines also move.



Slithy animation: nodes b and f move in and out, arrows stay connected, numbers on moving arrows change.



Slithy animation: arrows trace out paths from sources to targets.

A.5 Animator 5

- Interview date: 2/20/2004
- Occupation: computer science graduate student
- Primary animation media: Slithy (animation description language)
- Scenarios: (prototyping to test length and value of animation)

  47–Dynamic programming (professional, prototype)

  48–Our algorithm (professional, prototype)

A.5.1 Notes on Initial Contact

Met through computer science connections

A.5.2 Interview Questions

1) How long have you been working as an animator?

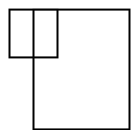
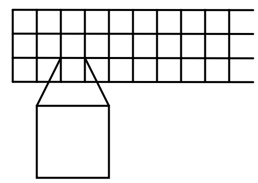
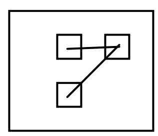
One big conference presentation with Slithy

2) Could you describe the steps in your work process? Give greater detail on the early parts of the process and the parts that involve sketching.

- Sketching storyboards (Tablet PC, PowerPoint)

- write some Slithy to see if visualization works
- Move to full development

Notes on the development of this participant's only presentation:

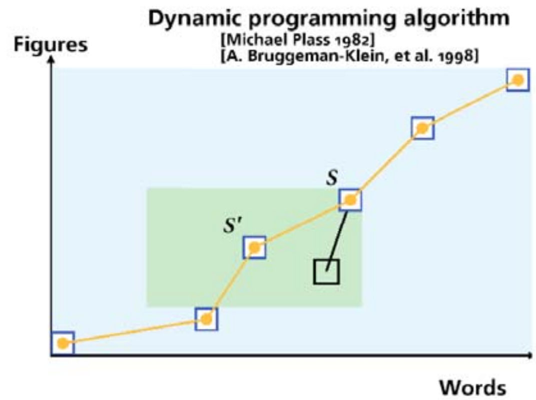
- Made twice as much stuff as had time for in presentation.
- Did various Slithy experiments
    -  Built just to see how to develop Slithy code
    - Try grid with parts exploding out 
    - 
- Quotes
    - "We did a lot of polishing of stuff that never got used."
    - "Just being able to prototype quickly is enough – don't need to produce Slithy code."
    - "If we could have narrowed down specifically what animations we had time to include in the talk, that would have saved a lot of design time."

3) What hardware or software tools do you use to do your work?

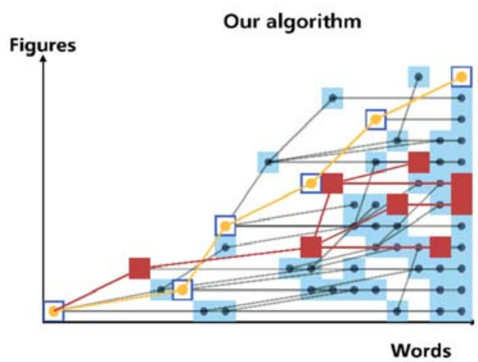
PC's, Slithy

A.5.3 Images and Artifacts

Sketch of an abandoned animation



Source of 47–*Dynamic programming*



Source of 48–*Our algorithm*

A.6 Animator 6

- Interview date: 9/15/2004
- Occupation: producer, animator
- Primary animation media: key framed drawings and images
- Scenarios: (prototype interstitials by sketching them with clients)

  No scenarios (interview took place after I stopped adding to library)

A.6.1 Notes on Initial Contact

1/30/2004 Met at Northwest Animator's Social

A.6.2 Interview Questions

1) How long have you been working as an animator?

4 years with AfterEffects (working at a game company after college).

11 years including college

2) Could you describe the steps in your work process? Give greater detail on the early parts of the process and the parts that involve sketching.

Rock video (0% of income, 20% of time)

- Shot on Sony PD150 Camera
- DV Deck → firewire → Premiere Pro edit
- Import Premiere project into AfterEffects
- Encore DVD

Interstitials (in-between animations) (50% of income, 20% of time)

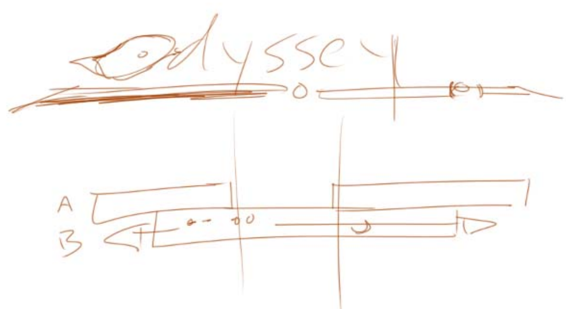
- See client designs, take notes
- Sketching on paper or Sketchbook Pro
- Sometimes start directly in AfterEffects
  - First design idea is often the best.
  - Once you work with clients long enough, you get to know what they're thinking.
  - Wonders if he should take time to consider more designs
- Resource grab (off the internet)
- AfterEffects (last step)
- Last clients needed 3–4 iterations before they were happy in beginning
  - Have to meet the clients face to face
  - Once he got the logos down, he could work in one iteration

Motion capture (50% of income, 60% of time)

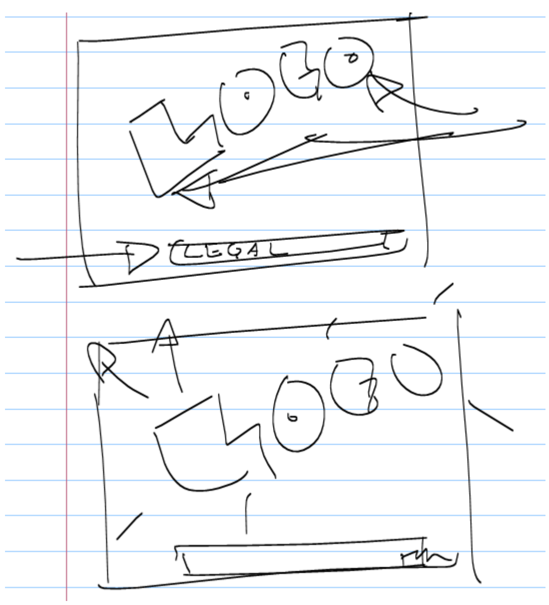
3) What hardware or software tools do you use to do your work?

Acer TravelMate C300, AliasSketch (for sketching), Illustrator (for drawings), *Photoshop (for beveling), *AfterEffects, Director (for UI mock-ups), Premiere (for video editing)

A.6.3 Images and Artifacts



Sketch of an interstitial animation (image courtesy of Troy McFarland)



Sketch of an interstitial animation (image courtesy of Troy McFarland)

A.7 Animator 7

- Interview dates: 7/2/2003, 8/5/2003, and e-mail 9/24/2006
- Occupation: web designer
- Primary animation media: Flash (key framed objects)
- Scenarios:

  No scenarios (prototypes directly in Flash)

A.7.1 Notes on Initial Contact

Met at a social gathering for Flash users

A.7.2 Interview Questions

1) How long have you been working as an animator?

12 years

2) Could you describe the steps in your work process? Give greater detail on the early parts of the process and the parts that involve sketching.

Preproduction

1. contracts
2. client questions
3. discuss document
4. spec document

Production Process

1. visual storyboarding
   - layout storyboards
   - animation storyboards
   - interaction storyboards
2. actionscript planning
   - discuss document
   - PDL or pseudo-code
   - flowcharting

3. functionality tests
4. prototyping

Notes on layout storyboards:

- Use for interface system projects (web or interactive).
- Create in FreeHand, Illustrator, Flash or sketch in a sketchbook.
- Use as your first point where the client signs off on your progress
- When starting a project, the first presentation that I make is usually with printed or displayed concepts that I've either mocked up in a program like FreeHand or Flash, or that I've hand sketched. Layout storyboards are a simple way to present the first set of visual concepts for the client to decide a direction. They are also specific to projects that involve an interface system of some sort (web page or interactive project). Graphic designers almost always use this type of storyboard for developing their printed work. For our purposes, we will use the storyboard to show the client or art director two or three choices of style, color and the arrangement of interactive and display elements.

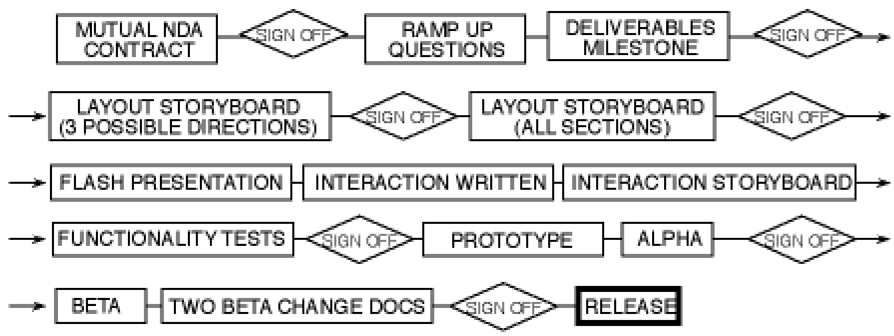
Notes on "PDL or pseudo-code":

- Flowchart storyboarding for complex actionscript
    - prototyping
- While prototyping, you'll also want to keep your interaction storyboard and interaction description up to date with changes that need to be applied to the whole project, especially if you are working with a large group of people. If you need to move really quickly, then at least keep a sketchpad to jot notes about what you are doing so that you can come back through the project after you've finished and create updated versions of the documents. This may seem stupid, but when you have to come back—six days, weeks, or months later—to make changes, I promise that you'll be very happy that you did this.

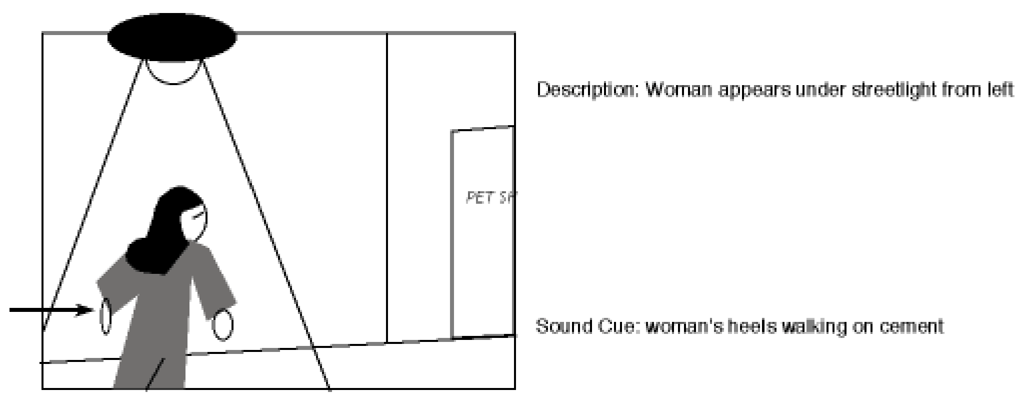
3) What hardware or software tools do you use to do your work?

For storyboards: Visio, Illustrator, Freehand, Photoshop, *Fireworks (tool of choice), Flash, text files

A.7.3 Images and Artifacts



Flowchart of preproduction process



Animation storyboard

A.8 Animator 8

- Interview date: e-mails on 3/25/2004, 3/29/2004, and 4/3/2004
- Occupation: animator
- Primary animation media: cel animation
- Scenarios: (possible medium for finished works)

49–Line music (artist, entertain)

A.8.1 Notes on Initial Contact

Records lost

A.8.2 Interview Questions

1) How long have you been working as an animator?

Three or more years (not asked in original interviews, estimate based on information on participant's web site)

2) Could you describe the steps in your work process? Give greater detail on the early parts of the process and the parts that involve sketching.

- Start with rough drawing
- Scan into computer
- Using smoothing in Illustrator
- Looking for a better process (piece of software) for animating string-like characters

3) What hardware or software tools do you use to do your work?

Hand sketches, Illustrator (Mac), drawing tablet

A.9 Non-animator 1

- Interview date: 8/5/2003
- Occupation: education graduate student
- Domain: biology
- Scenarios :

    50–Meiosis (student, think)

A.9.1 Notes on Initial Contact

Met in an education research group

A.9.2 Interview Questions

1) Description of animations

Wants high school biology students to create animations of meiosis process to help them understand the process. Pointed to numerous examples of meiosis and mitosis animations on the web, including *Mitosis on the Run* by Tom Diab of Saline High School (http://biology.about.com/library/blmitosisanim.htm), which I used as a basis for analysis.

2) Why not use existing tools to create these animations?

This graduate student is aware of no tools that would allow students to create such animations quickly and easily.

3) What tasks do these animations support?

Learning biology

4) Why are animations necessary to accomplish these tasks?

This graduate student uses the following words to justify her use of animation: "I just re-read a fantastic paper that inspired some ideas for me—if you are interested in continuing to think about this problem, see Kindfield (1994) Science Education 78(3): 255-283. In it, she discusses how important it is for students to make their own models in order to really understand a process, a second reason why a modeling task in this [research] project seems like a great idea."

A.10 Non-animator 2

- Interview date: 8/21/2003
- Occupation: education graduate student
- Domain: physics
- Scenarios :

    51–Detection of distant planets (teacher, explain)

A.10.1 Notes on Initial Contact

Met in an education research group

A.10.2 Interview Questions

1) Description of animations

Animation showing how the presence of distant planets is detected and how their mass is measured. As a planet rotates around a star, the star wobbles slightly. The star's wobble causes the wavelength of light it emits to vary (Doppler shift). The animation shows the planet, the star, these waves, and the Doppler shift.

2) Why not use existing tools to create these animations?

This graduate student does not know how to use any animation tools and cannot find the time to learn how to use any tool.

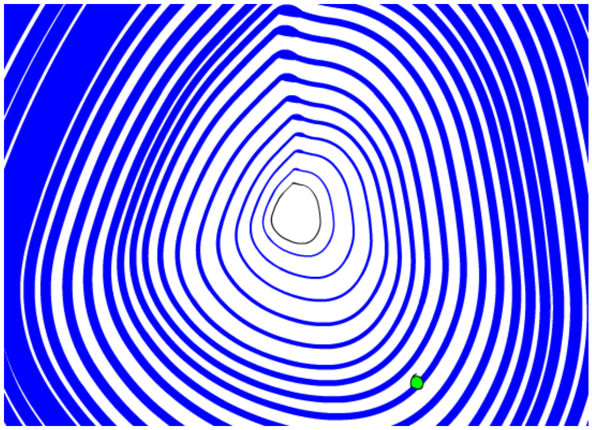
3) What tasks do these animations support?

Teaching physics

4) Why are animations necessary to accomplish these tasks?

This is a dynamic process that involves three distinct but connected features: the planet's revolution around the star, the star's wobble, and the Doppler shift. It is difficult for students to understand these connections without studying an animation that shows all features together.

A.10.3 *Images and Artifacts*



A version of 51–*Detection of distant planets* that I created to verify my understanding of this graduate student's wishes. Her response shows that improvements were needed. "I had imagined one wave that radiates from one point on the star—a wave that compresses and expands as the star rotates. This compression and expansion is what causes the 'red shift' that scientists use as a tell-tale sign of an orbiting planet."

<u>Modeling Activity Goals</u>

Using the *Gravitation* environment to model the effects of mass, velocity, and orbital radius on planetary motion, students will demonstrate an understanding of the way these impact (a) a planet's **habitability**, and (b) whether the planet's presence will cause a **detectable** change in the motion of it's host star.

<u>Modeling Activity Summary</u> (3 parts)

I. Following a short introductory class discussion about the purpose of modeling in this activity, the first activity has students examine models of mass, velocity, and orbital radius for each of planets in our solar system. Students assess for each planet (a) it's habitability, and (b) it's effect on the motion of the sun. In so doing, they confront preconceptions about planetary motion and apply their knowledge of how mass, velocity, and radius (distance between the planet and its sun) interact to determine the speed and trajectory of a planet's orbit.

II. In the second part of this activity, students are given a data table describing the orbital characteristics of several exosolar planets. Students then apply their knowledge by making qualitative (i.e. yes/no) predictions about habitability and detectability for each new planet. After justifying and discussing their predictions, students return to the modeling environment to test their hypotheses.

III. By now students will have noticed that they have not yet encountered any planets that are both habitable and detectible. In part three, they will be asked to (a) explain why none of the planets detected so far are habitable (b) attempt to alter a modeled planet to determine if a planet can be both habitable and detectible, and (c) explain why it is impossible for a planet to be both habitable and detectible.

A high level description of the modeling activity that requires students to understand the concepts in this animation.

Activity (3) Models Part 2: Applying Concepts - 30 minutes
In the second part of the modeling activity, students are given a data table describing the orbital characteristics of several exosolar planets (all real planets). Students then apply their knowledge by making qualitative (i.e. yes/no) predictions about habitability and detectability for each new planet. After justifying and discussing their predictions, students return to the modeling environment to test their hypotheses.

Make Your Predictions.
In the table below, you will find information about 5 planets. Based on what you know about life on Earth and life on Jupiter, as well as the data provided here, predict whether or not each planet is habitable (suitable for life). Next, hypothesize whether you think each planet would have an effect on its sun – enough to be detected by scientists using 'stellar wobble' methods.

Note that each planet's mass is calculated in Earth Masses. The value in each column represents how many times greater the mass of each planet is than the mass of Earth.

Planet	*Mass (Earth Masses)*	*Distance to star (AU*)*	*Habitable?*	*Detectable?*
1	*10.96*	*0.351*		
2	*14.68*	*0.768*		
3	*0.20*	*0.284*		
4	*16.96*	*2.87*		
5	*0.92*	*3.39*		
Earth	*1.00*	*1*		
Jupiter	*318.00*	*5.203*		

** 1 AU = the distance from Earth to the sun*

A more detailed description of the parts of the modeling activity that use animation.

A.11 Non-animator 3

- Interview date: 4/2/2004
- Occupation: mechanical engineering professor

- Domain: engineering
- Scenarios :

  52–Gear interference (teacher, explain)

  53–Lattice slip (teacher, explain)

A.11.1 Notes on Initial Contact

Met through connections to teachers interested in animation

A.11.2 Interview Questions

1) Description of animations

Explain mechanical engineering concepts to students. One shows the slip of a dislocation through a molecular lattice. Another shows how gears are designed to avoid *interference* (i.e., locking up).

2) Why not use existing tools to create these animations?

This professor rarely finds the time to use the complex 3D tools that he knows of for creating animations.

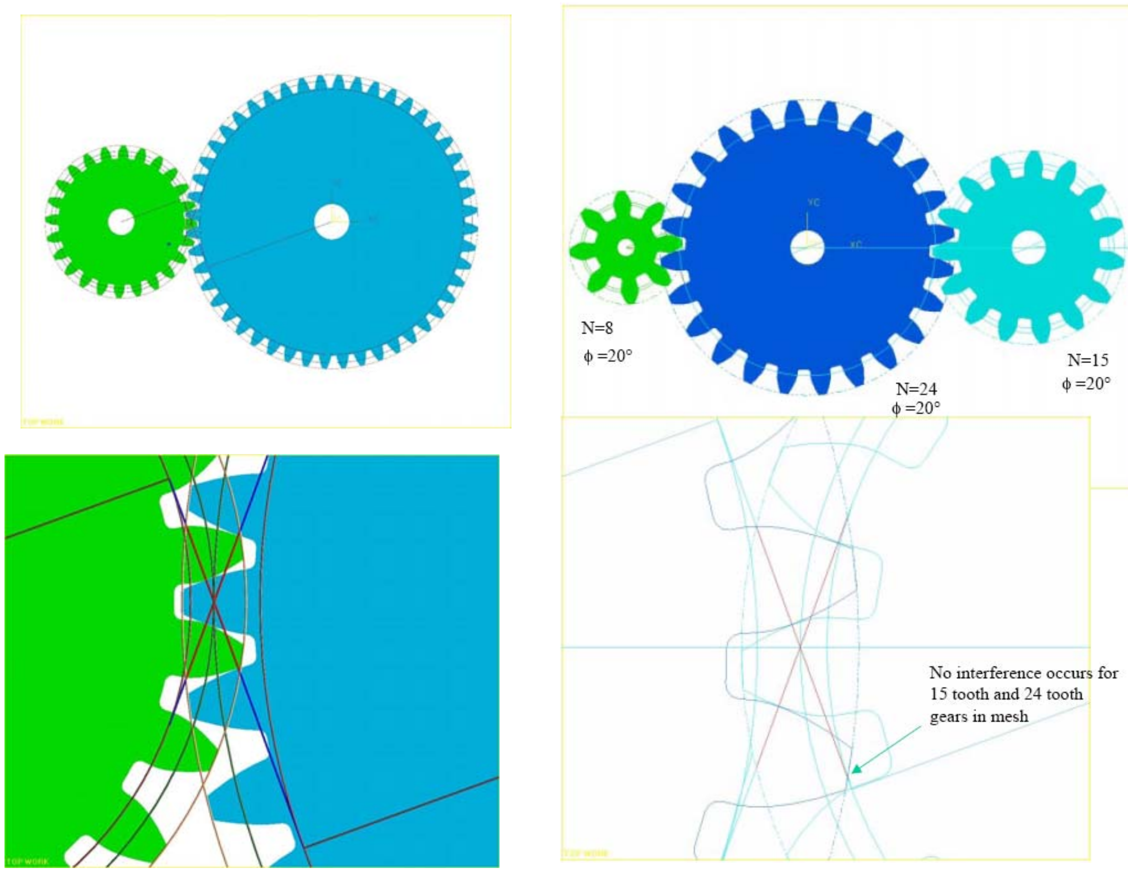
3) What tasks do these animations support?

Teaching mechanical engineering

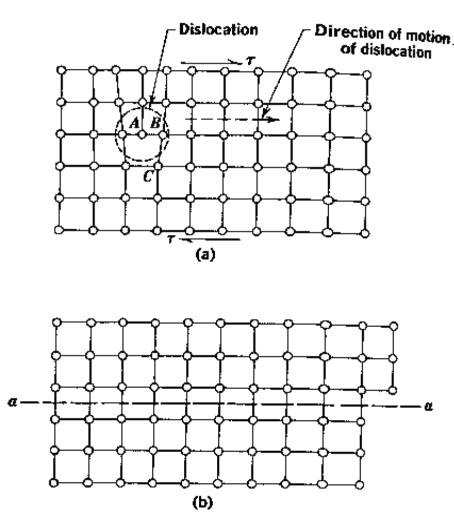
4) Why are animations necessary to accomplish these tasks?

Seeing a smooth animation give students a much better understanding of how these processes work.

A.11.3 Images and Artifacts



Source images for 52–*Gear Interference*.



Source of 53–*Lattice slip*. From Richards CW, *Engineering Materials Science*. San Francisco: Wadsworth, 1961, p 78.

A.12 Non-animator 4

- Interview dates: 3/25/2004, with follow-up e-mails on 3/29/2004 and 3/30/2004
- Occupation: computer science graduate student (and amateur contra dance caller)
- Domain: dance
- Scenarios:

  54–Contra dance (amateur, think)

A.12.1 Notes on Initial Contact

Met through computer science connections

A.12.2 Interview Questions

1) Description of animations

Shows the sequence of moves in a contra dance.

2) Why not use existing tools to create these animations?

The complex movements in these animations take a long time to create in current general-purpose animation tools. (This student was in the process of writing his own software to generate dance visualizations from textual descriptions.)

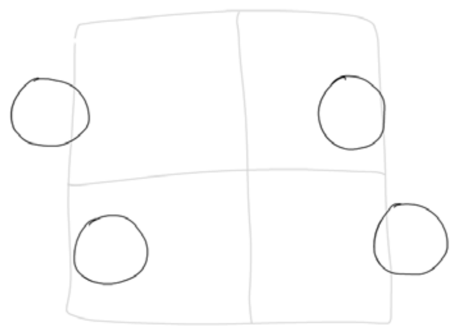
3) What tasks do these animations support?

Visualization of the dance by the *caller* (dance coordinator) before the dance takes place. This give the caller an opportunity to refine the dance before it takes place.

4) Why are animations necessary to accomplish these tasks?

No other external representation captures enough detail to visualize the complex movements in such a dance.

A.12.3 Images and Artifacts



A version of 54–*Contra dance* I created to verify my understanding of this dance caller's wishes. The caller would use these words to direct the sequence of steps shown, "Circle left three quarters. Now partners, balance...and partners swing. Ladies chain...and neighbors, swing."

A.13 Non-animator 5

- Interview date: multiple interviews between 4/9/2004 and 4/12/2004
- Occupation: chemistry professor
- Domain: chemistry
- Scenarios:
  - 55–Chemistry: particle collisions (teacher, explain)
  - 56–Chemistry: rusting reaction (teacher, explain)
  - 57–Chemistry: battery reaction (teacher, explain)

A.13.1 Notes on Initial Contact

Met through connections to teachers interested in animation

A.13.2 Interview Questions

1) Description of animations

Three illustrations of chemical processes to use during introductory chemistry lectures. One shows particles colliding to illustrate particle collision formulae. Another shows how rust forms where water and iron meet. A third shows the movement of particles in a battery reaction. The particle collision example came from the professor's own sketches. The rest came from images in his text book.

2) Why not use existing tools to create these animations?

This professor is very uncomfortable around computer tools and needs something very simple to overcome his discomfort.

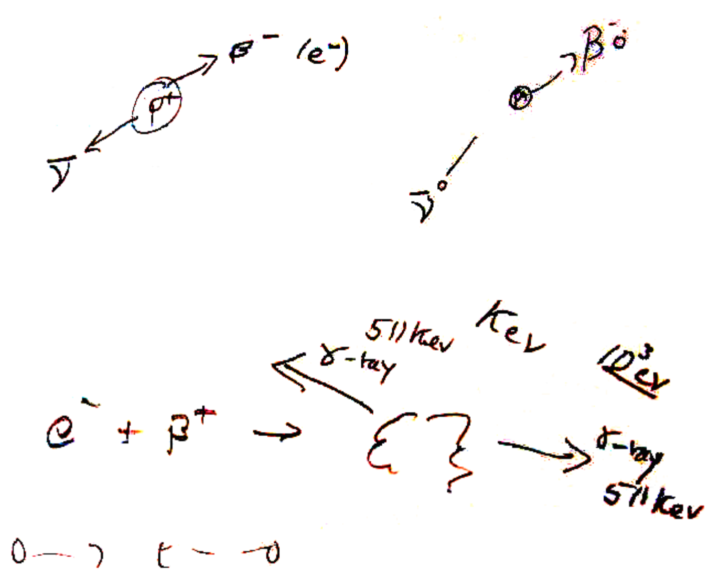
3) What tasks do these animations support?

Teaching chemistry

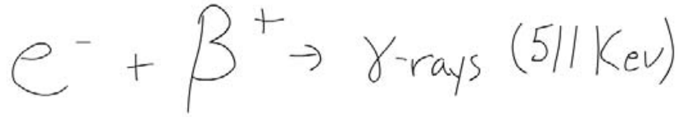
4) Why are animations necessary to accomplish these tasks?

All illustrate the dynamic nature of the reactions described. This dynamic nature is difficult for many students to comprehend, and many develop only a shallow understanding of these concepts (e.g., rote memorization of formulae without grasping the movement of particles involved).

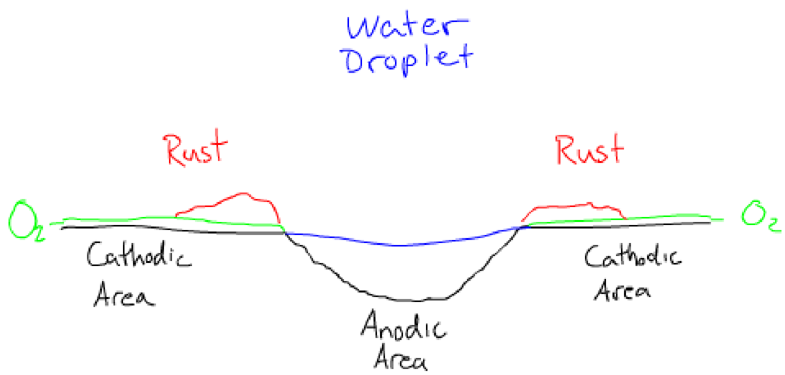
A.13.3 Images and Artifacts



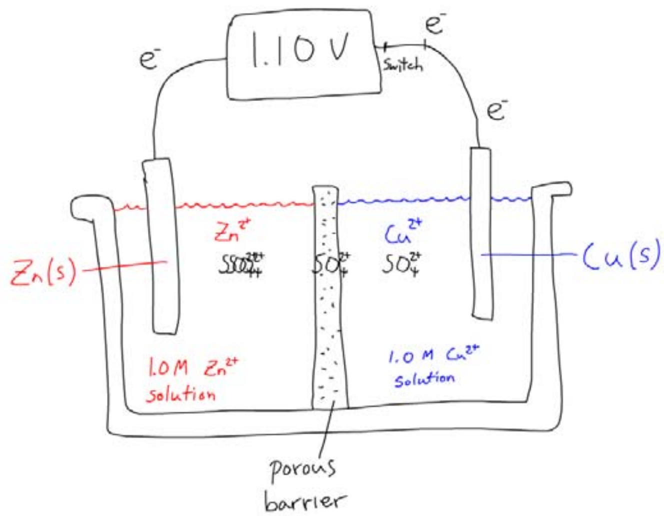
Source sketches for 55–*Chemistry: particle collisions*



A version of 55–*Chemistry: particle collisions* that I created to verify my understanding of this professor's wishes.



A version of 56–*Chemistry: rusting reaction* that I created to verify my understanding of this professor's wishes.



A version of 57–*Chemistry: battery reaction* that I created to verify my understanding of this professor's wishes.

A.14 Non-animator 6

- Interview date: 5/1/2004
- Occupation: control systems researcher
- Domain: engineering
- Scenarios:

 58–Construction equipment tread (professional, explain)

A.14.1 Notes on Initial Contact

Met through social circle

A.14.2 Interview Questions

1) Description of animations

An animation of a machine tread rolling over a bump. The tread is fixed to a milling machine used in road construction that rolls over bumps in asphalt to prepare a level surface for a new layer of asphalt. A line is traced out as the tread rolls over the bump to illustrate the signal that is fed into the control system that must grind the bump into a level surface.

This researcher used the following words to describe what he is looking for in this animation: “What interests me is what wavelengths are damped to what extent. The tread is actually a nonlinear finite impulse response (FIR) filter. For this reason, if I only had your tablet PC with your imagined software here, I would start drawing the tread following different sinusoids, and of course different steps or ramps.”

2) Why not use existing tools to create these animations?

This researcher suspected that Flash would allow him to create this animation, but he could not make the time to learn how to use the software.

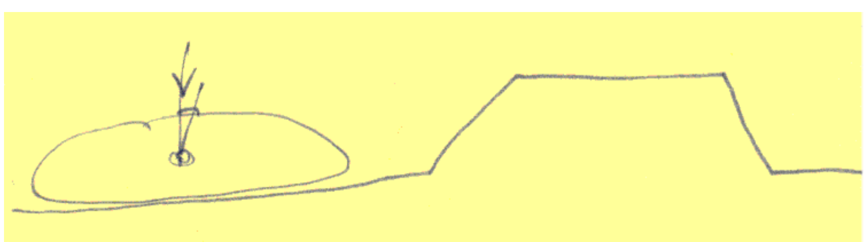
3) What tasks do these animations support?

Explanation of concepts to colleagues during informal meetings

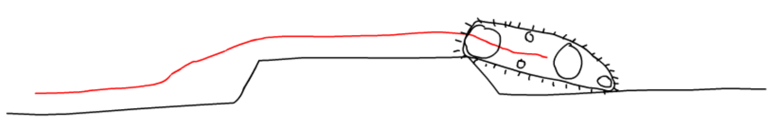
4) Why are animations necessary to accomplish these tasks?

Explaining these concepts has proved difficult in the past, and this researcher suspects that an animation would be a more effective means of communication.

A.14.3 Images and Artifacts



Source of 58–*Construction equipment tread*



A version of 58–*Construction equipment tread* I created to verify my understanding of this researcher’s wishes.

A.15 Non-animator 7

- Interview date: 5/1/2004
- Occupation: college math instructor
- Domain: math
- Scenarios:

    59–Cantor set construction (teacher, explain)

A.15.1 Notes on Initial Contact

Met through social circle

A.15.2 Interview Questions

1) Description of animations

Shows the construction of a *cantor set,* a recursively defined set of numbers in the interval [0, 1].

2) Why not use existing tools to create these animations?

This instructor is aware of no tools that would allow her to create this animation quickly and easily.

3) What tasks do these animations support?

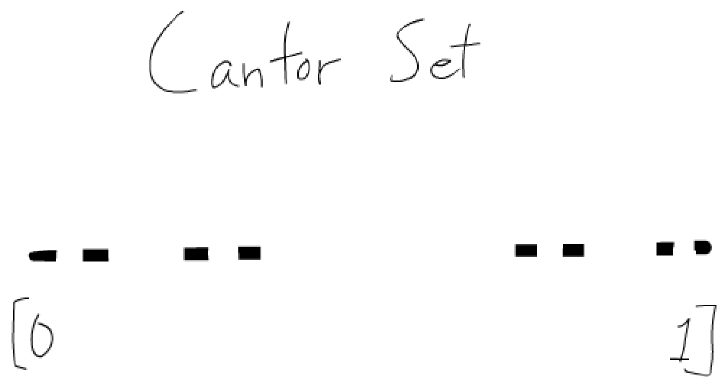
Teaching math

4) Why are animations necessary to accomplish these tasks?

Many students do not grasp the recursive nature of this definition. An illustration that shows steps occurring over time may be clearer.

A.15.3 *Images and Artifacts*

Source of 59–*Cantor set construction*



A version of 59–*Cantor set construction* that I created to verify my understanding of this instructor's wishes.

A.16 Non-animator 8

- Interview date: 5/23/2004
- Occupation: reading tutor
- Domain: reading
- Scenarios:

    60–Bouncing ball (teacher, entertain)

    61–Jumping rope (teacher, entertain)

    62–Face singing (teacher, entertain)

    63–Washing plate (teacher, entertain)

A.16.1 *Notes on Initial Contact*

Met on an airplane

A.16.2 Interview Questions

1) Description of animations

Short, fun animations of various subjects to be shown after students answer questions correctly. Currently, this tutor will show students a set of cards with images and ask a question like, "Pick the one with the 'A' sound." When they pick the right one, this tutor wants to play an animation.

2) Why not use existing tools to create these animations?

This tutor is aware of no animation tool that will allow her to create animations quickly and easily.

3) What tasks do these animations support?

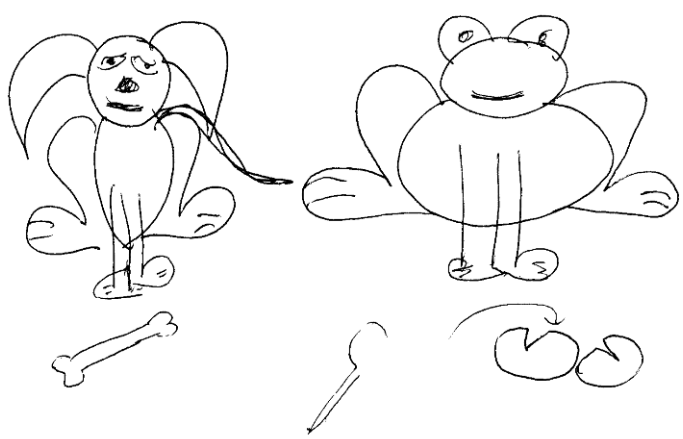
Motivating students to read

4) Why are animations necessary to accomplish these tasks?

These animations would be more compelling than the flash cards that this tutor normally uses. Kids today are bombarded with video games. Their synapses are conditions to respond to moving pictures. Also, animation if fun, and students learn better if they are having fun.

A.16.3 Images and Artifacts

Sketches drawn by this tutor during the interview. Top is source of 60–*Bouncing ball*. Bottom left is source of 61–*Jumping rope*. Bottom right is source of 62–*Face singing*.



More sketches drawn by this tutor.

A.17 Non-animator 9

- Interview date: 11/4/2004

- Occupation: aeronautical engineer
- Domain: engineering
- Scenarios:

  64–Robot arm (professional, explain)

A.17.1 Notes on Initial Contact

Met at a poster session while demonstrating K-Sketch

A.17.2 Interview Questions

1) Description of animations

This animation illustrates the behavior of a measurement system inside a wind tunnel. A robot arm holding a sensor array moves in a sweeping pattern around the wing to measure the flow of air at each point.

2) Why not use existing tools to create these animations?

This engineer is aware of no system that would allow him to create this animation quickly and easily.

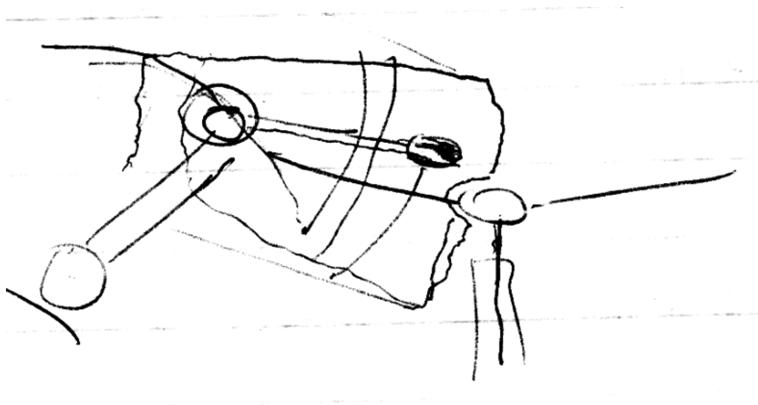
3) What tasks do these animations support?

Explaining the behavior of the sensor array to customers during discussions in the laboratory

4) Why are animations necessary to accomplish these tasks?

Explaining these concepts has proved difficult in the past, and this engineer suspects that an animation would be a more effective means of communication.

A.17.3 Images and Artifacts



Source of 64–*Robot arm*

A.18 Non-animator 10

- Interview date: 11/4/2004
- Occupation: engineering manager
- Domain: engineering
- Scenarios:

 65–Casing slide (professional, explain)

A.18.1 Notes on Initial Contact

Met at a poster session while demonstrating K-Sketch

A.18.2 Interview Questions

1) Description of animations

A box-shaped object slides into a casing.

2) Why not use existing tools to create these animations?

This manager is aware of no system that would allow him to create this animation quickly and easily.

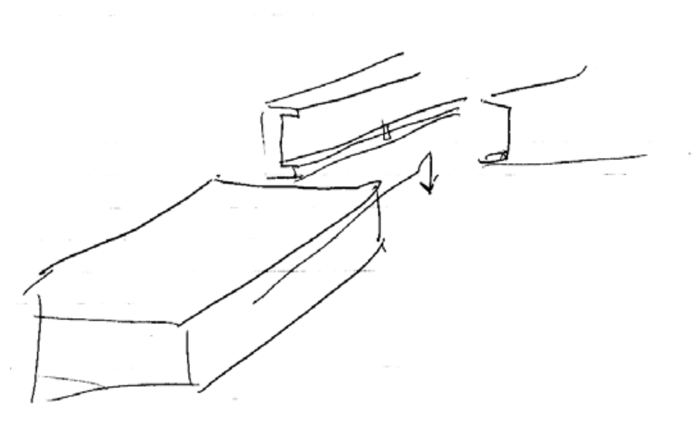
3) What tasks do these animations support?

Explaining the desired sliding motion to an engineer who will design the casing

4) Why are animations necessary to accomplish these tasks?

This manager has had difficulty communicating his desires effectively to those working for him. He wishes to communicate both the shape of the casing and the action that should be required to insert an object into the casing. He suspects that an animation would be a more effective means of communication.

A.18.3 Images and Artifacts



Source of 65–Casing slide

A.19 Non-animator 11

- Interview date: 3/5/2006
- Occupation: geochemical researcher
- Domain: geochemistry
- Scenarios:

  No scenarios (interview took place after I stopped adding to library)

A.19.1 Notes on Initial Contact

Met through social circle

A.19.2 Interview Questions

1) Description of animations

A sinusoidal waveform stretches, tilts, and shifts position relative to a set of axes. The movement pattern of this waveform constitutes *etalon noise*. This researcher is engaged in designing systems that filter out such noise.

2) Why not use existing tools to create these animations?

This researcher has used PowerPoint, but does not believe he can use it to create this animation "on the spot" during meetings with programmers. As this researcher explains, "Often I do use cartoons to relay the illusion of motion, however, this requires both imagination on the part of the observer, and that their imagination is similar to mine."

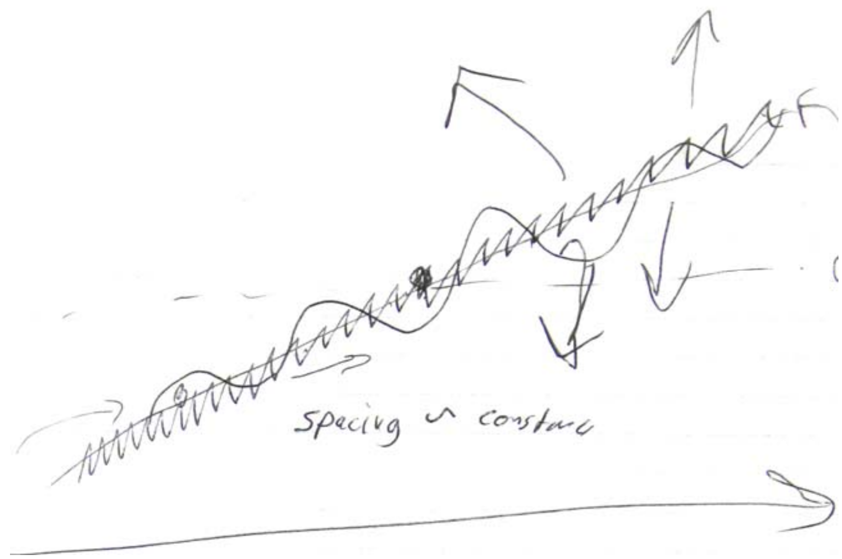
3) What tasks do these animations support?

Teaching concepts to clients, vendors, and patent lawyers during electronic conferences

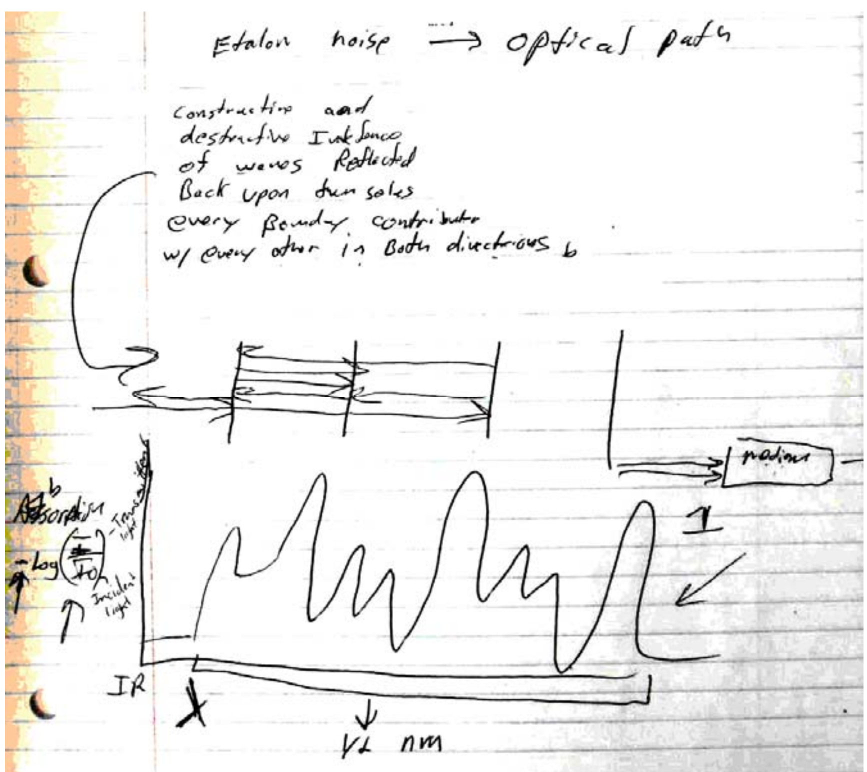
4) Why are animations necessary to accomplish these tasks?

Explaining these concepts has proved difficult in the past, and this researcher suspects that an animation would be a more effective means of communication.

A.19.3 Images and Artifacts



Sketch of an animation showing a waveform experiencing etalon noise.



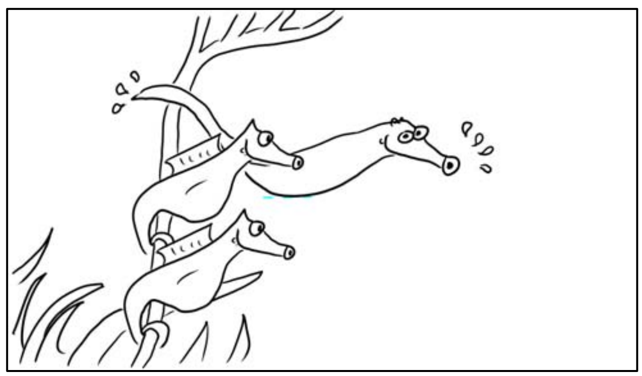
More sketches explaining etalon noise

APPENDIX

B Scenario Library

This appendix contains a partial list of the scenarios I compiled for this research. A more complete list that includes the actual animations can be found online at http://www.k-sketch.org/dissertation/index.html#apB. Each scenario listed here includes a short description and briefly explains how each can be created in K-Sketch.

B.1 Simple Potbelly Fall



User category: artist

Goal category: prototype

Source: Animator 1

Length: 3.1 sec.

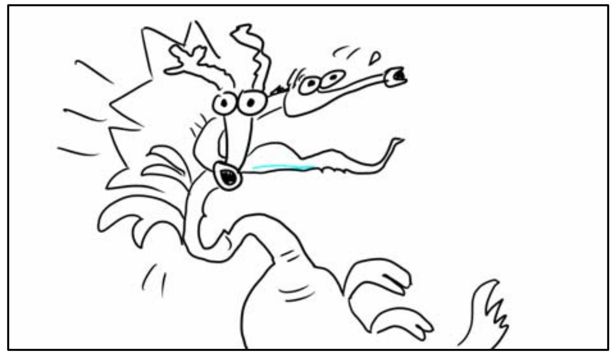
Description: This is a frame from an animated storyboard synchronized with a sound track. A still image of a seahorse sitting with others moves slightly to the right to give the illusion that he is falling off.

Animating in K-Sketch

Time: 3 min. 22 sec.

Notes: Most of the time spent animating this scenario was spent drawing. The motion itself took only a few seconds. Synchronizing with a sound track would require *play sound.*

B.2 Simple Potbelly Crash



User category: artist

Goal category: prototype

Source: Animator 1

Length: 3 sec.

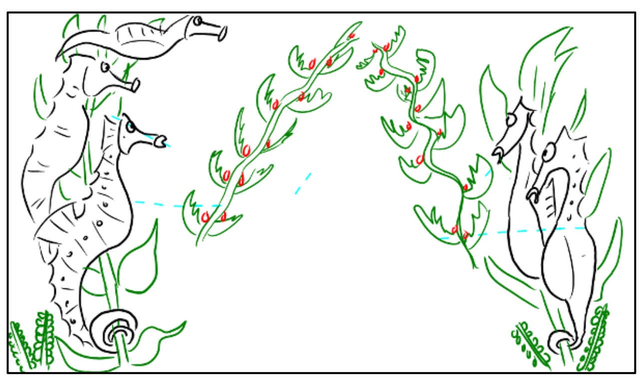
Description: This is a frame from an animated storyboard synchronized with a sound track. A still image of one seahorse colliding with a second seahorse moves slightly to the right and back to create the illusion of movement. The scene then switches to show the second seahorse smiling at the first.

Animating in K-Sketch

Time: 5 min. 54 sec.

Notes: Most of the time was spent drawing the initial scene and the following scene. The motion of the initial scene took seconds. Synchronizing with a sound track would require *play sound.*

B.3 King Leafy Entrance



User category: artist

Goal category: prototype

Source: Animator 1

Length: 12.8 sec.

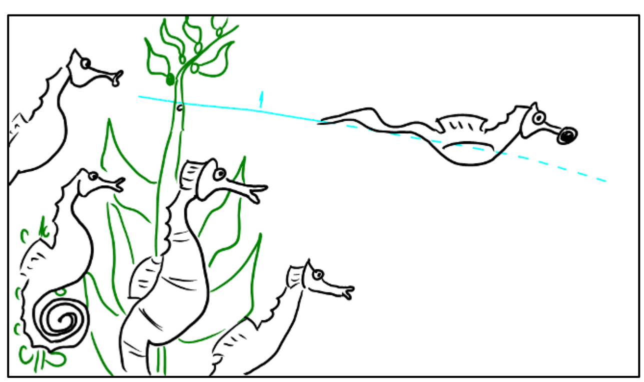
Description: This is a frame from a fairly polished animated storyboard synchronized with a sound track. A crowd of seahorses waits for the entrance of their king. As the camera zooms on the point of entrance, he crowd moves out of view. A trumpet fish comes to announce the king's entrance, its mouth stretching and turning with the trumpet sound. The trumpet fish exits, and the king enters with a graceful swooping and zooming motion.

Animating in K-Sketch

Time: 29 min. 15 sec.

Notes: The complex motions in the scene were fairly easy to produce with the fix last operation dialog box. The simultaneous translate and scale of several groups went very quickly. The somewhat complex motion of the trumpet fish's mouth was very easy to create. Also, the ability to move the handle came in handy when scaling from a particular point or when manipulating characters off screen. Unfortunately, the presence of detailed drawings still made this a difficult animation to produce. This is not only because drawing took time, but also because the current version of K-Sketch has performance problems when complex drawings are present. The remaining points are minor. Separating overlapping characters was somewhat difficult, a sign of the need for improvements in K-Sketch's selection and grouping. Since this scenario required more polished sketches, it would have benefitted from the *occlude* operation and better drawing tools (or *import*. The presence of *move forward/back* also would have made this scenario somewhat easier. Also, synchronizing with a sound track would require *play sound.*

B.4 Complex Potbelly Fall



User category: artist

Goal category: prototype

Source: Animator 1

Length: 4.2 sec.

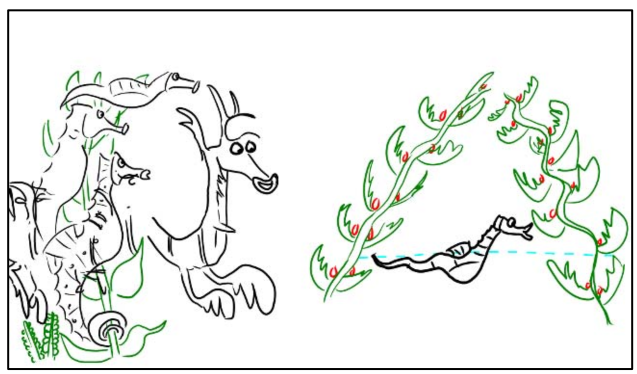
Description: This is a frame from a fairly polished animated storyboard synchronized with a sound track. A seahorse sitting with others on a branch shakes twice and falls off screen.

Animating in K-Sketch

Time: 10 min. 1 sec.

Notes: Most of the time spent animating this scenario was spent drawing. The shaking motion which would have taken four key frames was done in one swift motion using K-Sketch. The somewhat polished nature of the drawings in this animation would require better drawing tools, the *occlude* operation, and better performance for complex drawings. Also, synchronizing with a sound track would require *play sound.*

B.5 Potbelly Heads Out



User category: artist

Goal category: prototype

Source: Animator 1

Length: 10.3 sec.

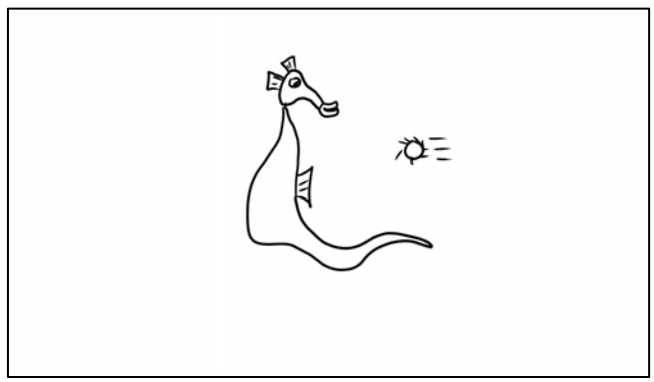
Description: This is a frame from an animated storyboard synchronized with a sound track. Many seahorses look at the seahorse on the right while the king speaks to him. The seahorse on the right then turns and move off screen.

Animating in K-Sketch

Time: 19 min. 49 sec.

Notes: The production of this animation was made simpler by the fact that drawings could be reused from scenario 3. The motion was also trivially easy to produce. Once again, however, the complex drawings made K-Sketch perform poorly, and this animation took much longer to produce than it should have. Like scenario 3, this animation would have benefitted from *occlude*, *import*, and *move forward/back*. Also, synchronizing with a sound track would require *play sound*.

B.6 Potbelly Collision



User category: artist
Goal category: prototype
Source: Animator 1
Length: 3 sec.
Description: This is an animated character sketch of a rock colliding with a seahorse.

Animating in K-Sketch

Time: 8 min. 4 sec.

Notes: This animation was done by drawing and erasing 7 frames. Most of the time was spent trying to get the drawings to look right. The ghosts that appeared after erasing each frame mimicked a light box, making it easy to line up successive frames. Also, the *nudge forward* button made it easy to keep a consistent time between frames. The ability to see multiple frames at once (as in a traditional onion skin) might have made it easier to get a sense of the full motion. The *deform* operation might have made this scenario easier.

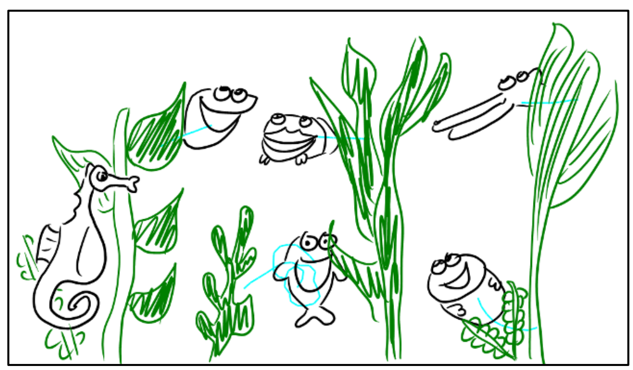
B.7 Hitchhiker

User category: artist
Goal category: prototype
Source: Animator 1
Length: 6.2 sec.
Description: This is an animated character sketch of a hitchhiker waving his arm to thumb a ride.

Animating in K-Sketch

Time: 2 min. 46 sec.
Notes: This was a very simple animation with only three frames. Most of the time was spent drawing the character. The *nudge forward* button made it easy to keep a consistent time between frames.

B.8 Sea Animals



User category: artist
Goal category: prototype
Source: Animator 1
Length: 10.1 sec.
Description: This is a fairly detailed animated character sketch and movement test. Five sea animals emerge from hiding. One of them hovers in an irregular path as others emerge.

Animating in K-Sketch

Time: 10 min. 52 sec.
Notes: The drawings in this animation were somewhat complex and took

nearly two thirds of the animation time. The motions were fairly simple, but the timing of them took a few attempts. The movable timeline tic marks were helpful in this regard. As in similar scenarios from this artist, this scenario would have benefitted from *import, occlude*, *move forward/back,* and *define background.*

B.9 Security Card Slide

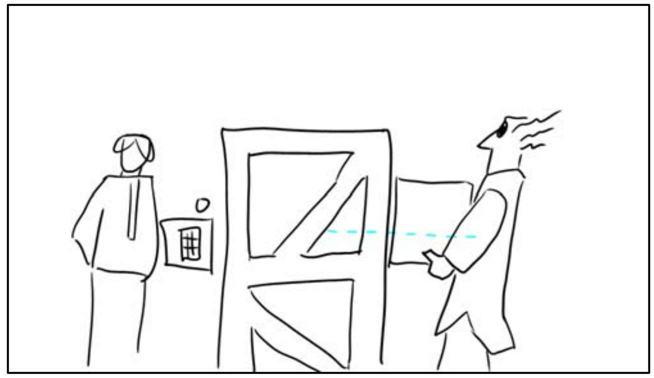
User category: artist
Goal category: prototype
Source: Animator 2
Length: 1.5 sec.
Description: This is an animated storyboard frame showing a hand sliding a card though a card reader.

Animating in K-Sketch

Time: 4 min. 46 sec.

Notes: This animation was easy in most respects. The arm was broken into two parts: the hand moved with *translate* and the arm moved with *rotate.* I slowed down the animation to 25% to coordinate these moving parts. Though it was not strictly necessary, I simulated the arm occluding the line of the man's shirt by breaking the line into small parts and erasing them as the arm moved. An eraser that erased parts of lines instead of whole lines would have made this easier.

B.10 Perpetrator Entrance



User category: artist

Goal category: prototype

Source: Animator 2

Length: 3 sec.

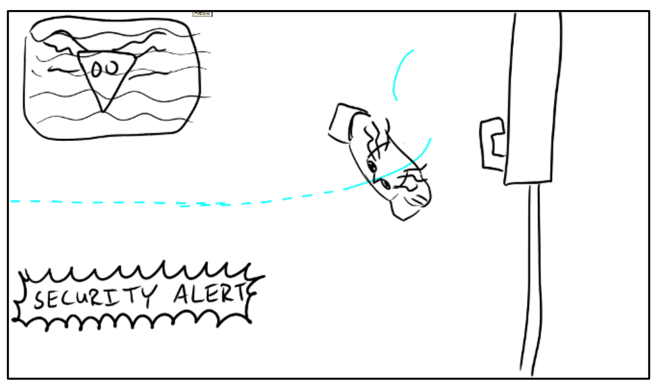
Description: This is an animated storyboard frame showing a man with a box walking up to another man.

Animating in K-Sketch

Time: 2 min. 4 sec.

Notes: Drawing the scene was the hardest part of this animation. The motion was nothing more than a single translate. This animation would have looked better with *occlude*, and it would have been slightly easier with *define background*.

B.11 Perpetrator Turning



User category: artist

Goal category: prototype

Source: Animator 2

Length: 3.8 sec.

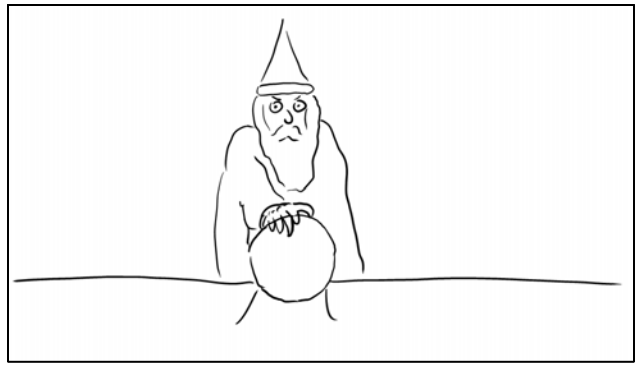
Description: This is an animated storyboard frame showing a man turning away from a wall and walking away.

Animating in K-Sketch

Time: Drawing the scene was the hardest part of doing this animation. The motion was created using two *orient to path* motions followed by a *translate*. This animation would have looked better with *occlude*.

Notes:

B.12 Sorcerer Crystal Ball



User category: artist

Goal category: prototype

Source: Animator 2

Length: 0.8 sec.

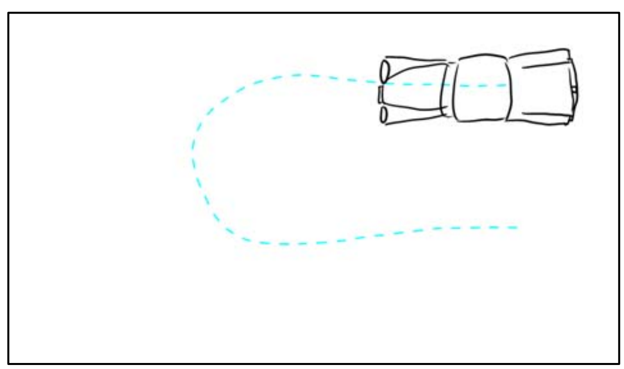
Description: This is an animated character sketch showing a wizard rubbing a crystal ball with his hand.

Animating in K-Sketch

Time: 5 min. 28 sec.

Notes: I spent too long trying to make the crystal ball and the hands match the artist's vision. Otherwise, this animation was a straightforward, 4-frame character sketch. It would have looked better with *occlude*.

B.13 Car Steering



User category: artist

Goal category: prototype

Source: Animator 2

Length: 5.2 sec.

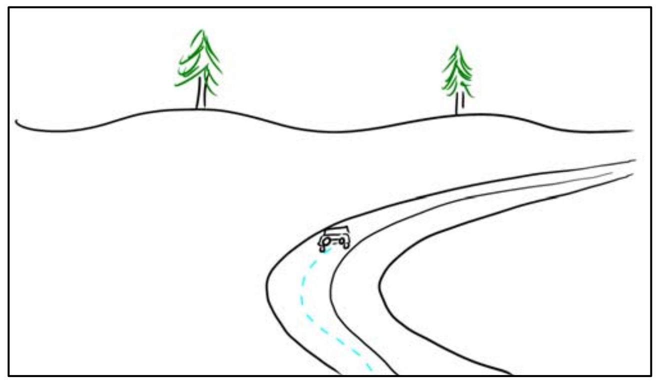
Description: This is an animated character sketch of a car making a U-turn.

Animating in K-Sketch

Time: 1 min. 59 sec.

Notes: This was a simple sketch to draw and animate using *orient to path.* I did the motion three times with three different pivot points until I got the motion that this artist was after.

B.14 Car on Country Road



User category: artist

Goal category: prototype

Source: Animator 2

Length: 9.1 sec.

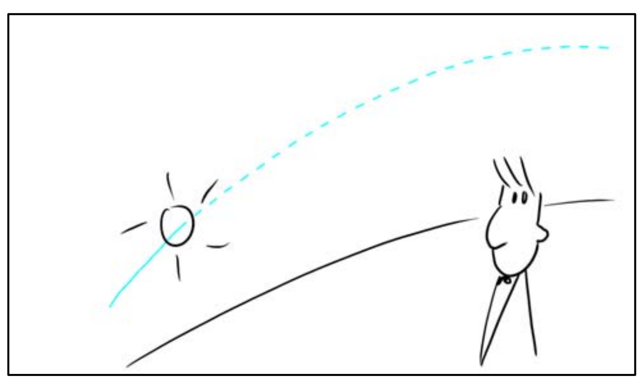
Description: This is a pair of animated storyboard frames showing a car driving down a country road and then through a city.

Animating in K-Sketch

Time: 3 min. 41 sec.

Notes: This was a fairly simple animation as well. The first motion of the car involved a *translate* and a *scale*, which required the *fix* dialog box. The others were simply *translates.*

B.15 Man Watching Sun



User category: artist

Goal category: prototype

Source: Animator 2

Length: 4 sec.

Description: This is a very simple character sketch of the sun moving in an arc and a man turning as the sun moves. The sun moves smoothly, the man moves in a series of three frames.

Animating in K-Sketch

Time: 2 min. 32 sec.

Notes: This was a very easy animation to produce.

B.16 Thermometer With More

User category: artist

Goal category: prototype

Source: Animator 2

Length: 1.8 sec.

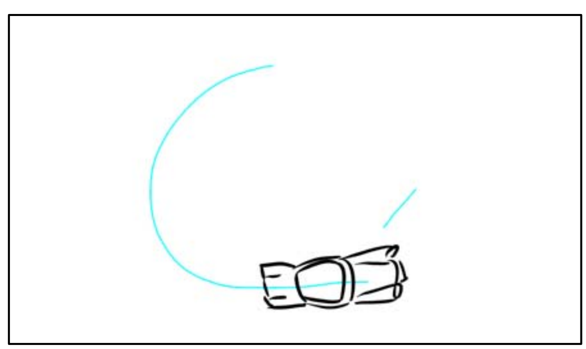
Description: In this character sketch, the temperature indicator in a thermometer rises, and the word "more" rises with it.

Animating in K-Sketch

Time: 1 min. 49 sec.

Notes: This was a fairly easy animation to produce using *trace* and a non-uniform *scale*. I did the scale motion twice, because the first time I forgot to move the center point from the center of the word to the bottom, which caused the word to stretch both up and down. I demonstrated the motions in about six seconds and then sped up the animation to the speed I desired.

B.17 Translate Rotate Scale



User category: artist

Goal category: prototype

Source: Animator 2

Length: 3 sec.

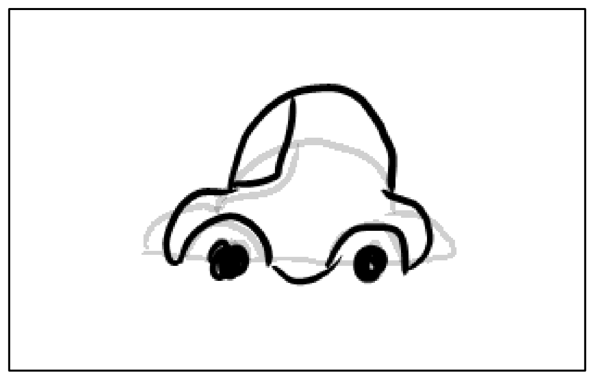
Description: This character sketch of a car making a U-turn is very similar to scenario 13–*Car steering*, except that the car grows as it moves.

Animating in K-Sketch

Time: 1 min. 5 sec.

Notes: This scenario required a demonstrated *orient to path* motion, followed by a *scale,* after which the *fix* dialog was used to make both motions occur at the same time. I made fewer mistakes in drawing an animating, which is why this scenario was completed faster than scenario 13.

B.18 Car Bobbing



User category: artist

Goal category: prototype

Source: Animator 2

Length: 5.6 sec

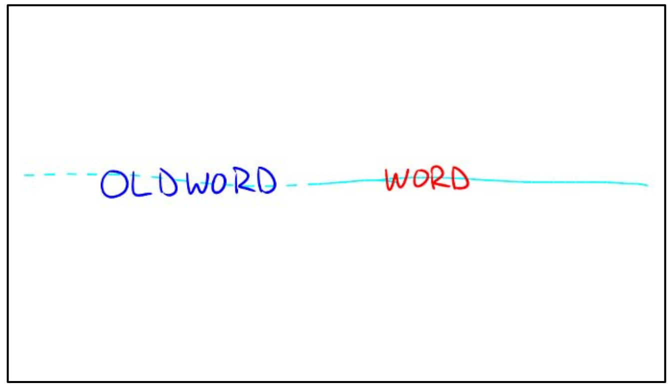
Description: This character sketch shows a car bobbing up and down in a cel cycle (a cycle of repeating images) as it moves.

Animating in K-Sketch

Time: 3 min. 50 sec.

Notes: I simulated a cel cycle by drawing three versions of the car in succession, copying the second version to the end, and then setting *repeat playback.* This should have been trivially easy, but after I had drawn the first half of the cycle, a bug surfaced that made it impossible to select and object. I shut down K-Sketch and restarted. The time shown here includes the time needed to restart.

B.19 Word Collision



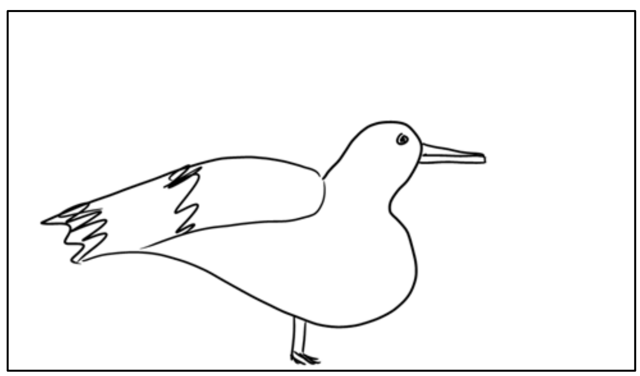
User category: artist
Goal category: prototype
Source: Animator 2
Length: 3 sec.
Description: A word flies in from the right and bumps an old word off to the left.

Animating in K-Sketch

Time: 1 min. 4 sec.

Notes: This was also a very easy animation to produce. I moved the handles to locations that allowed me to pull the new word from off screen and push the old world off screen. My first attempt to coordinate the collisions did not succeed, so I selected the motion path and moved it into proper position.

B.20 Simple Seagull



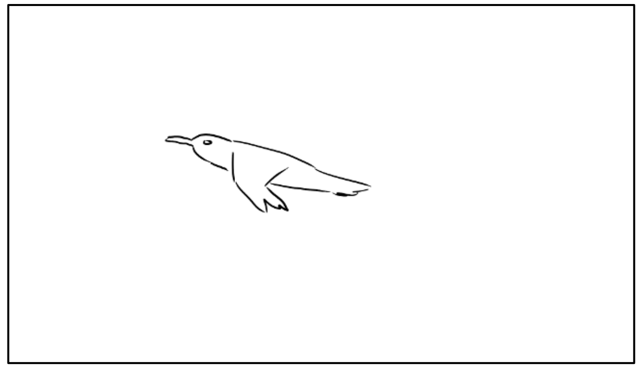
User category: artist
Goal category: prototype
Source: Animator 3
Length: 5.7 sec.
Description: This is a character sketch or a scene from an animated storyboard. A seagull stands, pecks at the ground, looks at the viewer, and flies off.

Animating in K-Sketch

Time: 4 min. 19 sec.

Notes: This was a fairly simple animation done with four drawings, the last of which *translates* away. Most of the time was taken up by drawing.

B.21 Bird Swimming



User category: artist

Goal category: prototype

Source: Animator 3

Length: 3.6 sec.

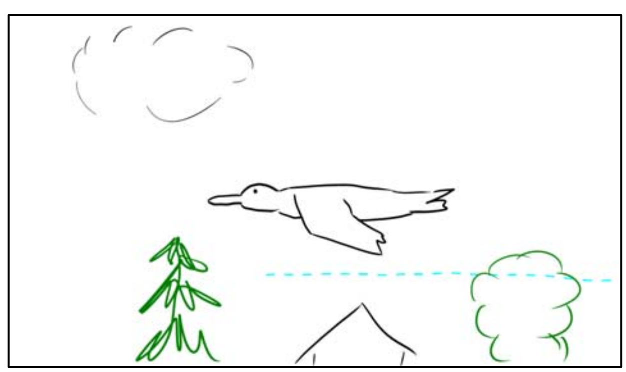
Description: This animated storyboard frame shows a bird swimming through water. The bird stays in place for a moment and then moves off screen. A cel cycle is used for the flapping wings.

Animating in K-Sketch

Time: 18 min. 30 sec.

Notes: Because K-Sketch does not have the *define cels* animation operation, this animation took a long time to produce. I drew three images in succession by drawing and erasing to define the cycle. After that, I would copy the needed drawing in a previous frame and paste it in the current frame to continue the cycle.

B.22 Bird Flying



User category: artist

Goal category: prototype

Source: Animator 3

Length: 4.2 sec.

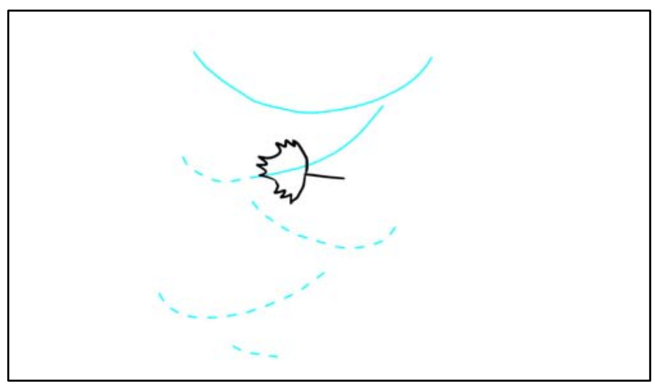
Description: This storyboard frame is similar to 21–*Bird swimming*, but the wings move differently and the bird stays in place as other objects move by.

Animating in K-Sketch

Time: 20 min. 42 sec.

Notes: This animation was hard to produce because K-Sketch does not provide the *define cels* animation operation. I drew three images in succession by drawing and erasing to define the cycle. After that, I would copy the needed drawing in a previous frame and paste it in the current frame to continue the cycle. Also, it was difficult to make objects in the background move in exactly the same way to give the illusion of the bird's movement. If K-Sketch had provided the *zoom* operation, I could have zoomed out, drawn a large background scene, and translated all background objects together.

B.23 Leaf Falling



User category: artist

Goal category: prototype

Source: Animator 3

Length: 13.5 sec.

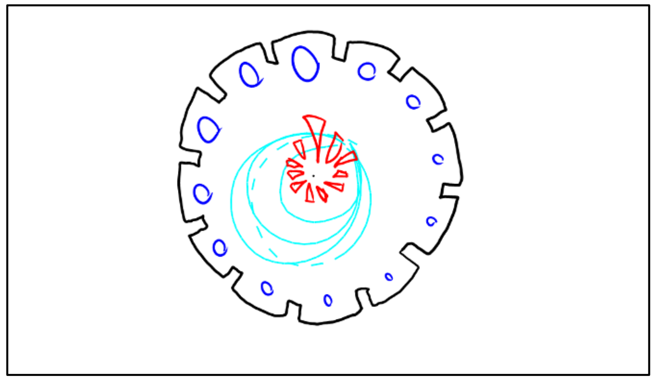
Description: This is a character sketch of a leaf falling gracefully to the ground.

Animating in K-Sketch

Time: 1 min. 3 sec.

Notes: This animation was done easily with five *orient to path* operations.

B.24 Spinning Wheel



User category: amateur

Goal category: doodle

Source: child animator

Length: 3.7 sec.

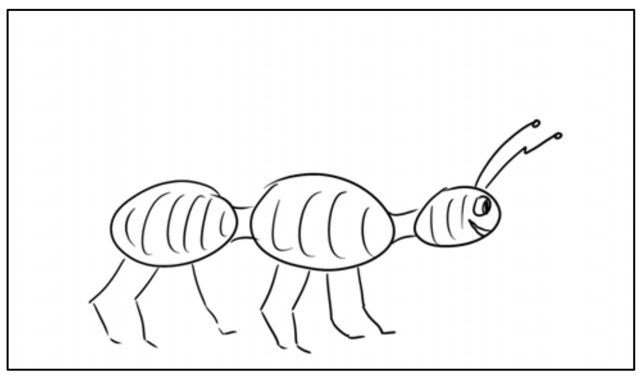
Description: This is a doodle with four decorated wheels that spin around.

Animating in K-Sketch

Time: 5 min. 55 sec.

Notes: Most of the time spent on this animation was drawing the decorated wheels. The spinning motion was trivially easy.

B.25 Ant Walking



User category: amateur

Goal category: doodle

Source: child animator

Length: 0.8 sec.

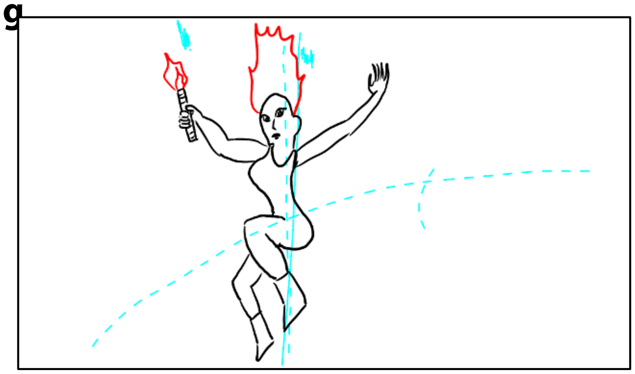
Description: This is a rough, 4-frame animated doodle of a seven-legged ant walking.

Animating in K-Sketch

Time: 3 min. 57 sec.

Notes: The four frames of this animation were done by copying the original body of the ant. The head was moved up slightly in the third frame. The leg movement and the subtle mouth changes were done by erasing and redrawing.

B.26 Lady With Torch Tumbling



User category: amateur

Goal category: doodle

Source: child animator

Length: 5.9 sec.

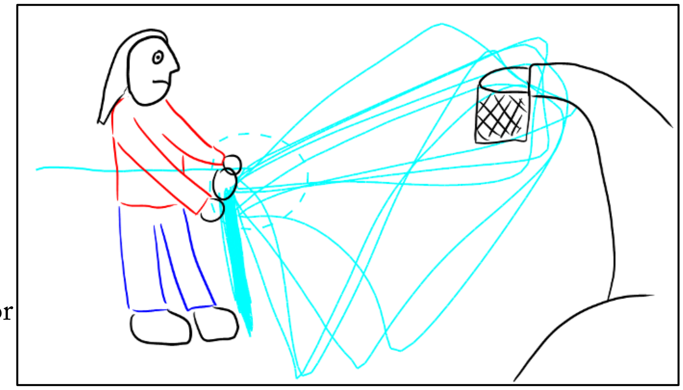
Description: This is a doodle of a lady with a torch flying across the screen (once down and once from right to left with a slight spin). Her hair and the flame of her torch flutter as they move.

Animating in K-Sketch

Time: 11 min. 14 sec.

Notes: I drew the lady first and then added a flutter to the torch and her hair by adding a non-uniform *scale* to both (after moving the center point to the base of each). I then selected the entire lady and moved the handle to her feet, allowing me to drag her in from off screen. I had to move the handle to the top of her hair to push her off screen, and this proved difficult to do, because I could not get the handle at an earlier moment in time due to a bug. To move the handle, I eventually put the tablet in portrait mode so that the handle would be visible. I then combined a *translate* and a *rotate* to get the tumbling motion. Managing the off-screen movement would have been easier if I could have *zoomed* out. Also, I could have represented the original drawing more faithfully if I had been able to fill regions with color or *define backgrounds.*

B.27 She Shoots and She…



User category: amateur

Goal category: entertain

Source: child animator

Length: 64.3 sec.

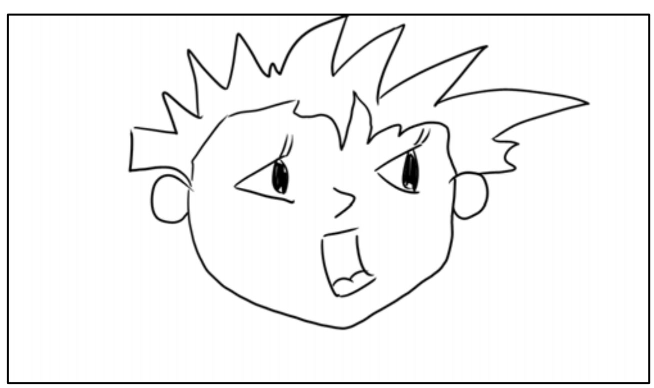
Description: This animation tells the story of a girl who has trouble getting a basketball through a hoop. She falls into despair, but then she gets up, visualizes the ball going through the hoop, and finds success. She does a flip in the air and then moves off screen.

Animating in K-Sketch

Time: 40 min. 55 sec.

Notes: This is a long animation with many parts, and it would have benefitted from *add scene*. Most motions were done easily with *translation* and redrawing. The tears were done with *copy motion*, and the spinning ball was done with non-uniform *scale*. The large number of strokes produced by erasing and redrawing made K-Sketch run slowly. This animation also uncovered two bugs. One made parts of characters disappear, and another made certain copied motions move faster than they should have.

B.28 Talking Heads



User category: amateur

Goal category: doodle

Source: child animator

Length: 12.1 sec.

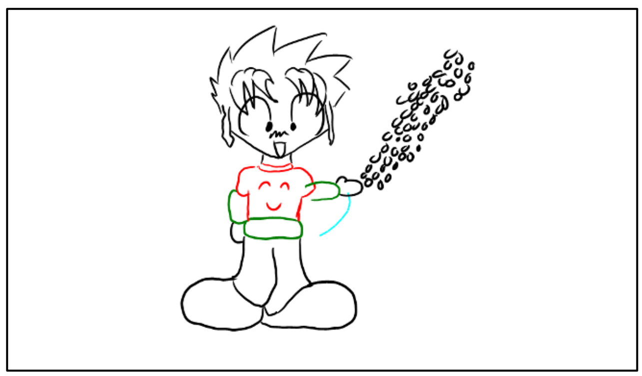
Description: A child created this doodle to experiment with lip movement and facial expressions. It shows two talking heads in series.

Animating in K-Sketch

Time: 23 min. 17 sec.

Notes: The mouth movements and expressions in this animation changed so quickly that it was as easy to redraw every frame individually as it would have been to transform them in some way. Copying previous drawings saved some time, but there was so much detail that the animation still took fairly long to make. Drawing the faces took a fairly long time, as well.

B.29 Zeig's Bad Hair Day



User category: amateur

Goal category: entertain

Source: child animator

Length: 20.8 sec.

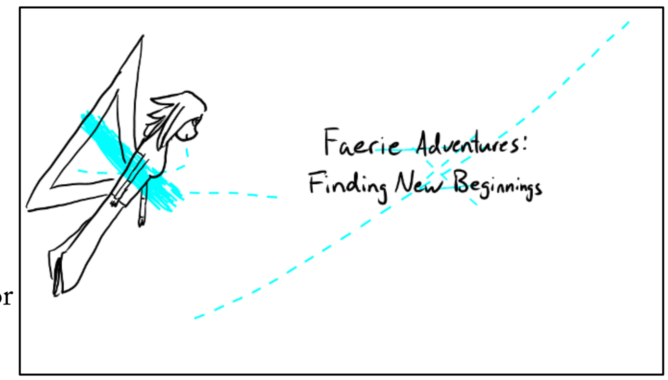
Description: This short movie tells the story of Zeig, who does battle with a dragon. Zeig waves at the camera, then sees the dragon breathing fire. The camera zooms on the dragon. Zeig gets his hair singed. He retaliates by shooting a beam of particles at the dragon, which extinguishes the dragon's flame and kills him. Zeig waves at the camera in victory.

Animating in K-Sketch

Time: 33 min. 55 sec.

Notes: The motions in this animation were very easy to represent in K-Sketch with *rotates* and *scales*. Difficulties surfaced when trying to manipulate objects that had grown larger than the screen (*zoom* would have made this easy) and when the animation grew so large that the interface became unresponsive. To be completely true to this child's vision, K-Sketch would have needed *define background*, *play sound*, and *occlude*.

B.30 Faerie Adventures



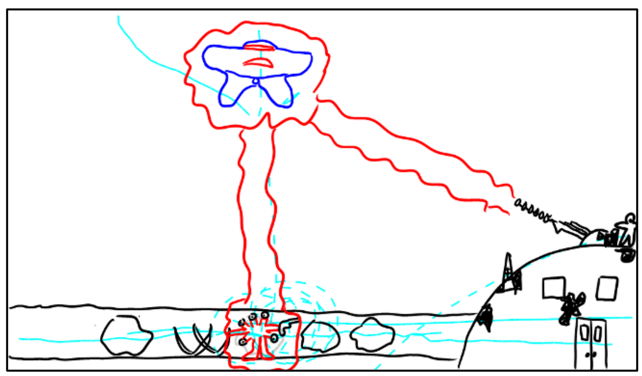
User category: amateur
Goal category: entertain
Source: child animator
Length: 63.1 sec.
Description: This movie tells the story of a faerie making an unexpected friend. A faerie if flying through fields and comes upon a forest. Peering into the forest, she spies a hoof and a wagging tail. Puzzled, she flies in to investigate. Suddenly, she realizes that she has grabbed hold of the tail belonging to an angry, fiery dog. The dog pounces on her, and KABOOM! The dog is licking her face and wagging its tail, quite happy. The dog then gives the faerie a gift of a large gem. The two become friends and walk off into the sunset together.

Animating in K-Sketch

Time: 67 min. 17 sec.
Notes: This animation took a long time to produce, primarily because of the many detailed drawings required and K-Sketch's sluggishness when handling detailed drawings. Even so, this animation shows that long, detailed animations are possible with K-Sketch in a reasonable time. Most motions were simple *translates*, *scales*, or *rotates*. The faerie's flapping wings were done in seconds with a non-

uniform *scale* that K-Sketch automatically put in the correct reference frame. (This would have taken 40 key frames in a traditional tool.) The tongue of the dog when licking the faerie's face has both a rotate and a subtle scale, and this required the *fix* dialog. Moving things from off screen to off screen required me to move the handle into a special position. Unfortunately, this animation leaves out some parts of the creator's vision, because most backgrounds could not be shown without *occlude* or *define background.* The very first scene also includes a long background that flies by the faerie as she flies, and this was not possible without *zoom.* Also, this animation triggered a bug that made it seem impossible to erase certain objects. This is why stray marks appear in the animation for a short time.

B.31 The Planet of the Robots



User category: amateur

Goal category: entertain

Source: child animator

Length: 53.2 sec.

Description: This movie tells a story about an alien invasion and the hero that saves the day. The opening scene shows the hero taking refuge in a bunker. Then, a large alien ship descends and drops an alien on the ground. The ship terrorizes another person as the alien shoots at the bunker. Both invaders then move off temporarily, and our hero emerges from the bunker. As he makes his way right, the ship returns and fires on him as he evades three times. The scene then changes to show another bunker with a gun. Our hero approaches the bunker, but the ground alien is in his way. As the alien approaches, he is alarmed and hides in the bushes. The alien passes by without noticing, and our hero makes a break for the bunker. He climbs to the top as the ground alien moves elsewhere. Then the flying alien returns and fires on the bunker. Our hero returns fire, paralyzing the ship. The ground alien returns, and he is also paralyzed. When the beam stops, the flying alien moves off, while the ground alien's gun tumbles into the air, falls, hits him, and kills him, ricocheting off to the right. We then see a view of the first scene at peace, and finally have a view of the hero's face as he celebrates victory.

Animating in K-Sketch

Time: 57 min. 42 sec.

Notes: While this movie had a huge number of events, K-Sketch was able to handle the coordination and movement easily. There were several movements with spins, which the *fix* dialog handled easily. The animation would have been more true to the creator's vision if it had *define background* had been available. As in other scenarios, most of the time taken to make this animation was spent drawing complex scenes and waiting for K-Sketch to respond, as the complex drawings trigger performance problems. This animation also triggered bugs in K-Sketch that make erased objects reappear and cause the center point of rotating objects to be computed incorrectly.

APPENDIX

C Interface Optimization Data

C.1 Full List of Animation Operations

Most of this dissertation uses an abbreviated set of 18 animation operations. The full set of 31 animation operations also includes five *variants* and eight *additional operations* that were omitted from previous discussions for brevity. The five variants were limited versions of *translate*, *rotate*, *scale*, and *set timing* that enabled me to compare my designs to other animation tools. The eight additional operations made fairly obvious and independent improvements to most animation interfaces.

The following table lists all animation operations. Variants are shown in italics and are grouped by type (*translate, rotate, scale,* or *set timing*). Each group of variants is followed by a single, non-italicized operation of the same type, which is the sole operation of this type that appeared elsewhere in this dissertation. Additional operations appear at the end, separated from other operations, and are also in italics.

#	Animation Operation	Description
1	*translate: straight*	*Move object along a straight path.*
2	translate: arbitrary	Move object along an arbitrary path.
3	*scale: uniform*	*Resize an object uniformly in both dimensions.*
4	scale: full	Resize an object either uniformly or non-uniformly.
5	*rotate: limited*	*Rotate an object less than 180 degrees about an unspecified center.*
6	*rotate: any center*	*Rotate an object through any angle about an unspecified center.*
7	rotate: full	Rotate an object through any angle about a specific center.
8	*set timing: acceleration*	*Specify the timing of a motion with an optional acceleration/deceleration command.*

#	Animation Operation	Description
9	set timing: continuous	Specify the timing of a motion continuously throughout the motion.
10	move relative	Move object(s) relative to some reference frame.
11	appear	Make an object appear at any moment in time.
12	disappear	Make an object disappear at any moment in time.
13	trace	Make a line appear over time as if it were being drawn, or disappear as if it were being erased.
14	repeat motion	Repeat all or part of a motion until the object no longer exists, another motion starts, or the animation ends.
15	copy motion	Select one object's motion (or set of motions) and apply it to another object (or same object at different time).
16	define cels	Create a sequence of cels to define a character's motions.
17	morph	Define two objects and let the system compute motion that changes the first into the second.
18	physically simulate	Specify objects' physical characteristics and let the system compute motion.
19	interpolate	Define an object's beginning and ending positions and let the system compute motion.
20	move forward/back	Move an object in front of or behind other objects.
21	deform	Bend an object from its original state.
22	move limb	Define a skeleton, each node associated with an object, and move a single limb.
23	orient to path	Move object along a curved path, pointing down the path.
24	*play sound*	*Make a sound start at a specified time.*
25	*occlude*	*Make an object occlude others.*
26	*repeat playback*	*Repeat the entire animation when playing back.*
27	*zoom*	*Zoom in/out (possibly to manipulate object off-frame).*
28	*copy object*	*Make one or more copies of an object.*
29	*add scene*	*Add a new scene that plays after the current scene.*
30	*import*	*Define a formal object by bringing it in from another application.*
31	*define background*	*Define a background that cannot be selected with other objects.*

C.2 Coded Scenario Library

The following tables present the coding of the scenario library that I described in Section 4.1 on page 88. Recall that each scenario is divided into features, and multiple approaches may exist for representing each feature. The first approach listed for each feature is the preferred approach, and others may require extra steps. Each approach is tagged with the animation operations required by that approach (tags are shown here by black dots in the cell corresponding to an animation operation.

Note that the animation operations in these tables appear at first glance to be in reverse order. These tables are intended to be read by rotating this document 90° clockwise, in which case the animation operations are in order from left to right.

Scenario	1–Simple potbelly fall			2–Simple potbelly crash					3–King leafy entrance									4–Complex potbelly fall						5–Potbelly heads out				
Feature	move frame	sound	storyboard context	move frame	sound	multiple parts		storyboard context	drawings	sound	enter/exit	spectator motion	herald eyes/water	herald motion		leafy motion	storyboard context	potbelly motion		drawings	sound	exit	storyboard context	change/motion	drawings	sound	exit	storyboard context
Approach	trans.	play sound	scenes	trans.	play sound	appear, disapp.	scenes	scenes	import	play sound	zoom	trans.	multiple	multiple, scale	multiple, deform	multiple	scenes	trans. cont.	trans.	import	play sound	zoom	scenes	trans. cont.	import	play sound	zoom	Scenes
Define background																				●					●			
Import									●											●					●			
Add scene			●				●	●									●						●					●
Copy object																												
Zoom											●											●					●	
Repeat playback																												
Occlude									●											●					●			
Play sound		●			●					●											●					●		
Orient to path																												
Move limb																												
Deform															●													
Move forward/back																												
Interpolate																												
Physically simulate																												
Morph																												
Define cels																												
Copy motion																												
Repeat motion																												
Trace																												
Disappear						●							●											●				
Appear						●							●											●				
Move relative														●	●													
Set timing: continuous																		●										
Set timing: acceleration																●												
Rotate: full																												
Rotate: any center																												
Rotate: limited														●	●													
Scale: full														●														
Scale: uniform																												
Translate: arbitrary																●												
Translate: straight	●			●								●		●	●			●	●					●				
Extra operations																			4									

Scenario	6–Potbelly collision					7–Hitchhiker					8–Sea animals					9–Security Card Slide						10–Perpetrator Entrance		
Feature	ball		potbelly motion			arm movements			rep.		drawings	animal movement	rep. movement		Storyboard context	artwork	Storyboard context	arm movements				artwork	Storyboard context	perp movement
Approach	trans, dis	trans, scenes	deform	redraw	physical sim	deform cont.	redraw	Morph	all	parts	multiple	smooth curve	rep. parts	no rep.	scenes	occlude	scenes	limbs	move relative	deform	redraw	occlude	scenes	trans., zoom
Define background											●													
Import											●													
Add scene		●													●		●						●	
Copy object																								
Zoom			●																					●
Repeat playback									●															
Occlude											●					●						●		
Play sound																								
Orient to path																								
Move limb																		●						
Deform			●			●														●				
Move forward/back																								
Interpolate																								
Physically simulate					●																			
Morph								●																
Define cels																								
Copy motion																								
Repeat motion										●			●											
Trace																								
Disappear	●			●			●														●			
Appear				●			●														●			
Move relative			●																●					
Set timing: continuous						●																		
Set timing: acceleration												●												
Rotate: full						●													●					
Rotate: any center																								
Rotate: limited																								
Scale: full																								
Scale: uniform																								
Translate: arbitrary												●	●	●										
Translate: straight	●	●	●																					●
Extra operations				11			4	8						2							4			

Scenario	11–Perpetrator Turning					12–Sorcerer Crystal Ball				13–Car steering			14–Car on country road					15–Man watching sun		16–Thermometer with more				17–Translate Rotate Scale		
Feature	artwork	Storyboard context	perp movement		Enter/Exit	Artwork	Storyboard context	hand movement		Car movement			Storyboard context	Enter/Exit	car motions	Multiple parts		sun /man motion		thermometer			more	Car movement		
Approach	occlude	scenes	orient	trans, rot	zoom	occlude	scenes	cels, rep.	draw, copy	orient	trans, rot	straight trans	scenes	zoom	trans, scale	appear, disapp.	scenes	appear, disapp.	scenes	stretch	deform	trace	scale	Orient	rotate	straight
Define background																										
Import																										
Add scene		●					●						●				●		●							
Copy object									●																	
Zoom					●									●												
Repeat playback																										
Occlude	●					●																				
Play sound																										
Orient to path			●							●														●		
Move limb																										
Deform																					●					
Move forward/back																										
Interpolate																										
Physically simulate																										
Morph																										
Define cels								●																		
Copy motion																										
Repeat motion								●																		
Trace																						●				
Disappear									●							●		●								
Appear									●							●		●								
Move relative				●							●				●									●	●	●
Set timing: continuous																										
Set timing: acceleration																										
Rotate: full																										
Rotate: any center																										
Rotate: limited				●							●														●	
Scale: full																				●						
Scale: uniform															●								●	●	●	●
Translate: arbitrary			●	●						●	●				●			●	●					●	●	
Translate: straight												●														●
Extra operations									6		1	4							3							12

Scenario / Feature / Approach	18–Car bobbing / car path / trans.	car bob / cel cycle	scale	scale no rep.	deform	19–Word collision / Off-screen / zoom	Collision / trans.	physical sim	20–Simple Seagull / Storyboard context / scenes	Exit / zoom	Bird movement / redraw, trans	21–Bird swimming / Storyboard context / scenes	Bird path / trans.	bird cycle / cels	22–Bird flying / Storyboard context / scenes	bird cycle / cels	background / trans.	23–Leaf falling / leaf motion / cont. trans, rot	physical sim	accel, orient	24–Spinning Wheel / Wheel design / Import	Wheel motion / Spin	multiple wheels / appear, disapp.	Scenes	Repetition / rep. all
Define background																									
Import																					●				
Add scene									●			●			●									●	
Copy object																									
Zoom						●				●															
Repeat playback																									●
Occlude																									
Play sound																									
Orient to path																				●					
Move limb																									
Deform					●																				
Move forward/back																									
Interpolate																									
Physically simulate								●											●						
Morph																									
Define cels		●												●		●									
Copy motion																									
Repeat motion		●	●		●									●		●									
Trace																									
Disappear											●												●		
Appear											●												●		
Move relative																		●							
Set timing: continuous				●														●							
Set timing: acceleration																				●					
Rotate: full																									
Rotate: any center																						●			
Rotate: limited																		●							
Scale: full			●	●																					
Scale: uniform																									
Translate: arbitrary													●					●		●					
Translate: straight	●						●				●						●								
Extra operations				4	2															6					

Scenario	25–Ant Walking						26–Lady with torch tumbling																	
Feature	Leg and head						Lady's path			Flaming hair												Artwork		Repetition
Approach	move limb, rep.	move limb	move rel	frame by frame	rep., frames	move rel, rep.	orient	Move relative	Straight parts	cels	scale	scale, cont.	scale, rep.	deform	deform, cont.	deform, rep.	morph	morph, rep.	rotate	rotate, cont.	rotate, rep.	Import	drawing	rep. all
Define background																								
Import																						●		
Add scene																								
Copy object																								
Zoom							●	●	●															
Repeat playback	●				●	●																		●
Occlude																							●	
Play sound																								
Orient to path							●																	
Move limb	●	●																						
Deform														●	●	●								
Move forward/back																								
Interpolate																								
Physically simulate																								
Morph																	●	●						
Define cels										●														
Copy motion																								
Repeat motion										●			●			●		●			●			
Trace																								
Disappear				●	●																			
Appear				●	●																			
Move relative			●			●		●			●	●	●	●	●	●	●	●	●	●	●			
Set timing: continuous												●			●					●				
Set timing: acceleration																								
Rotate: full																			●	●	●			
Rotate: any center																								
Rotate: limited								●	●															
Scale: full																								
Scale: uniform											●	●	●											
Translate: arbitrary							●	●																
Translate: straight									●															
Extra operations		18	18	126	18						10	2		10	2		10		10	2				

Scenario	27–She Shoots and She…																28–Talking Heads						
Feature	Artwork		Ball motion		arm movements			Tears			ball spin in place	Other parts	Enter/Exit	Multiple parts	Spin		mouth, eyes			Multiple heads		Imported Drawings	Repetition
Approach	import	occlude	trans.	physical sim	relative	limbs	deform	cels, rep.	trans. s., rep	trans. s., copy	appear, disapp.	appear, disapp.	zoom	scenes	one motion	two motions	cels	scale	scale, scenes	scenes	appear, disapp.	import	rep. all
Define background	●																						
Import	●																					●	
Add scene														●					●	●			
Copy object														●									
Zoom													●										
Repeat playback																							●
Occlude		●										●											
Play sound																							
Orient to path																							
Move limb						●																	
Deform							●																
Move forward/back																							
Interpolate																							
Physically simulate				●																			
Morph																							
Define cels								●									●						
Copy motion										●													
Repeat motion								●	●								●						
Trace																							
Disappear											●	●						●			●		
Appear											●	●						●			●		
Move relative					●																		
Set timing: continuous			●		●													●	●				
Set timing: acceleration																							
Rotate: full					●																		
Rotate: any center																●							
Rotate: limited															●								
Scale: full																		●	●				
Scale: uniform														●									
Translate: arbitrary																							
Translate: straight			●						●	●			●										
Extra operations																1							

Scenario	29–Zeig's Bad Hair Day											30–Faerie Adventures												
Feature	Artwork	Multiple parts	Sound	Fire			arm movements			magic ray		Artwork	Sound	Multiple Parts	Flapping wings			Move background	Faerie/dog path		Emphasis	wag tail/tongue		Kaboom
Approach	import	scenes	sound	cels	scale, cont.	scale, rep.	rotate, cont.	rotate, rep.	rotate	cels	copied bits	import	sound	scenes	scale, rep.	scale, cont.	cels	trans.	trans., rot	orient	scale	rot, trans cont	rot, trans	appear
Define background	●											●												
Import	●											●												
Add scene		●												●										
Copy object		●									●			●										
Zoom																		●						
Repeat playback																								
Occlude												●												
Play sound			●										●											
Orient to path																				●				
Move limb																								
Deform																								
Move forward/back																								
Interpolate																								
Physically simulate																								
Morph																								
Define cels				●													●							
Copy motion											●													
Repeat motion				●		●		●							●		●							
Trace																								
Disappear										●	●							●						
Appear										●	●													●
Move relative															●	●			●					
Set timing: continuous					●		●									●						●		
Set timing: acceleration																								
Rotate: full							●	●	●													●	●	
Rotate: any center																								
Rotate: limited																			●					
Scale: full					●	●									●	●								
Scale: uniform		●																			●			
Translate: arbitrary																				●				
Translate: straight											●								●			●	●	
Extra operations									4							7							18	

Scenario	31–Planet of the Robots										32–Schmoo skateboard									
Feature	movement			Background	Multiple parts		Other parts	arm movements			Jump				Slide/Split/Fall		Skateboard tumble		Multiple sections	
Approach	all	orient	trans, rot	drawing	scenes	appear, disapp.	appear, disapp.	relative rotation	limbs	deform	appear, disapp.	multiple	multiple, deform	morph	trans, rot	morph	trans, rot	physical sim	appear, disapp.	scenes
Define background				●																
Import																				
Add scene					●															●
Copy object					●	●														
Zoom	●	●	●												●	●				
Repeat playback																				
Occlude				●											●	●				
Play sound																				
Orient to path	●	●																		
Move limb									●											
Deform										●			●							
Move forward/back																				
Interpolate																				
Physically simulate																		●		
Morph														●		●				
Define cels																				
Copy motion																				
Repeat motion																				
Trace																				
Disappear	●	●	●			●	●				●								●	
Appear	●	●	●			●	●				●								●	
Move relative	●		●					●				●	●		●		●			
Set timing: continuous	●	●	●					●	●	●										
Set timing: acceleration																				
Rotate: full								●												
Rotate: any center	●		●														●			
Rotate: limited												●	●		●					
Scale: full												●	●							
Scale: uniform																				
Translate: arbitrary	●	●	●																	
Translate: straight												●	●		●		●			
Extra operations																				

Scenario	33–The Trickster Mummy															34–Rocket and palm trees										
Feature	Background	Enter/Exit	Multiple parts			Google eyes	eyes roll, fly			Limb stretch				Eyebrows		Rocket/Brick	Bird Head			Exhaust			Palm trees		Kaboom	
Approach	drawing	trans.	scale	scenes	appear, disapp.	trans.	trans, rot, rep.	trans, rot	physical sim	deform, rep.	rot, scale, rep.	deform	rot, scale	deform	rotate	simple	appear, disapp.	deform	morph	appear, disapp.	cels	Trans, scale	cels	scale	appear, disapp.	scenes
Define background	●																									
Import																										
Add scene				●																						●
Copy object				●	●																		●	●		
Zoom		●	●													●						●				
Repeat playback																										
Occlude	●															●				●	●	●	●	●		
Play sound																										
Orient to path																										
Move limb																										
Deform										●		●		●				●								
Move forward/back																										
Interpolate																										
Physically simulate									●																	
Morph																			●							
Define cels																					●		●			
Copy motion																							●	●		
Repeat motion							●			●	●															
Trace																										
Disappear					●												●			●					●	
Appear					●												●			●					●	
Move relative							●	●																		
Set timing: continuous																										
Set timing: acceleration																										
Rotate: full											●		●		●											
Rotate: any center							●	●																		
Rotate: limited																										
Scale: full											●		●													
Scale: uniform			●																			●		●		
Translate: arbitrary						●	●	●																		
Translate: straight		●														●						●				
Extra operations								4				2	6													

Scenario	Feature	Approach	Define background	Import	Add scene	Copy object	Zoom	Repeat playback	Occlude	Play sound	Orient to path	Move limb	Deform	Move forward/back	Interpolate	Physically simulate	Morph	Define cels	Copy motion	Repeat motion	Trace	Disappear	Appear	Move relative	Set timing: continuous	Set timing: acceleration	Rotate: full	Rotate: any center	Rotate: limited	Scale: full	Scale: uniform	Translate: arbitrary	Translate: straight	Extra operations
35–Small creature whomp	walking	trans., rep.					●		●											●				●								●		
		trans.					●		●															●								●		2
	club	appear, disapp.																				●	●											
	pow	multiple																				●	●											
	blast	multiple							●													●	●								●			
	fly away	multiple									●											●									●	●		
	star	scale																				●	●					●			●			
36–Monster Cop	Background	drawing	●						●																									
	Enter/Exit	zoom, move					●																											
	Multiple parts	scale					●																								●			
		scenes			●	●																												
		appear, disapp.				●																●	●											
	Monster move	trans.																															●	
	arm move	limbs										●																						
		deform											●																					
		appear, disapp.																				●	●											14
	Cop movement	deform											●												●							●		
		limbs										●													●							●		
		move rel, rot																						●	●		●					●		
	Shoving into jail	trans., rep.												●						●													●	
		trans., contin												●											●								●	2
		trans.												●																			●	12
37–Crawling 'O' Creature	motion	appear, disapp.																				●	●											
		limbs										●																						
		orient									●																					●		
	Enter	Zoom					●																											
38–The Apple	all	simple																				●	●											
		scenes			●																													

Scenario	39–Cobalt							40–Hello from Instructor							41–Hatching from egg												
Feature	Multiple heads		stars		mouth		Artwork	rise up	hair		eyes	Artwork		Background	egg expanding			break out		wave			fly away		egg tumble	multiple parts	
Approach	scenes	no scenes	move w/rep.	move cont.	scale w/rep.	scale cont.	import	trans.	rotate	deform	all	import	occlude	background	scale, rep.	appear, disapp.	scale, cont.	appear, disapp.	morph	rotate, cont.	rotate, rep.	appear, disapp.	trans., disapp.	trans., off frame	trans, rot	appear, disapp.	scenes
Define background														●													
Import	●	●					●					●															
Add scene	●																										●
Copy object																											
Zoom								●																●			
Repeat playback																											
Occlude													●														
Play sound																											
Orient to path																											
Move limb																											
Deform										●																	
Move forward/back																											
Interpolate																											
Physically simulate																											
Morph																			●								
Define cels																											
Copy motion																											
Repeat motion			●		●										●						●						
Trace																											
Disappear		●									●					●		●				●	●			●	
Appear		●									●					●		●				●				●	
Move relative																											
Set timing: continuous				●		●											●			●							
Set timing: acceleration																											
Rotate: full									●											●	●						
Rotate: any center																											
Rotate: limited																									●		
Scale: full																											
Scale: uniform					●	●									●		●										
Translate: arbitrary																											
Translate: straight			●	●				●			●												●	●	●		
Extra operations																4						14					

Approach	cels	deform	move limb	appear, disapp.	scale, cont.	scale, rep.	appear, disapp.	scale	appear, disapp.	appear, disapp.	scale	appear/disapper	appear, disapp.	scenes
Scenario	42–I'm Hungry													
Feature	arm move				mouth move			table appear/grow		speech balloons	walk in/away		multiple parts	
Define background														
Import														
Add scene														●
Copy object														
Zoom														
Repeat playback														
Occlude														
Play sound														
Orient to path														
Move limb			●											
Deform		●												
Move forward/back														
Interpolate														
Physically simulate														
Morph														
Define cels	●													
Copy motion														
Repeat motion						●								
Trace														
Disappear				●			●		●	●		●	●	
Appear				●			●		●	●		●	●	
Move relative														
Set timing: continuous					●									
Set timing: acceleration														
Rotate: full														
Rotate: any center														
Rotate: limited														
Scale: full														
Scale: uniform					●	●		●			●		●	
Translate: arbitrary														
Translate: straight														
Extra operations				1			6		4			12		

Define background																
Import																
Add scene	●															
Copy object	●	●														
Zoom															●	
Repeat playback																
Occlude																
Play sound																
Orient to path																
Move limb						●					●					
Deform								●	●							
Move forward/back																
Interpolate																
Physically simulate													●			
Morph										●						
Define cels																
Copy motion				●												
Repeat motion			●													
Trace																
Disappear		●														
Appear		●														
Move relative			●	●	●		●					●				
Set timing: continuous																
Set timing: acceleration																
Rotate: full			●	●	●											
Rotate: any center												●				
Rotate: limited																
Scale: full																
Scale: uniform														●	●	
Translate: arbitrary																
Translate: straight			●	●	●											●
Extra operations					12											
Approach	scenes	appear, disapp.	w/rep.	w/copy	no rep.	limbs	relative rotation	deform	deform	morph	limbs	relative rotation	physical sim	scale	scale, zoom	scale
Feature	Multiple parts		Tumble			arm move			tumble to ball					growing text	walking away	walking up
Scenario	43–Acrobatics															

Scenario	44–Spy Story														45–Tidal Wave					
Feature	Bomb appear		fuse	Multiple parts		explosion	moving car/plane	parachute appear	parachute move	parachute land					truck		plane, explosion		wave	Enter/Exit
Approach	zoom, trace	zoom, lines	trace (erase)	scenes	appear, disapp.	concentric lines	trans.	occlude	trans.	deform	morph	redraw	trans chute	physical sim	trans.	appear, disapp.	trans.	simple	simple	zoom
Define background																				
Import																				
Add scene				●																
Copy object				●	●															
Zoom							●													●
Repeat playback																				
Occlude								●					●							
Play sound																				
Orient to path																				
Move limb																				
Deform										●										
Move forward/back																				
Interpolate																				
Physically simulate														●						
Morph											●									
Define cels																				
Copy motion																				
Repeat motion																				
Trace	●		●																	
Disappear					●							●	●			●	●	●	●	
Appear					●	●						●	●			●	●	●	●	
Move relative															●					
Set timing: continuous																				
Set timing: acceleration																				
Rotate: full																				
Rotate: any center																				
Rotate: limited																				
Scale: full																				
Scale: uniform	●	●													●					
Translate: arbitrary																	●			
Translate: straight							●		●				●		●					
Extra operations		10										13	4							

Scenario	46–Plane Story										47–Dynamic Programming										48–Our Algorithm				49–Line Music			
Feature	Multiple parts		plane turn around	plane fly	sun		explosion		Person appear		multiple parts		Box motion		Stretchable line						multiple parts		Stretchable line		objs enter, leave	Dance move		music
Approach	scenes	appear, disapp.	appear, disapp.	trans+rot, scale	appear, disapp.	and scale	trace	appear, disapp.	trace	appear	appear, disapp.	scenes	trans., copy	trans.	morph, copy	rot, scale, copy	redraw, copy	morph	rot, scale	redraw	appear, disapp.	scenes	trace	scale	multiple	deform	morph	sound
Define background																												
Import																												
Add scene	●											●										●						
Copy object	●	●																										
Zoom				●																					●			
Repeat playback																												
Occlude																												
Play sound																												●
Orient to path																												
Move limb																												
Deform																										●		
Move forward/back																												
Interpolate																												
Physically simulate																												
Morph															●			●									●	
Define cels																												
Copy motion													●		●	●	●											
Repeat motion																												
Trace							●		●														●		●			
Disappear		●	●		●	●		●			●						●			●	●				●			
Appear		●	●		●	●		●		●	●						●			●	●				●			
Move relative				●												●			●									
Set timing: continuous				●																								
Set timing: acceleration																												
Rotate: full																●			●									
Rotate: any center																												
Rotate: limited				●																								
Scale: full																●			●					●				
Scale: uniform				●		●																						
Translate: arbitrary				●																								
Translate: straight													●	●														
Extra operations								3		2				14		5	20	5	10	40								

Scenario	50–Meiosis												51–Detect. Distant Planets		52–Gear Interference			53–Lattice slip		54–Contra Dance								
Feature	Zoom in	cpy chromasome	bend objects					move into place		emphasis, other	stretchy bits	re-arrange	star/planet move	waves	gear movement	background	gear image	slip		circle	step in		swing out		ladies' chain		swing out 2	
Approach	scale	copy	morph	deform	limbs	move relative	redraw	morph	trans, rot	redraw	trace	trans.	trans., rep.	multiple	rotate	lines	import	physical sim	interpolate	trans	trans, cont	trans	trans, copy	trans	trans, copy	trans	trans, copy	trans
Define background	●															●												
Import																	●											
Add scene																												
Copy object		●	●																									
Zoom	●														●													
Repeat playback																												
Occlude																●												
Play sound																												
Orient to path																												
Move limb					●																							
Deform				●																								
Move forward/back																												
Interpolate														●					●									
Physically simulate																		●										
Morph			●					●																				
Define cels																												
Copy motion														●									●		●		●	
Repeat motion													●	●														
Trace											●																	
Disappear							●			●				●					●									
Appear							●			●				●														
Move relative						●			●																			
Set timing: continuous																					●							
Set timing: acceleration																												
Rotate: full															●													
Rotate: any center						●																						
Rotate: limited									●																			
Scale: full																												
Scale: uniform	●																											
Translate: arbitrary												●	●							●	●	●	●	●	●	●	●	●
Translate: straight									●					●					●									
Extra operations				24	24	24	16															4						

Scenario	55–Chem: Particle Collisions				56–Chem: Rusting Reaction							57–Chem: Battery Reaction										58–Constr. equip. tread					
Feature	Multiple parts		Motion		moving label	growing rust		water droplet		decay		electron motion			switch	ion motion		ion grow/shrink	anode/cathode			tread motion					line
Approach	appear, disapp.	scenes	trans.	physical sim	trans.	morph	scale	morph	redraw	morph	redraw	path, copy, rep.	path, copy	path	appear, disapp.	tr, copy, rep.	trans.	scale	morph	deform	appear, disapp.	trans, rot, cont	trans, rot	orient, cont	orient	physical sim	trace
Define background																											
Import																											
Add scene		●																									
Copy object																											
Zoom																											
Repeat playback																											
Occlude												●	●	●													
Play sound																											
Orient to path																								●	●		
Move limb																											
Deform																				●							
Move forward/back																											
Interpolate																											
Physically simulate				●																						●	
Morph						●		●		●									●								
Define cels																											
Copy motion												●	●			●											
Repeat motion												●				●											
Trace																											●
Disappear	●								●		●				●						●						
Appear	●								●		●				●						●						
Move relative																						●	●				
Set timing: continuous																						●		●			
Set timing: acceleration																											
Rotate: full																											
Rotate: any center																											
Rotate: limited																						●	●				
Scale: full																											
Scale: uniform							●											●									
Translate: arbitrary												●	●	●								●		●			
Translate: straight			●		●											●	●						●		●		
Extra operations									3		3						3				3		8		4		

Scenario	59–Cantor Set Construction	60–Bouncing Ball			61–Jumping Rope						62–Face Singing				63–Washing plate			64–Robot Arm	65–Casing Slide		66–Gear reduction					
Feature	all	bounce		boing boing	jump	rope		rep.		1 2 buckle my shoe	mouth	Laaa	rep.		elements	movements		movements	Objects	Slide	gear rotation				gear image	
Approach	multiple	trans.	physical sim	sound	trans.	scale	morph	all	motion	sound	scale	sound	all	motion	occlude	trans., rep.	trans.	limbs	occlude	trans., front	rot., rep.	rot,, copy	rot., interpolate	rotate	occlude	import
Define background																										
Import																										●
Add scene																										
Copy object																									●	
Zoom																										
Repeat playback	●							●					●													
Occlude															●				●						●	
Play sound				●						●		●														
Orient to path																										
Move limb																		●								
Deform																										
Move forward/back																				●						
Interpolate																							●			
Physically simulate			●																							
Morph							●																			
Define cels																										
Copy motion																						●				
Repeat motion									●					●		●					●					
Trace																										
Disappear	●																									
Appear																										
Move relative																										
Set timing: continuous		●																								
Set timing: acceleration																										
Rotate: full																					●	●	●	●		
Rotate: any center																										
Rotate: limited																										
Scale: full						●					●															
Scale: uniform																										
Translate: arbitrary		●																								
Translate: straight					●											●	●			●						
Extra operations																	2							11		

Approach	appear, disapp.	scenes	orient	trans, rot	occlude	scenes	appear, disapp.	cels	appear, disapp.	trans.	trans., copy	trans.	cels	rot., copy, cont	rot., cont	appear, disapp.	orient, copy	trans, rot, copy	orient	trans, rot	trans, scale, trace	orient	trans, rot	rel rot	limbs	cels	rel deform	rel morph
Scenario	67–Automobile Accident					68–Bachelor Party																69–Tack vs. Jibe						
Feature	labels		car motions		Fire station	Multiple parts	Titles, speech bal.	Character views		movement	conveyor		conveyor switch				Plane				end titles	boat movement		sail movement				
Define background																												
Import																												
Add scene		●				●																						
Copy object		●				●			●							●												
Zoom			●	●						●	●	●									●							
Repeat playback																												
Occlude					●																							
Play sound																												
Orient to path			●														●		●			●						
Move limb																									●			
Deform																											●	
Move forward/back																												
Interpolate																												
Physically simulate																												
Morph																												●
Define cels								●					●													●		
Copy motion											●			●			●	●										
Repeat motion																												
Trace																					●							
Disappear	●						●		●							●												
Appear	●						●		●							●												
Move relative																					●		●	●			●	●
Set timing: continuous														●														
Set timing: acceleration																												
Rotate: full														●	●									●				
Rotate: any center																												
Rotate: limited				●														●		●			●					
Scale: full																												
Scale: uniform																					●							
Translate: arbitrary			●	●													●	●	●	●		●	●					
Translate: straight										●	●	●									●							
Extra operations												8				4			4	4						10		

Scenario	70–Lewis, Clark, Vancouver		71–Trapeze				72–Ski Jump				
Feature	background	movements	swings		people		Skier views	launch		somersault	land
Approach	import	trace	cont.	parts	trans. curved	trans. straight	appear/disapper	orient	trans, rot	trans, rot	trans.
Define background	●										
Import											
Add scene											
Copy object											
Zoom											
Repeat playback											
Occlude											
Play sound											
Orient to path								●			
Move limb											
Deform											
Move forward/back											
Interpolate											
Physically simulate											
Morph											
Define cels											
Copy motion											
Repeat motion											
Trace		●									
Disappear							●				
Appear							●				
Move relative									●	●	
Set timing: continuous			●								
Set timing: acceleration											
Rotate: full			●	●							
Rotate: any center										●	
Rotate: limited									●		
Scale: full											
Scale: uniform											
Translate: arbitrary					●			●	●	●	●
Translate: straight						●					
Extra operations				3		8			1		

C.3 Full Interface Optimization Results

Figure 4-6 on page 96 and Table 4-1 on page 98 present subsets of the data I collected through interface optimization. This section presents all the data I collected. Most of this data was generated with the assumption that the eight additional animation operations would be provided by any animation interface. I present five tables that make this assumption and allow 0, 1, 2, 3, and 4 extra steps.

I also present two tables generated without the assumption that additional operations would be present, one allowing 0 extra steps, and one allowing 4 extra steps. Because my optimization algorithm's running time grew exponentially with the number of operations considered, these tables were harder to generate. I created a parallel version of my optimization program and ran it on 32 nodes of a network of workstations, each a Linux machine with a 2.8 GHz processor and 4 GB of RAM. Each node ran for 110–130 minutes, taking a total of 127 hours and 50 minutes of processing time to generate the two tables.

Note that the tables in this section ignore most of the variants of *translate, scale, rotate,* and *set timing.* Since these variants are all less expressive versions of another animation operation, it is not surprising that they would seldom appear in an optimal solution. (Recall that an optimal solution is one with the smallest number of animation operations that supports a given number of scenarios.) I include only those variants that appear once or more in optimal solutions.

Each cell in the body of these tables corresponds to an animation operation and a number of scenarios. (If the tables are viewed by rotating this document 90° clockwise, then a column corresponds to a set of solutions for a given number of scenarios.) Cells contain three types of values: nothing, an asterisk (*), or a list of numbers. If a cell contains nothing, then no solution in that column contains that animation operation. If a cell contains an asterisk, then all solutions in that column contain that animation operation. If a cell contains a list of numbers, then some solutions in that column contain that animation operation. Each number in a column identifies a solution, and all solutions appearing in a cell contain the given animation operation.

C.3.1 Additional Operations Assumed with 0 Extra Steps

Scenarios supported	8%	14%	21%	28%	38%	47%	53%	65%	71%	76%	81%	86%	90%	93%	96%	99%	100%
Num. of operations	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Num. of solutions	3	8	1	1	2	1	1	1	1	1	5	1	1	3	2	1	1
Translate: straight	1	1,4,6															
Translate: arbitrary	2	2,3,5,7	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Scale		1,2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Rotate				*	*	*	*	*	*	*	*	*	*	*	*	*	*
Set timing			*		1	*	*	*	*	*	*	*	*	*	*	*	*
Move relative				*	*	*	*	*	*	*	*	*	*	*	*	*	*
Appear							*	*	*	*	*	*	*	*	*	*	*
Disappear							*	*	*	*	*	*	*	*	*	*	*
Trace										*	*	*	*	*	*	*	*
Repeat motion					2	*		*	*	*	*	*	*	*	*	*	*
Copy motion											2,3	*	*	*	*	*	*
Define cels									*		4,5	*	*	*	*	*	*
Morph											3,5	*	*	*	*	*	*
Physically simulate	3	6–8												3			
Interpolate														1	2	*	*
Move forward/back															1	*	*
Deform										*	1,2,4			2	*	*	*
Move limb		4,5,8															*
Orient to path		3									1		*	*	*	*	*

C.3.2 Additional Operations Assumed with 1 Extra Step

Scenarios supported	8%	14%	21%	29%	39%	49%	57%	69%	74%	79%	85%	89%	92%	94%	97%	99%	100%
Num. of operations	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Num. of solutions	3	8	1	1	2	1	1	1	2	2	1	1	3	2	1	2	1
Translate: straight	1	1,4,6															
Translate: arbitrary	2	2,3,5,7	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Scale		1,2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Rotate				*	*	*	*	*	*	*	*	*	*	*	*	*	*
Set timing			*		1	*	*	*	*	*	*	*	*	*	*	*	*
Move relative				*	*	*	*	*	*	*	*	*	*	*	*	*	*
Appear							*	*	*	*	*	*	*	*	*	*	*
Disappear							*	*	*	*	*	*	*	*	*	*	*
Trace									1	*	*	*	*	*	*	*	*
Repeat motion					2	*		*	*	*	*	*	*	*	*	*	*
Copy motion											*	*	*	*	*	*	*
Define cels									2			*	*	*	*	*	*
Morph										2	*	*	*	*	*	*	*
Physically simulate	3	6–8											3				
Interpolate													1	2	*	*	*
Move forward/back														1	*	*	*
Deform										1			2	*	*	*	*
Move limb		4,5,8														2	*
Orient to path		3														1	*

C.3.3 Additional Operations Assumed with 2 Extra Steps

Scenarios supported	10%	15%	24%	31%	43%	49%	61%	69%	74%	79%	85%	89%	92%	94%	97%	99%	100%
Num. of operations	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Num. of solutions	2	8	1	2	1	1	1	1	2	3	1	1	3	2	1	2	1
Translate: straight	1	1,5,7															
Translate: arbitrary	2	2–4,6,8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Scale		1,2	*	1	*	*	*	*	*	*	*	*	*	*	*	*	*
Rotate				*	*	*	*	*	*	*	*	*	*	*	*	*	*
Set timing		4	*	2	*	*	*	*	*	*	*	*	*	*	*	*	*
Move relative				*	*	*	*	*	*	*	*	*	*	*	*	*	*
Appear							*	*	*	*	*	*	*	*	*	*	*
Disappear							*	*	*	*	*	*	*	*	*	*	*
Trace									1	*	*	*	*	*	*	*	*
Repeat motion						*		*	*	1,3	*	*	*	*	*	*	*
Copy motion										2	*	*	*	*	*	*	*
Define cels									2			*	*	*	*	*	*
Morph										2,3	*	*	*	*	*	*	*
Physically simulate		7,8											3				
Interpolate													1	2	*	*	*
Move forward/back														1	*	*	*
Deform										1			2	*	*	*	*
Move limb		5,6														2	*
Orient to path		3														1	*

C.3.4 Additional Operations Assumed with 3 Extra Steps

Scenarios supported	11%	17%	25%	33%	44%	50%	65%	74%	78%	82%	86%	89%	92%	94%	97%	99%	100%
Num. of operations	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Num. of solutions	1	6	1	1	1	1	1	1	1	1	1	3	5	3	1	2	1
Translate: straight																	
Translate: arbitrary	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Scale		2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Rotate		1		*	*	*	*	*	*	*	*	*	*	*	*	*	*
Set timing		4	*		*	*	*	*	*	*	*	*	*	*	*	*	*
Move relative				*	*	*	*	*	*	*	*	*	*	*	*	*	*
Appear							*	*	*	*	*	*	*	*	*	*	*
Disappear							*	*	*	*	*	*	*	*	*	*	*
Trace										*	*	*	*	*	*	*	*
Repeat motion						*		*	*	*	*	*	*	*	*	*	*
Copy motion												1,3	*	*	*	*	*
Define cels									*		*	*	*	*	*	*	*
Morph												3	3–5	2,3	*	*	*
Physically simulate		6											5				
Interpolate													2,3	1,3	*	*	*
Move forward/back												2	1	1,2	*	*	*
Deform										*	*	1,2	1,2,4	*	*	*	*
Move limb		5														2	*
Orient to path		3														1	*

C.3.5 Additional Operations Assumed with 4 Extra Steps

Scenarios supported	15%	21%	28%	36%	46%	51%	68%	75%	79%	83%	86%	90%	93%	94%	97%	99%	100%
Num. of operations	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Num. of solutions	1	4	1	1	1	1	1	1	3	3	8	1	2	11	3	5	2
Translate: straight																	
Translate: arbitrary	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Scale		2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Rotate		1		*	*	*	*	*	*	*	*	*	*	*	*	*	*
Set timing			*		*	*	*	*	*	*	*	*	*	*	*	*	*
Move relative				*	*		*	*	*	*	*	*	*	*	*	*	*
Appear						*	*	*	*	*	*	*	*	*	*	*	*
Disappear						*	*	*	*	*	*	*	*	*	*	*	*
Trace									1,2	*	*	*	*	*	*	*	*
Repeat motion								*	2,3	3	2,5–8	*	*	*	*	*	*
Copy motion									1	1,2	1–5	*	*	*	*	*	*
Define cels									3	3	5–8	*	*	*	*	*	*
Morph										2	2–47	*	*	2–11	*	*	*
Physically simulate		4									4,8		2	8–11	3	4,5	2
Interpolate													1	1,3–5, 7,11	1,2	1–3,5	*
Move forward/back											1			1,2,5, 6,10	*	*	*
Deform										1	1,3,6			1,6,7	2	2,3	1
Move limb		3												2,4,9	1,3	1,3,4,5	*
Orient to path														3,8		1,2,4	*

C.3.6 No Operations Assumed with 0 Extra Steps

Scenarios supported	3%	6%	8%	11%	14%	18%	21%	24%	26%	31%	35%	39%	44%	51%	58%	65%	71%	76%	81%	86%	90%	93%	96%	99%	100%
Num. of operations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Num. of solutions	1	1	2	2	4	2	1	1	5	5	5	8	4	1	2	1	1	1	5	1	1	3	2	1	1
Translate: straight				1	1,3	1			3																
Translate: arbitrary				2	2,4	2	*	*	1,2, 4,5	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Scale			1		1,2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Rotate										1,2	1,2, 4,5	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Set timing										1	1,4	1–3, 6,7	1–3	*	*	*	*	*	*	*	*	*	*	*	*
Move relative										1,2	1,2, 4,5	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Appear												3–6	1,3,4	*	*	*	*	*	*	*	*	*	*	*	*
Disappear												3–6	1,3,4	*	*	*	*	*	*	*	*	*	*	*	*
Trace			2		3,4													*	*	*	*	*	*	*	*
Repeat motion						*	*	*	*	*	*	1–5, 7,8	2–4	*	*	*	*	*	*	*	*	*	*	*	*
Copy motion																			2,3	*	*	*	*	*	*
Define cels							*	*	1,5	4,5	1–3	5,7,8	3,4				*		4,5	*	*	*	*	*	*
Morph																			3,5	*	*	*	*	*	*
Physically simulate	*	*	*	*	*	*	*	*	*	2–5	2,3,5	4,8	4									3			
Interpolate																						1	2	*	*
Move forward/back																							1	*	*
Deform																		*	1,2,4			2	*	*	*
Move limb									1,3,4	3,5	3														*
Orient to path									2	3,4	3								1		*	*	*	*	*
Play sound				*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Occlude								*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Repeat playback												1,6	1,2		1	*	*	*	*	*	*	*	*	*	*
Zoom									2–5	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Copy object											4,5	7,8			2	*	*	*	*	*	*	*	*	*	*
Add scene		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Import												1,2	1,2	*	*	*	*	*	*	*	*	*	*	*	*
Define background												2	2	*	*	*	*	*	*	*	*	*	*	*	*

C.3.7 No Operations Assumed with 4 Extra Steps

Scenarios supported	3%	7%	11%	15%	18%	22%	25%	28%	33%	39%	44%	51%	57%	64%	68%	75%	79%	83%	86%	90%	93%	94%	97%	99%	100%
Num. of operations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Num. of solutions	2	1	1	1	2	1	1	3	1	2	2	1	1	1	2	1	3	3	8	1	2	11	3	5	2
Translate: straight																									
Translate: arbitrary	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Scale					1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Rotate								3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Set timing										1	2	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Move relative								3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Appear								1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Disappear								1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Trace					2												1,2	*	*	*	*	*	*	*	*
Repeat motion						*	*	1,2		2	1				1	*	2,3	3	2,5–8	*	*	*	*	*	*
Copy motion																	1	1,2	1–5	*	*	*	*	*	*
Define cels							*	2									3	3	5–8	*	*	*	*	*	*
Morph																		2	2–4,7	*	*	2–11	*	*	*
Physically simulate	2		*	*	*	*	*	*											4,8		2	8–11	3	4,5	2
Interpolate																					1	1,3–5 7,11	1,2	1–3,5	*
Move forward/back																			1			1,2,5, 6,10	*	*	*
Deform																		1	1,3,6			1,6,7	2	2,3	1
Move limb																						2,4,9	1,3	1,3–5	*
Orient to path																						3,8		1,2,4	*
Play sound				*	*	*	*	1,2			1	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Occlude								2,3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Repeat playback											2	*	*	*	2	*	*	*	*	*	*	*	*	*	*
Zoom								1,3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Copy object															*	*	*	*	*	*	*	*	*	*	*
Add scene		*	*	*	*	*	*	2,3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Import													*	*	*	*	*	*	*	*	*	*	*	*	*
Define background														*	*	*	*	*	*	*	*	*	*	*	*

APPENDIX

D Formative Laboratory Evaluation

D.1 Documents Used

D.1.1 Script

Materials

2	Ball-point pens
1	Study Script
2	Informed Consent Form (2-sided)
1	Demographic Survey (2-sided)
1	Experimental Data Sheet (2-sided)
1	Task Instructions Sheets (2-sided)
3	Comfort-level questionnaires (2-sided)
1	Amazon Gift Certificate

Procedure

1) Participant fills out Consent Forms
2) Give copy of consent form
3) Participant fills out Demographics/Screening Form
4) Set up K-Sketch
5) Administer Tutorial
6) Give practice task/Offer Play Time
7) Observe/assist as user experiments, limiting to 15 minutes
8) Repeat three times (for three experimental tasks)
    a. Show participant Task Instructions
    b. Observe as task is performed, recording time and critical incidents
    c. Participant fills out Comfort-level Questionnaire

9) Participant fills out Exit Questionnaire
10) Give participant $20 Amazon.com gift certificate, if available.

D.1.2 Informed Consent

UNIVERSITY OF WASHINGTON CONSENT FORM

Investigators:

James A. Landay | Associate Professor | Computer Science & Engineering
Telephone: (206) 685-9139 | e-mail*: landay@cs.washington.edu

Richard C. Davis | Graduate Student | Computer Science & Engineering
Telephone: (206) 992-9363 | e-mail*: rcdavis@cs.washington.edu

*Please note that we cannot ensure the confidentiality of information sent via e-mail.

INVESTIGATORS' STATEMENT

We are asking you to be in a research study. The purpose of this consent form is to give you the information you will need to help you decide whether or not to be in the study. Please read the form carefully. You may ask questions about the purpose of the research, what we would ask you to do, the possible risks and benefits, your rights as a volunteer, and anything else about the research or this form that is not clear. When all your questions have been answered, you can decide if you want to be in the study or not. This process is called 'informed consent.'

PURPOSE OF THE STUDY

We want to better understand how novice animators respond to our experimental animation tool, "K-Sketch." The goal of this study is to observe users' speed and satisfaction when producing animations with K-Sketch.

PROCEDURES

If you choose to participate, we would like you to fill out some questionnaires, receive a short tutorial, and to create three animations using K-Sketch. We would like to observe you as you create the animations. The entire session will take about one hour.

The first questionnaire will briefly ask you questions about your background, such as your age and occupation. We will also ask you about your experience using animation tools. You do not have to answer every question.

Next, you will receive a verbal tutorial that explains how to use K-Sketch. Following this, you'll have 15 minutes to experiment with the tool. During this period, you will be free to ask as many questions as you like.

Next, we would like you to create three animations using K-Sketch. We will observe you as you are using each tool and measure how long it takes you. After finishing each animation, we'd like you to tell us how comfortable the tool was to use. You may opt out of any request at any time.

RISKS, STRESS, OR DISCOMFORT

Some people feel that providing information for research is an invasion of privacy.

BENEFITS OF THE STUDY

It is hoped that this research will benefit Human-Computer Interaction researchers in general and help us in particular to make K-Sketch better. You may not directly benefit from taking part in this study.

OTHER INFORMATION

Taking part in this research is voluntary. You can stop at any time. Information about you is confidential. I will not link study information with your name. No one will be able to tell which animations you create. The following groups may review study records for this project: University of California at Berkeley researchers; Institutional oversight review offices at the University of Washington, or the University of California Berkeley; and federal regulators. We may publish or present animation created during this study. If we publish or present the results of this study, we will not be able to use your name.

You will receive a $20 gift certificate to Amazon.com for your participation in this study.

____

Signature of investigator Printed Name Date

Subject's statement

This study has been explained to me. I volunteer to take part in this research. I have had a chance to ask questions. I agree to let the researcher present my computer work publicly as described above in the consent form. If I have questions later on about the research I can ask one of the investigators listed above. If I have questions about my rights as a research subject, I can call the University of Washington Human Subjects Division at (206) 543-0098. I will receive a copy of this consent form.

____

Signature of subject Printed name Date

Copies to: Investigator's file
Subject

D.1.3 Demographic Survey

Participant # ________ Age _________ Occupation ____________________

Sex ________ (Students Only) Ugrad/Grad, Year, Field of Study ________________________________

1) How would you rate your drawing skills? (Circle one)

none poor fair good excellent

2) How would you rate your computer programming skills? (Circle one)

none poor fair good excellent

3) When did you last create an animation? (Circle one)

less than 24 hours ago	less than one week ago	less than one month ago	less than six months ago	more than six months ago

4) What tools do you use to create animation? (Circle all that apply)

Physical Media	Macromedia Flash	Adobe AfterEffects	Microsoft PowerPoint	Other Software (Specify)

5) How often do you want to create animations?

6) Please describe any animations you have created or would like to create.

7) What do you normally do when you want to create an animation?

D.1.4 K-Sketch Tutorial

1. Familiarize yourself with K-Sketch. In the center is a canvas where you will draw your animations and view them. At the bottom, there are controls for starting or stopping your animations and navigating through time. At the top, there are buttons for managing files, undoing and redoing operations, and choosing pen styles.
2. K-Sketch has two modes, Going and Stopped. Time will only advance when you tell K-Sketch to "Go." If you draw or edit things when K-Sketch is "Going", then everything you do will be recorded in real time.
3. Try it now. Press "Go" to make the canvas Go and draw a line slowly (count to 5). Now press "Stop" and draw another line. Now drag the slider back to the beginning and press "Go" again. Notice how the first line appears over time just as if you were drawing it, but the other appears instantly? Also, did you notice how the animation stopped when it reached the end? When you "Go" from a

moment in time before the end of the animation, K-Sketch will automatically stop when it reaches the end.

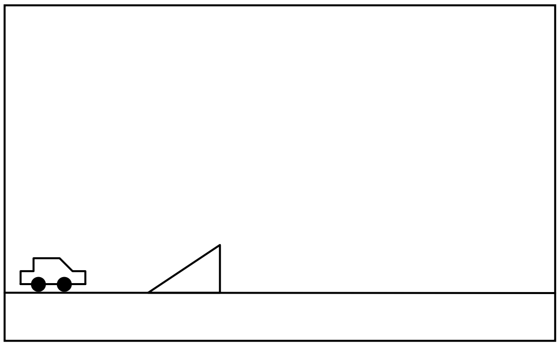
4. Take a closer look at the animation controls. Notice that tick marks appeared over the slider at the moments when you started or stopped drawing lines? These tick marks appear at any moment in time when you do something. Drag the slider bar and notice that it snaps to these events. Also notice that by tapping on either side of the slider thumb, you'll skip to the next event. The other buttons are for Going to the beginning or end of the animation or for nudging time forward or back.
5. We can record new things as the animation is going. Go back to the beginning (by pressing the button) and press "Go" again. Now draw a line quickly (count to 2). Did you notice how your first line was appearing as you drew your new line? Everything happens at once when you go.
6. Now, did you notice how there was a pause at the beginning of the animation before your lines start to appear? (Play again if you didn't.) Why do you suppose that's there? (Because of time between "Go" and draw.) If you don't want that delay, you can tell K-Sketch to start Going only when you put your pen down. Press "Go" and hold your pen down for a second. See the button change? Now draw a third line, and play your animation. Notice how it started drawing immediately?
7. Now let's look at how to select objects. Go back to the beginning and draw a triangle using three distinct lines. Now hold the "control" button and draw a loop around your triangle. Notice some strokes turn into outlines? That means they are selected. You will select a stroke if more than half of it lies within your selection loop. You can also select things by tapping on them.
8. Let's look at the box that appears over your selected object. You can modify your selected object by dragging in these different regions. Try it out now.
    a. Translate
    b. Rotate
    c. Drag

   d. Scale
   e. Non-uniform scale
   f. Move center – moves the box relative to the thing you're controlling.
9. Now press "Go" and move your triangle. Notice how, when you play it, the motion was recorded? Did you also notice that delay at the beginning? You can press and hold the "Go" button like you did earlier to get rid of this delay, but there is another short-cut. Draw a square, select it, and hold the "control" button when you move it. Notice how K-Sketch was going while you dragged? This is a handy shortcut to remember.
10. With the triangle selected, take a look at the timeline. Notice how the tick-marks look different? When an object is selected, you see a summary of all the changes you made to that object over time. Changes over time appear bracketed by half circles, and instant changes are squares. Now select just one side of your triangle. Notice how the timeline feedback is grayed-out? That's because you've selected only part of an object that was moved.
11. Grab the eraser and erase your triangle. The triangle is now gone at all future moments in time. Notice how it appears faded? That's a reminder of where it was. It will only look like that at this moment in time. At future points in time, it will be gone altogether.
12. Click undo to undo your erasure. Undo and Redo are very handy ways to back out of things you didn't mean to do.
13. Finally, touch the "Show Motion Paths" button. Notice the light blue lines that appear on the screen? In this mode, you can see the path of every movement you demonstrate. This can sometimes help you to coordinate things. You can also select and change these, but strange things can happen, so I suggest you leave this mode off.

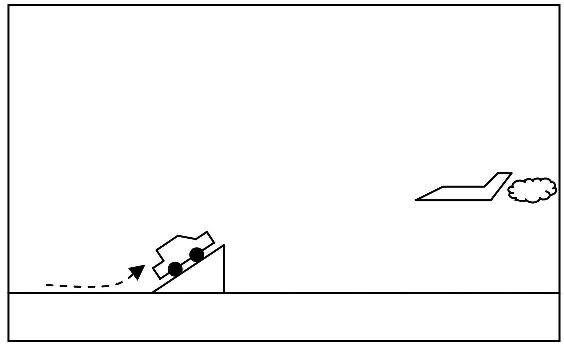
D.1.5 Task Instructions

Practice Task Instructions

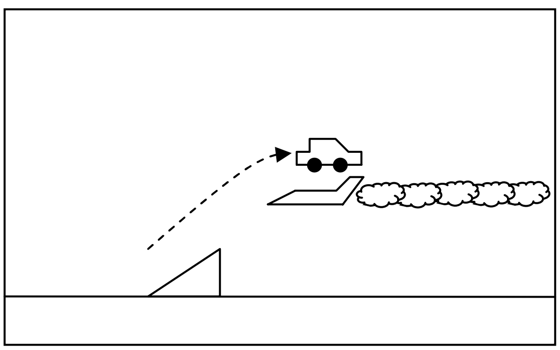
In this task, you will create an animation of a car jumping over an airplane and crashing.



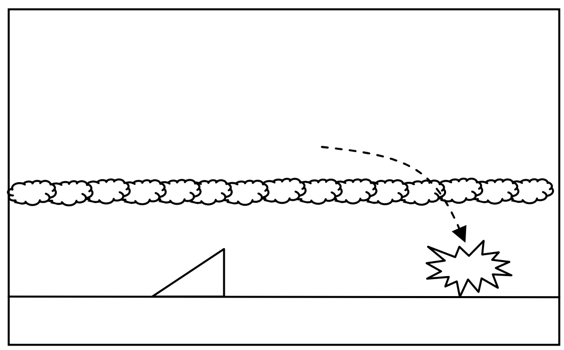
The action begins as above with a car on the left preparing to drive over a ramp.



As the car is moving, a plane starts speeding from right to left, leaving a trail of smoke. (The arrow indicating the car's path should not be shown.)



Just as the car hits its highest point, the plane passes beneath it.



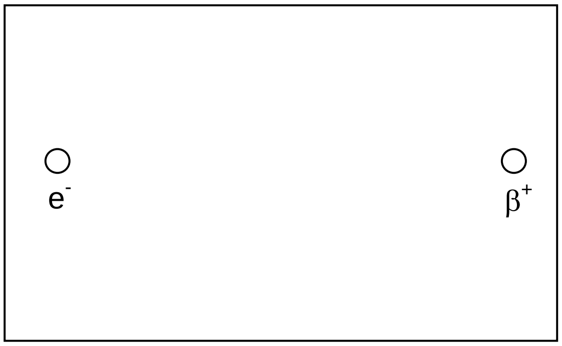
The plane speeds away, and the car crashes when it hits the ground.

Your task is to represent the important features of this animation as fast as you can. You do not need to make the objects or their motions look perfect, but keep the sequence of events the same.

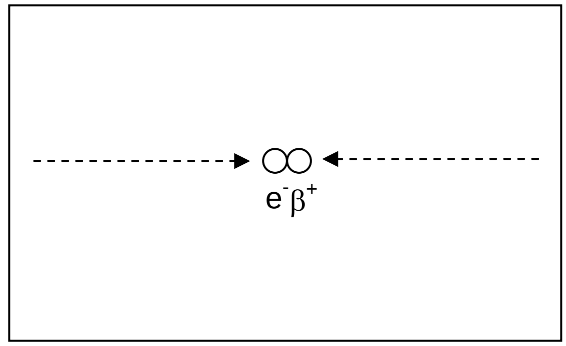
Take time to familiarize yourself with the tool you are using as you complete this task. You may take as much time as you like to experiment with the tool after the task is completed up to a maximum of 30 minutes. As you practice, the experimenter may ask you to verbalize what you are thinking. Also, feel free to ask the experimenter for help or advice as you work.

Task A Instructions

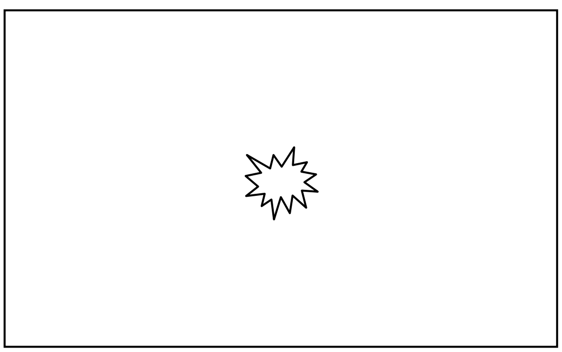
In this task, you will create an animation of particles colliding and creating gamma rays.



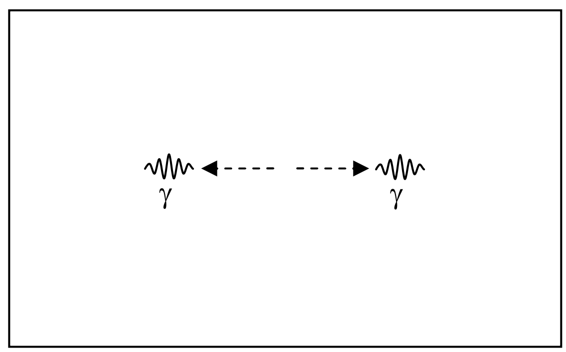
The action begins with an electron (e^-) on the left and a positron (β^+) on the right.



The particles move toward each other at constant speed until they touch in the center. (The arrows indicating the particles' paths should not be shown.)



There is a very brief explosion…



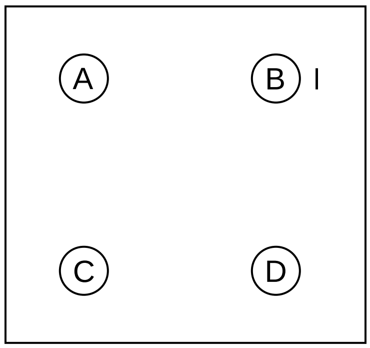
…and two gamma rays move away from the explosion at the same speed as the particles. The gamma rays move until they are off screen.

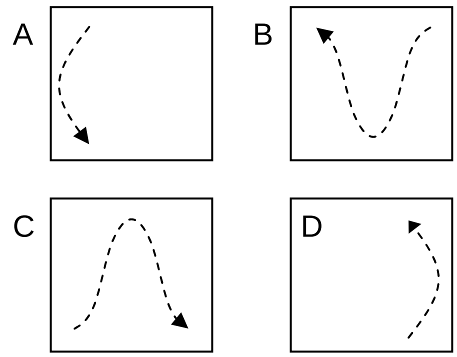
Your task is to represent the important features of this animation as fast as you can. You do not need to make the objects or their motions look perfect, but keep the sequence of events the same.

As you work, the experimenter may ask you to verbalize what you are thinking. Avoid asking the experimenter for help or advice, but feel free to do so if you get stuck.

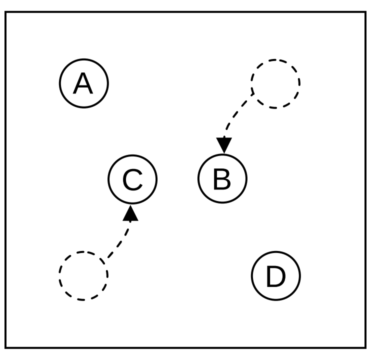
Task B Instructions

In this task, you will create an animation of a dance maneuver.

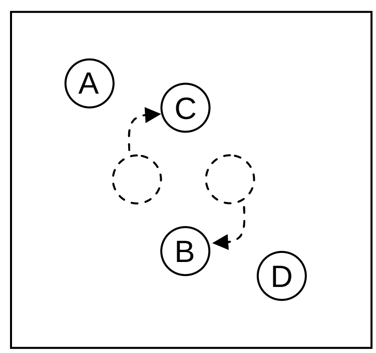




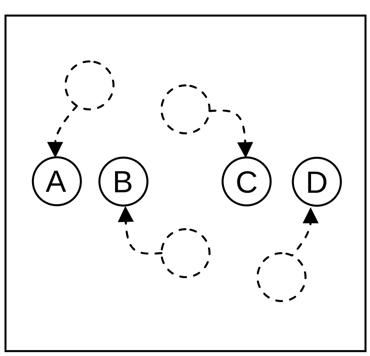
The action begins with four dancers arranged as shown above left. Dancers B and C move throughout the maneuver, but dancers A and D move only during the second half. The path each dancer follows is shown in the diagrams above right. The diagrams below separate this dance maneuver into four stages and show the progression. (The dashed circles show old positions and the dashed arrows show movement. Your animation should not show these dashed circles and arrows.)



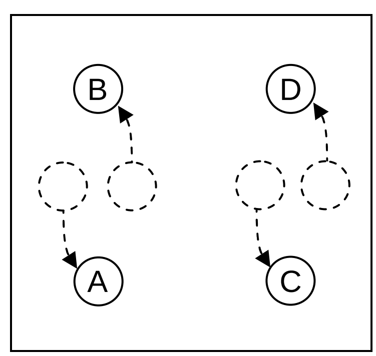
First stage



Second stage



Third stage



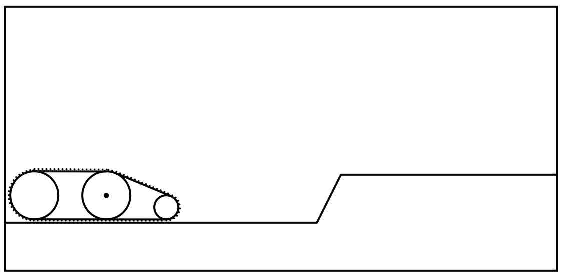
Fourth stage

Your task is to represent the important features of this animation as fast as you can. You do not need to make the objects or their motions look perfect, but keep the sequence of events the same.

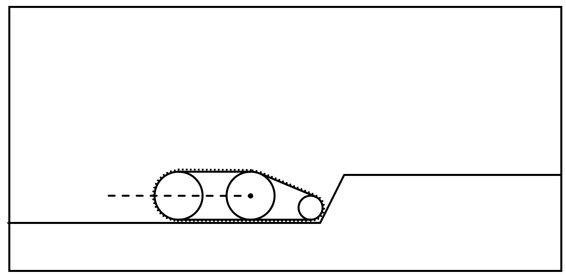
As you work, the experimenter may ask you to verbalize what you are thinking. Avoid asking the experimenter for help or advice, but feel free to do so if you get stuck.

Task C Instructions

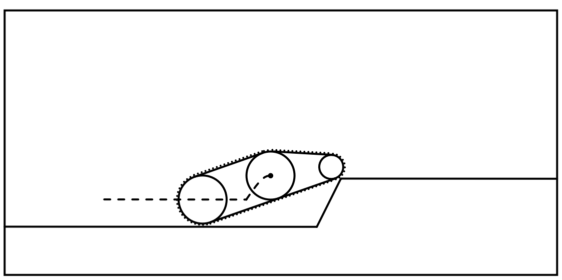
In this task, you will create an animation of a machine tread rolling over a bump.



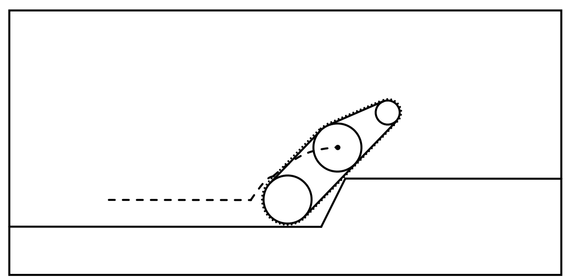
The action begins with a machine tread to the left of a bump in the road.



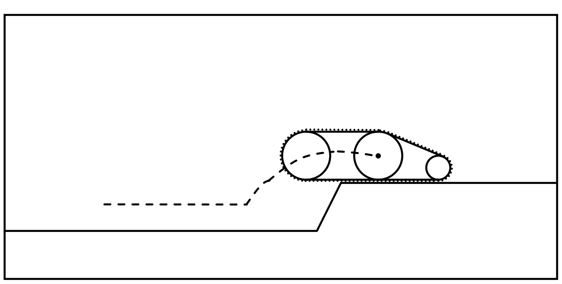
As the tread moves, the dot in the center wheel traces out a visible path.



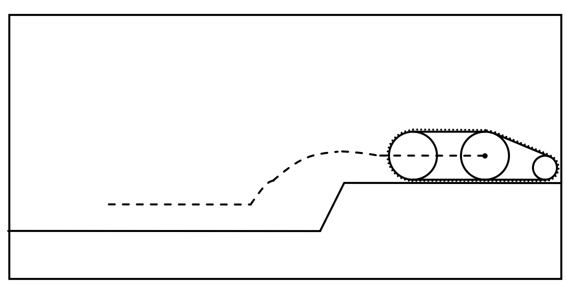
The tread starts to move over the bump, and the path trace moves up sharply.



After the tread reaches this point, it begins to swing down…



…to a horizontal orientation as shown here.



The tread stops at the far right so that the bump in the path trace is visible.

Your task is to represent the important features of this animation as fast as you can. You do not need to make the objects or their motions look perfect, but keep the sequence of events the same.

As you work, the experimenter may ask you to verbalize what you are thinking. Avoid asking the experimenter for help or advice, but feel free to do so if you get stuck.

D.1.6 Comfort-level Questionnaire

Please let us know how comfortable you would be ***showing*** the animation you just created to the following groups of people. Circle a response to each of the following.

	Extremely Uncomfortable			Neutral			Extremely Comfortable	Don't know
1) To myself (keep in private notebook only)	1	2	3	4	5	6	7	?
2) To a single colleague in a private meeting	1	2	3	4	5	6	7	?
3) To ten colleagues in a private meeting	1	2	3	4	5	6	7	?
4) To a student I am tutoring	1	2	3	4	5	6	7	?
5) To 30 students in a class I am teaching	1	2	3	4	5	6	7	?
6) To 300 students in a class I am teaching	1	2	3	4	5	6	7	?
7) To 30 professionals watching a business presentation I'm making	1	2	3	4	5	6	7	?
8) To 300 professionals watching a business presentation I'm making	1	2	3	4	5	6	7	?

Please let us know how comfortable you would be ***creating*** the animation as you just created it ***while being watched by*** the following groups of people. Circle a response to each of the following.

	Extremely Uncomfortable			Neutral			Extremely Comfortable	Don't know
9) Watched by no one	1	2	3	4	5	6	7	?
10) Watched by a single colleague in a private meeting	1	2	3	4	5	6	7	?
11) Watched by ten colleagues in a private meeting	1	2	3	4	5	6	7	?
12) Watched by a student I am tutoring	1	2	3	4	5	6	7	?
13) Watched by 30 students in a class I am teaching	1	2	3	4	5	6	7	?
14) Watched by 300 students in a class I am teaching	1	2	3	4	5	6	7	?
15) Watched by 30 professionals during a business presentation I'm making	1	2	3	4	5	6	7	?
16) Watched by 300 professionals during a business presentation I'm making	1	2	3	4	5	6	7	?

D.1.7 Exit Questionnaire

	Not at all		Neutral		Very Much	Don't know
1) How did you enjoy the experience of using K-Sketch?	1	2	3	4	5	?
2) How likely would you be to use K-Sketch if the software and hardware were readily available to you? (Write any conditions below)	1	2	3	4	5	?

3) Do you have any comments to make about K-Sketch or how you might use it?

D.2 Data Collected

The following tables list most of the data collected in this study (apart from participants' animations). Since participants with little or no interest in animation were analyzed separately from others, these participants are italicized in the tables.

D.2.1 Demographic Data

#	Occupation	Age	Sex	Animation interest	Programming skill
1	undergrad. student, molecular biology	20	F	medium	fair
2	grad. student, education	25	F	medium	none
3	*administrative*	*45*	*F*	*none*	*none*
4	post doc., biology	29	M	medium	none
5	undergrad student, marketing and business	21	F	medium	poor
6	*administrative*	*>18*	*F*	*none*	*fair*
7	*administrative*	*31*	*F*	*low*	*poor*
8	*administrative*	*>18*	*F*	*none*	*none*
9	*administrative*	*40*	*M*	*none*	*fair*
10	grad. student, educational psychology	33	F	medium	none
11	grad. student, education and cognitive studies	33	F	medium	none

Main demographics information

#	Drawing skills	Anim. Experience	Experience Comments	Anim Uses
1	fair	some PPT	Like to learn anim., but it would take too long. Has someone else do for her.	presentations, wishes teachers would use in chemistry and biology classes
2	poor	some PPT	Want to but don't know how. Seen people do. Asks someone else to do.	
3	*poor*	*some PPT*	*Ask for help*	*cheerleading presentations, web sites, e-mails*
4	fair	more PPT	Did once over last year, but once a week during class he taught	biological processes (hormone / receptor interactions), maps of species distributions, animations of testing procedures
5	far	none		
6	*poor*	*none*	*Ask for help*	*cartoons for friends, project reports*
7	*poor*	*none*		
8	*fair*	*none*		
9	*fair*	*some PPT*		
10	poor	none		K-12 students learning languages (animate story and explain in foreign language
11	poor	none		demonstrating scientific models, installation in children's museum

Extended demographics information

D.2.2 Experimental Data

#	Motion path visibility	Tutorial time (min)	Practice time (min)	Total learning time (min)	Task order	Interventions
1	on	19	4	23	ABC	
2	on	19		19	BAC	(Task C) forgot how to move handle
3	*on*	*21*	*6*	*27*	*CAB*	*(Task A) (2x) make motions simultaneous* *(Task C) how to draw trail*
4	on	17		17	ABC	
5	off	16	13	29	ABC	
6	*off*	*18*		*18*	*ABC*	*(Task C) how to draw trail*
7	*off*	*13*	*9*	*22*	*ABC*	*(Task C) point out accidentally erased motion*
8	*off*	*16*	*17*	*33*	*ABC*	*(Task A) general problems with animation*
9	*off*	*18*		*18*	*ABC*	*(Task B) widget record shortcut (alt. button)* *(Task C) how to draw trail (record drawings)*
10	on	17	7	24	ABC	
11	on	16	13	29	ABC	
Med. interest avg.		17.3		23.5		
Low interest avg.		*17.2*		*22.3*		

Practice and miscellaneous task data

Number	Audience
1	No one[a]
2	A single colleague in a private meeting
3	Ten colleagues in a private meeting
4	A student I am tutoring
5	30 students in a class I am teaching
6	300 students in a class I am teaching
7	30 professionals during a business presentation I'm making
8	300 professionals during a business presentation I'm making

[a] For comfort showing, this was listed as "myself (keep in private notebook only)."

Audiences for comfort-level questionnaire

#	Task A time (min)	Task A average comfort showing	Comfort showing by audience (1 = extremely uncomfortable, 7 = extremely comfortable)								Task A average comfort creating	Comfort creating by audience (1 = extremely uncomfortable, 7 = extremely comfortable)							
			A1	A2	A3	A4	A5	A6	A7	A8		A1	A2	A3	A4	A5	A6	A7	A8
1	5	6.13	7	7	6	7	7	7	4	4	4.63	7	5	4	7	3	3	4	4
2	4	5.00	7	7	6	6	4	4	3	3	4.25	6	6	5	5	3	3	3	3
3	10	3.38	5	5	4	3	3	2	3	2	4.00	5	5	4	4	4	3	4	3
4	4	5.00	7	7	6	7	5	4	2	2	7.00	7	7	7	7	7	7	7	7
5	4	3.38	7	6	4	4	2	2	1	1	3.38	5	5	4	4	4	3	1	1
6	7[a]	2.75	4	4	3	4	3	2	1	1	3.75	5	5	5	5	4	3	2	1
7	4	5.00	7	7	6	6	5	4	3	2	4.88	7	7	6	6	5	3	3	2
8	12	4.88	6	7	6	6	6	5	2	1	4.25	7	7	6	4	4	3	2	1
9	6	6.38	7	7	7	7	7	7	5	4	7.00	7	7	7	7	7	7	7	7
10	5	7.00	7	7	7	7	7	7	7	7	5.25	7	7	6	7	6	3	3	3
11	5	5.50	7	7	7	7	6	5	3	2	5.25	7	7	6	7	6	3	3	3
MA	4.5	4.9									5.3								
LA	6.7	5.3									5.0								

[a] Participant was unable to complete this task according to instructions.

Task A data. MA = Medium interest average, LA = Low interest average. Low interest average omits participants 6 and 8, because they did not complete all tasks.

#	Task B time (min)	Task A average comfort showing	Comfort showing by audience (1 = extremely uncomfortable, 7 = extremely comfortable)								Task B average comfort creating	Comfort creating by audience (1 = extremely uncomfortable, 7 = extremely comfortable)							
			A1	A2	A3	A4	A5	A6	A7	A8		A1	A2	A3	A4	A5	A6	A7	A8
1	5	6.13	7	7	6	7	6	6	5	5	5.00	7	6	5	6	4	4	4	4
2	6	4.38	7	7	6	5	3	3	2	2	3.88	6	6	5	5	3	2	2	2
3	9	3.75	5	5	4	4	3	3	3	3	4.00	5	5	4	5	4	3	3	3
4	7	5.63	7	7	7	7	6	5	3	3	3.63	7	7	3	6	2	1	2	1
5	4	4.00	7	6	5	5	4	3	1	1	3.38	7	6	5	3	2	2	1	1
6	6[a]	5.25	7	7	6	6	5	4	4	3	5.00	7	7	6	6	5	4	3	2
7	3	5.25	7	7	6	6	5	5	3	3	5.13	7	7	6	6	5	4	3	3
8																			
9	8	6.25	7	7	7	7	7	6	5	4	6.63	7	7	7	7	7	7	6	5
10	6[b]	4.63	7	7	5	7	5	2	2	2	5.13	7	7	5	7	6	3	3	3
11	5	5.75	7	7	7	7	7	5	3	3	7.00	7	7	7	7	7	7	7	7
MA	5.5	5.1									5.3								
LA	6.7	5.1									4.7								

[a] Participant was unable to complete this task according to instructions.
[b] Participant was confused by instructions and did not complete this task correctly.

Task B Data. MA = Medium interest average, LA = Low interest average. Low interest average omits participants 6 and 8, because they did not complete all tasks.

#	Task C time (min)	Task C average comfort showing	Comfort showing by audience (1 = extremely uncomfortable, 7 = extremely comfortable) A1	A2	A3	A4	A5	A6	A7	A8	Task C average comfort creating	Comfort creating by audience (1 = extremely uncomfortable, 7 = extremely comfortable) A1	A2	A3	A4	A5	A6	A7	A8
1	6	6.25	7	7	6	7	6	6	6	5	5.38	7	7	6	7	4	4	4	4
2	7	3.50	7	6	3	3	3	2	2	2	3.00	7	6	3	3	2	1	1	1
3	*11*	*2.88*	*5*	*4*	*4*	*2*	*2*	*2*	*2*	*2*	*5.13*	*7*	*7*	*7*	*4*	*4*	*4*	*4*	*4*
4	4	3.75	6	5	4	6	3	2	2	2	4.25	7	6	5	7	3	2	2	2
5	6	3.50	7	6	5	3	3	2	1	1	3.50	7	6	5	3	3	2	1	1
6																			
7	*11*	*4.75*	*7*	*6*	*5*	*6*	*5*	*3*	*3*	*3*	*4.50*	*7*	*6*	*5*	*6*	*5*	*3*	*2*	*2*
8																			
9	*9*	*7.00*	*7*	*7*	*7*	*7*	*7*	*7*	*7*	*7*	*6.63*	*7*	*7*	*7*	*7*	*7*	*7*	*6*	*5*
10	6[a]	4.75	7	7	6	7	5	2	2	2	5.13	7	7	6	7	5	3	3	3
11	9[a]	2.75	5	3	3	3	2	2	2	2	3.38	6	5	3	5	3	2	2	1
MA	6.3	4.9									5.4								
LA	*10.3*	*4.1*									*4.1*								

[a] Participant was unable to complete this task according to instructions.

Task C Data. MA = Medium interest average, LA = Low interest average. Low interest average omits participants 6 and 8, because they did not complete all tasks.

D.2.3 Incidents

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8	Participant 9	Participant 10	Participant 11
Behavior observed											
editing previous points in time	1				1						
restarting motion sequence			1	1			1	1		1	3
left additional events at end	1										
animation not completed to specification						3				2	
Confusion observed											
Confusion: task	1		1							1	
Confusion: How to make animation go	1		1			1			1		1
Confusion: How to de-select					1					1	
Confusion: How hide objects			1								
Confusion: Deleting objects for all time					1				1		1
Confusion: Unexpected Timeline Feedback			1	1							
Confusion: Simultaneous motions Seq.			1	1		1		1			
Confusion: Do at wrong time					1					1	2

Behavior and confusion observed during this study (with number of occurrences)

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8	Participant 9	Participant 10	Participant 11
Problems observed											
Problems Selecting	2		1	1		1		1	1		1
Accidental Overwriting of later move	1										1
Accidental Undo of Move							1			1	
Accidentally selected wrong strokes			1								
Accidentally Selected Motion path	1	1	1							1	
Accidentally Going	2	1	1		1	1				1	1
Accidentally Stopped			1								
Accidentally moved handle	2		2							3	
Accidental Motion Type			1	1							3
Accidental Zero duration object				1			1				
Accidentally removed object from moving group										1	
Feature requests											
opaque objects	●										
partial stroke erasure		●									
import/draw vector shapes		●				●					
show time on timebar		●									
ability to slow time		●					●				
ability to erase extra time		●								●	
erase motion path (whole or portion)		●		●							●
overlay on video or image				●							
graphical cut/copy/paste									●		
move objects through time										●	
edit tic marks											●

Problems observed (with number of occurrences) and features requested during study

D.2.4 Other Data

Participant	Satisfaction (1 = not at all, 5 = very much)	Likely to use again (1 = not at all, 5 = very much)
1	5	5
2	4	2[a]
3	*5*	*5*
4	5	5
5	4	3
6	*4*	*5*
7	*4*	*4*
8	*5*	*5*[b]
9	*5*	*4*
10	*5*	*4*
11	5	4
Med interest avg.	4.7	3.8
Low interest avg.	*4.6*	*4.6*

[a] If import capabilities were included, this participant would have entered 4.
[b] If part of job only.

Exit questionnaire data

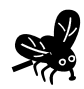
APPENDIX

E Laboratory Evaluation 1 Documents

This appendix contains most of the documents used in my first summative laboratory experiment. The only document missing is the comfort-level questionnaire, which was given to participants immediately after they completed each task (see the study script in Section E.2). The comfort-level questionnaire used in this study is identical to the one used in the formative laboratory study, and it can be found in Appendix D.

E.1 Recruiting poster

Interested in Animation?



If you are interested in creating animation but have never done so, you could participate in a three-hour ***research study*** comparing Macromedia Flash and Microsoft PowerPoint to a new animation tool called "K-Sketch."

*Participants will receive a **$50** gift certificate to Amazon.com for their time.*

This research study is being conducted by the University of California and the University of Washington on the University of Washington campus. Its purpose is to compare the animators' speed, mental workload, and satisfaction with animations produced in Flash, PowerPoint, and K-Sketch. To participate, you must be age 18 or older and must be interested in creating animation but have no experience doing so. The study will last approximately three hours, with most time spent creating animation with the tools above and some time spent answering questionnaires. Participants will receive a $50 gift certificate to Amazon.com. Some participants will be invited to take part in two additional three-hour sessions similar to the first session and will receive an additional $50 Amazon.com gift certificate for each additional session. Participation in this study is completely voluntary. Please note that we cannot ensure the confidentiality of information you send by e-mail.

To take part, or for further questions contact:

Richard Davis
(206)992-9363
rcdavis@cs.washington.edu

E.2 Study Script

Materials

2	Ball-point pens
1	Study Script
2	Informed Consent Form
1	Demographic Survey
2	Experimental Data Sheet (2-sided)
1	Task Instructions Sheets (2-sided)
1	Flash Practice Task Tutorial (2-sided)
1	PowerPoint Practice Task Tutorial (2-sided)
1	K-Sketch Practice Task Tutorial (2-sided)
6	Comfort-level questionnaires (2-sided)
2	Cognitive Load Worksheets
1	Cognitive Load Definitions
1	Cognitive Load Weights
1	Videotaping consent form

Procedure

11) Participant fills out Consent Form
12) Give copy of consent form
13) Read the following:

I would like to videotape you using these animation tools. Video taping is voluntary and will not affect the experiment. Only the research team will have access to the videotapes; you are welcome to view the video tapes if you wish. Videos will be destroyed on or before December 31, 2008. I will ask for your permission, on a separate form, to use video taped material in any publication.

Some people feel a little self-conscious when they are video taped. You may ask me to stop video taping at any time. Also, you may review the video and asked for it to be deleted.

If you agree to have the session(s) videotaped, then I will link study information to your name using a study code so you can withdraw the tape at any time. I will keep the link between your name and the study information until December 31, 2008. Then I will destroy the link. If you do not want your session videotaped, then I will not link study information to your name. If the results of this study are published or presented, I will not use your name.

Are you willing to be videotaped?

14) Participant fills out Demographics/Screening Form
15) Start Video (mark Recording Start Time)
16) Repeat twice (Once for K-Sketch, and once for PowerPoint or Flash)
    a. Set up animation tool (Check that PPT has Advanced Timeline Hidden)
    b. Ask participant to complete practice task (ask as many questions as necessary).
    c. Observe/assist as task is performed, limiting to 30 minutes
    d. Give participant task instructions
    e. Repeat two times (for two experimental tasks)
        i. Show participant task animation
        ii. Read task instructions aloud
        iii. Observe as task is performed, recording time and critical incidents
        iv. Participant fills out Comfort-level Questionnaire
    f. Participant receives Cognitive Load Definitions and fills out Cognitive Load Worksheet
    g. Participant takes 10 minute break at the end of block #1
17) Participant fills out Cognitive Load Weights form
18) Videotaping consent form
19) Give copy of video consent form
20) Tell participant that gift certificate will be sent via e-mail.

E.3 Informed Consent

UNIVERSITY OF WASHINGTON CONSENT FORM

Animation Tool Comparison Study

Investigators:

James A. Landay — Associate Professor — Computer Science & Engineering
Telephone: (206) 685-9139 — e-mail*: landay@cs.washington.edu

Richard C. Davis — Graduate Student — Computer Science & Engineering
Telephone: (206) 992-9363 — e-mail*: rcdavis@cs.washington.edu

**Please note that we cannot ensure the confidentiality of information sent via e-mail.*

INVESTIGATORS' STATEMENT

We are asking you to be in a research study. The purpose of this consent form is to give you the information you will need to help you decide whether or not to be in the study. Please read the form carefully. You may ask questions about the purpose of the research, what we would ask you to do, the possible risks and benefits, your rights as a volunteer, and anything else about the research or this form that is not clear. When all your questions have been answered, you can decide if you want to be in the study or not. This process is called 'informed consent.'

PURPOSE OF THE STUDY

We want to better understand how our experimental animation tool, "K-Sketch," compares with existing animation tools, such as Macromedia Flash and Microsoft PowerPoint. The goal of this study is to compare animators' speed, mental workload, and satisfaction when producing animations with these tools.

PROCEDURES

If you choose to participate, we would like you to fill out some questionnaires, and to create six animations using two animation software programs. We would like to observe you as you create the animations. The entire session will take about 3 hours. You will be allowed to take breaks so you don't become too tired.

The first questionnaire will briefly ask you questions about your background, such as your age and occupation. We will also ask you about your drawing skill and your interest in animation. You do not have to answer every question.

Next, we would like you to create three animations using K-Sketch, and three animations using either Microsoft PowerPoint or Macromedia Flash. We'll train you to use each tool before you do the three animation tasks. We'll time these tasks, and observe you as you perform them. After using each tool, we'd like you to tell us how comfortable the tool was to use, as well as how demanding the tool was to use. You will get a 10 minute break between using each tool. You may opt out of any request at any time.

Finally, we will ask you to compare the animation programs.

You may be invited to take part in two additional study sessions like this one at other times. In these sessions, you would repeat all the procedures described above, except for signing this form and answering initial questions. The animation tasks you perform, however, would be different from those in the first session, and you might perform more tasks or receive less training, but the total time you spend in each session would be roughly three hours

RISKS, STRESS, OR DISCOMFORT

Some people feel that providing information for research is an invasion of privacy.

BENEFITS OF THE STUDY

It is hoped that this research will benefit Human-Computer Interaction researchers in general and help us in particular to make K-Sketch better. You may not directly benefit from taking part in this study.

OTHER INFORMATION

Taking part in this research is voluntary. You can stop at any time. Information about you is confidential. I will link study information with your name only if you participate in multiple sessions. I will store this link in a locked filing cabinet separate from the data and destroy it after your last session. No one will be able to tell which animation is yours. The following groups may review study records for this project: University of California at Berkeley researchers; Institutional oversight review offices at the University of Washington, or the University of California Berkeley; and federal regulators. We may publish or present animation created during this study. If we publish or present the results of this study, we will not be able to use your name.
You will receive $50 worth of Amazon.com gift certificates for each session of this study that you participate in.

Signature of investigator Printed Name Date

Subject's statement

This study has been explained to me. I volunteer to take part in this research. I have had a chance to ask questions. I agree to let the researcher present my computer work publicly as described above in the consent form. If I have questions later on about the research I can ask one of the investigators listed above. If I have questions about my rights as a research subject, I can call the University of Washington Human Subjects Division at (206) 543-0098. I will receive a copy of this consent form.

Signature of subject Printed name Date

Copies to: Investigator's file
Subject

E.4 Demographic Survey

Participant # ____________ Age 18 or older? (Circle one): Yes / No Sex _____

Occupation ____________ (Students Only)
Ugrad/Grad, Year,
Field of Study ________________________________

1) How would you rate your drawing skills? (Circle one)

none poor fair good excellent

2) When did you last create an animation? (Circle one)

less than 24 hours ago	less than one week ago	less than one month ago	less than six months ago	more than six months ago

3) How If you had the time and skill, how often would you create animations? (Circle one)

at least once a day	at least once a week	at least once a month	at least once a year	not sure

4) What animation tools have you wanted to use? (Circle all that apply)

physical media (paper, clay)	Macromedia Flash	Adobe AfterEffects	Microsoft PowerPoint	Other Software (Specify)

5) Briefly describe the subject matter and/or purpose of the animations you would like to create.

How would you rate your expertise with the following tools? (Circle one for each)	Complete Beginner			Intermediate			Expert	Don't Know
6) Computers in general	1	2	3	4	5	6	7	?
7) Microsoft PowerPoint	1	2	3	4	5	6	7	?
8) Tablet PCs	1	2	3	4	5	6	7	?

9) Please estimate the number of hours experience you have with the following tools or any tool that resembles an animation tool. (Use the back of this sheet if necessary.)

Microsoft PowerPoint (Custom Animation features only): ____________________
Microsoft PowerPoint (any other animation features): ____________________
Other (specify)______________________________________: ____________________
Other (specify)______________________________________: ____________________
Other (specify)______________________________________: ____________________
Other (specify)______________________________________: ____________________

Please indicate your level of agreement or disagreement with the following statements. (Circle one for each)	Disagree Strongly			Neutral			Agree Strongly	Don't Know
10) I can get access to the tools I would need to make animations.	1	2	3	4	5	6	7	?
11) The complexity of animation tools discourages me from making animations.	1	2	3	4	5	6	7	?
12) The time required to make an animation with existing tools discourages me from making animations.	1	2	3	4	5	6	7	?

E.5 Task Instructions

E.5.1 Practice Task Instructions

You will begin by completing a practice task and working through a tutorial which will teach you how to use an animation tool. In this tutorial, you will be shown an idealized animation and shown how to make a similar animation using the tool. Try to work through the tutorial as fast as you can, but make sure you understand each step. You will be allowed to keep this tutorial as a reference as you complete the remaining tasks in this section of the study.

As you work, please "think aloud," that is, verbalize what you are thinking. Feel free to ask questions if there is anything that you do not understand.

E.5.2 Experimental Task Instructions

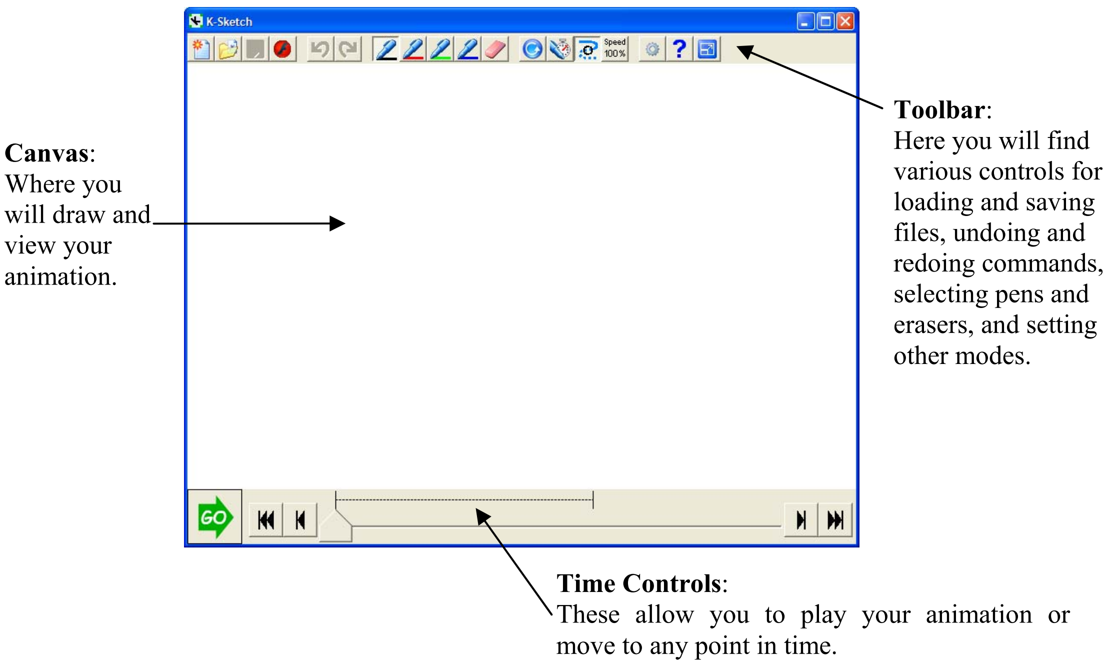
In the following task, you will be shown an idealized animation and asked to make a similar animation using a particular tool. Try to represent the important features of this animation as fast as you can. You do not need to make this animation look perfect. It is more important for you to work quickly than it is for you reproduce the objects or their motions precisely. However, please try to keep the sequence of events the same.

As you work, please "think aloud," that is, verbalize what you are thinking. Avoid asking the experimenter for help or advice, but please do so if you get stuck.

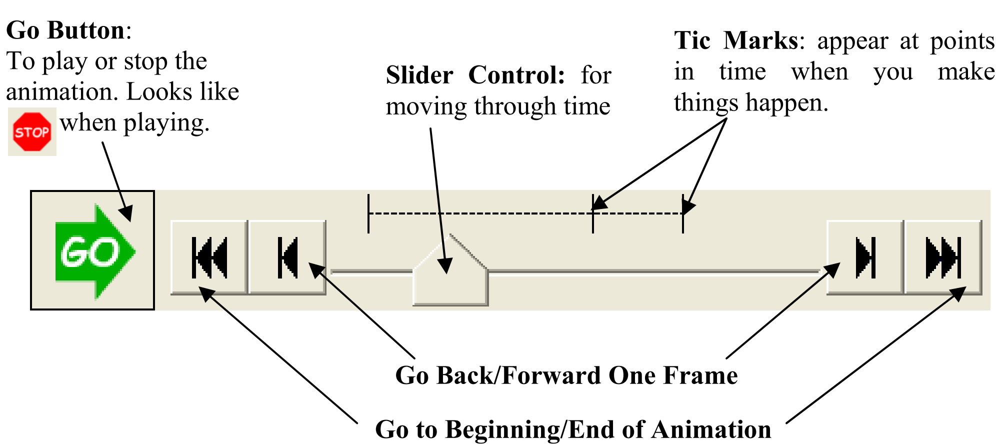
E.6 K-Sketch Practice Task Tutorial

Before we begin the practice task, we provide the following figures as a reference. We suggest you study them now and refer to them often in the future.

K-Sketch Overview



Time Controls

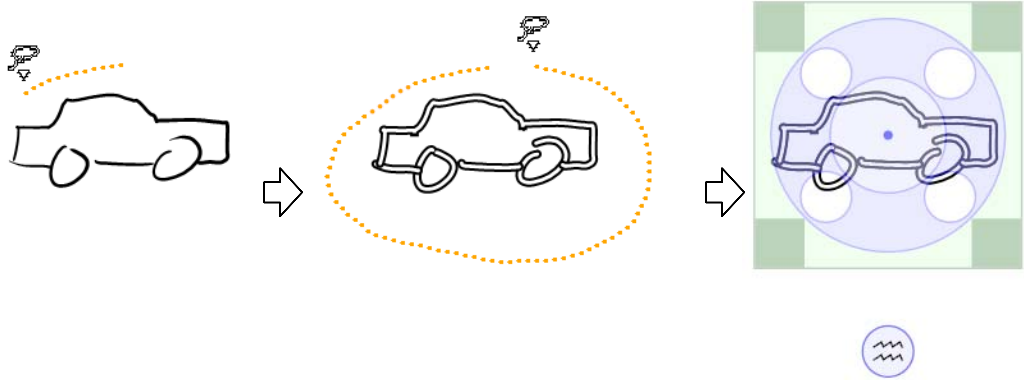


Selecting Objects

To select objects, you must hold down an **Alternate** button and draw a loop around the objects. You have several **Alternate** buttons available to you, shown in red circles below.

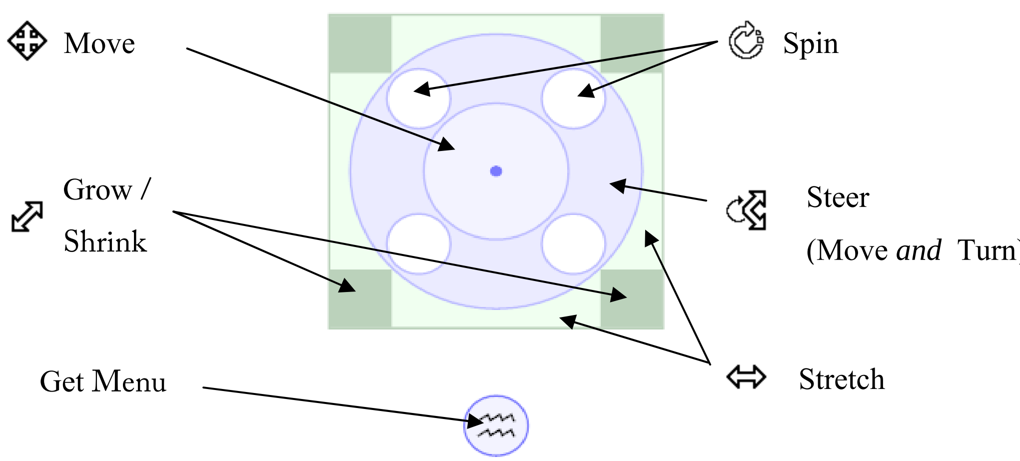
Handheld Controller
(either button)

Top Side of Tablet
(pull switch left or right)

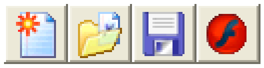


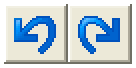
Object Handle

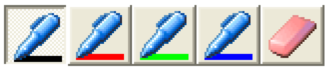
This control appears on top objects after you select them. It will change objects in various way, depending on where you touch it with your pen. Familiarize yourself with these.



Toolbar Buttons

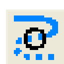
 **New Animation**, **Open Animation**, **Save Animation**, and **Export**

 **Undo** or **Redo** previous commands

 Choose **Pen Color** or **Eraser**

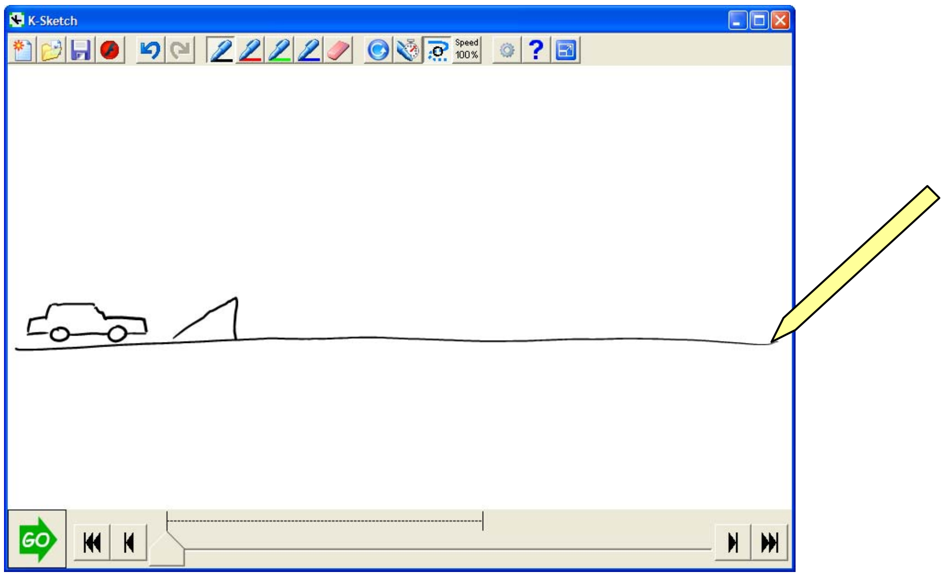
 **Loop Playback**: When pressed, the animation will repeat when it plays to the end.

 **Record Drawing**: When pressed, every line you draw will be animated in time.

 **Show Motion Paths**: When pressed, the paths of animated movements are visible.

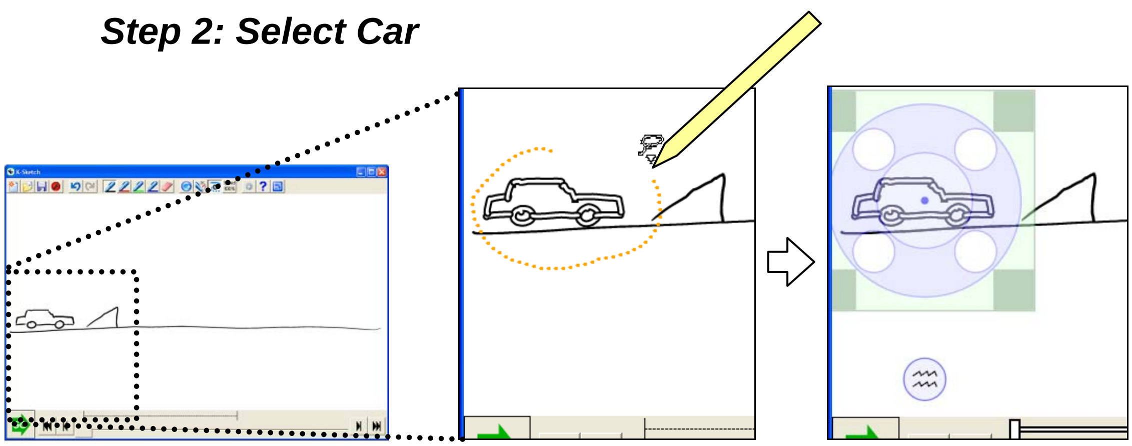
Adjust Speed: Shows or hides a slider for making the animation slower or faster.

Step 1: Draw Background and Car



Start by drawing the ground, the ramp, and the car.

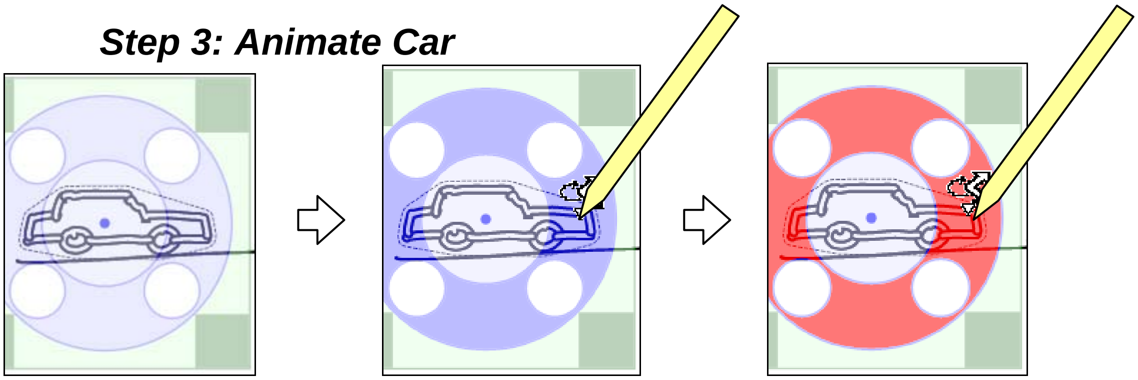
Step 2: Select Car



Select the car by holding an **Alternate** button (see top of page 2) and drawing a circle around the car with the pen. When you finish, all the lines of the car should be displayed as outlines (indicating that they are selected) and the **Object Handle** should be visible above them. Try putting your pen down in the various regions of the **Object Handle** and dragging a line to see all the ways you can change the car.

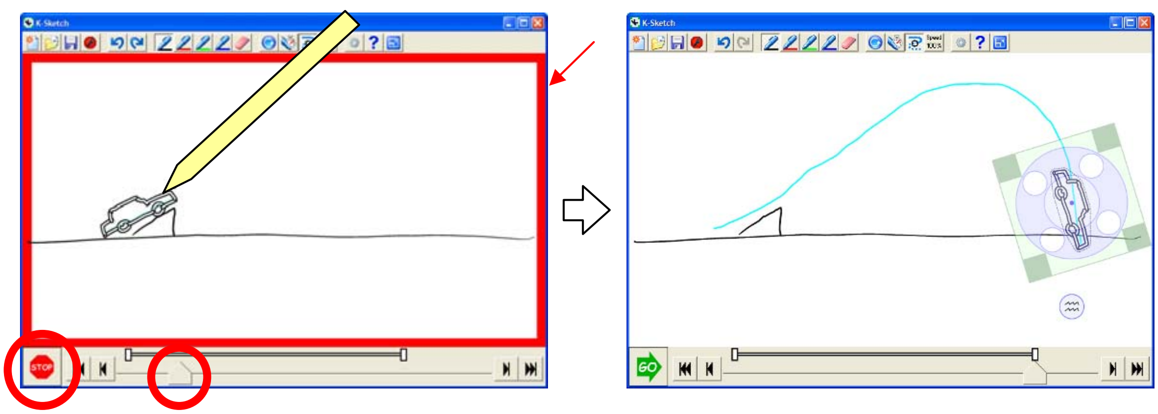
Note: To get rid of the **Object Handle**, select empty space (hold an **Alternate** button and tap the pen in a place with no ink strokes). The **Object Handle** will also disappear if you draw or erase outside of it.

Step 3: Animate Car



Move your pen to the **Object Handle's** "Steer" region. When you do so, the region will turn dark blue as shown above center. Dragging the **Object Handle** in this region will make the car turn to face the direction it is moving as it follows the pen. This is the motion we want, but we need to take another step in order to record this motion as an animation. Hold an **Alternate** button to indicate your desire to record the motion over time. When you hold this button, the "Steer" region turns red as shown above right.

Note: Changes to an object occur instantly during playback unless you record them.



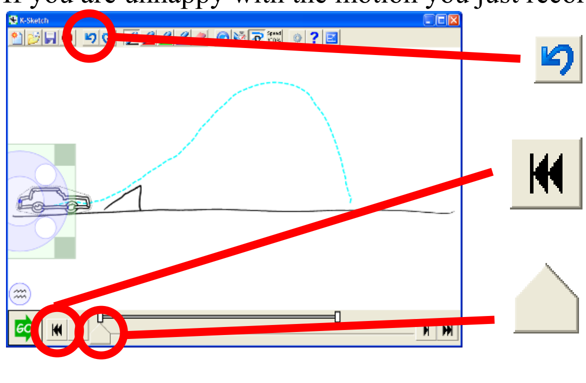
Now drag the car at a constant speed along the path shown. (Don't worry if the motion doesn't look quite right. We'll fix it in the next step.) Note all the ways that K-Sketch tells you it is recording a motion. The **Object Handle** disappears, the **Go Button** changes to **Stop**, the **Time Slider Control** begins to move to the right, and a red frame appears around the canvas. If **Show Motion Paths** is pressed, the path of the motion is shown as well. When you finish dragging, the screen should look approximately as it does above right.

Congratulations! You've recorded your first animation. Press the **Go Button** to view it. If you do not hold the pen near the screen while the animation is playing, the motion paths and **Object Handle** will be invisible.

Remember: Unless you press an **Alternate** button before dragging with the **Object Handle**, the edits you make will appear instantaneous when you play back the animation.

Step 4: Rewind and Play

If you are unhappy with the motion you just recorded, there are several ways to re-do it.



Undo: Return to the state just before the last operation

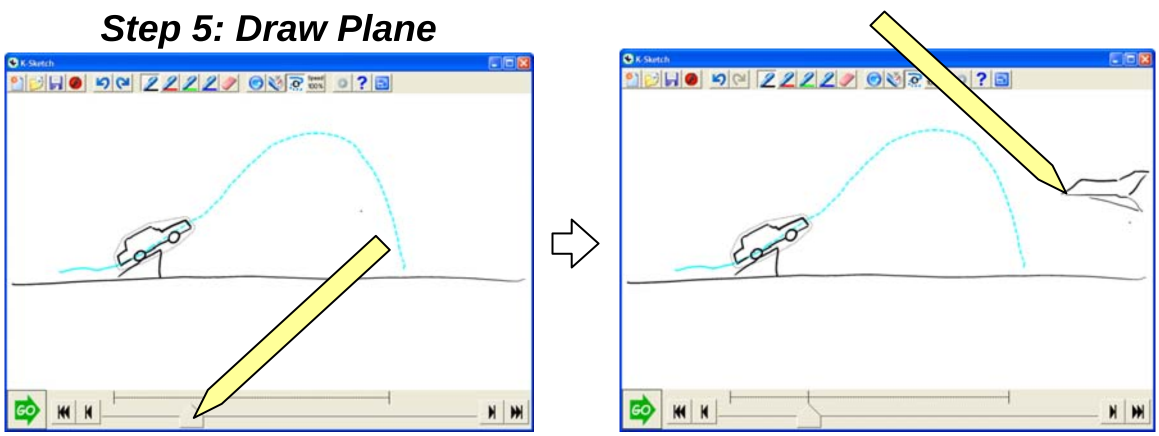
Go to Beginning: Recording a new motion that happens at the same time as an existing motion erases the old one.

Time Slider Control: Move to any point in the animation. Recording a new motion part way through an old motion erases the part after this point.

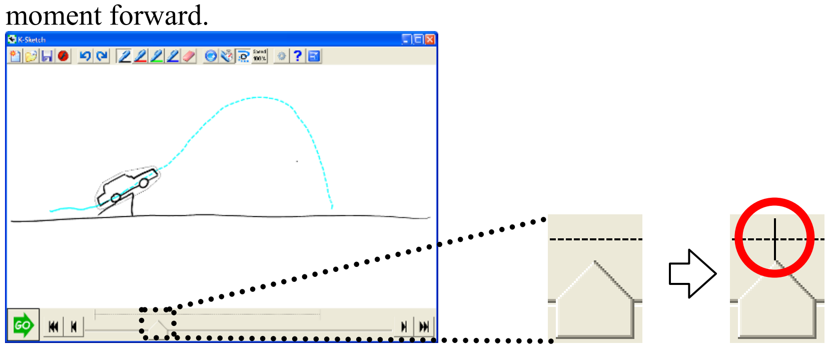
Try re-recording all or part of the car's motion using all of these methods.

Note: If you are unhappy with your drawing of the car, you can rewind to the beginning, select the **Eraser** tool and erase it (see Step 7).

Step 5: Draw Plane

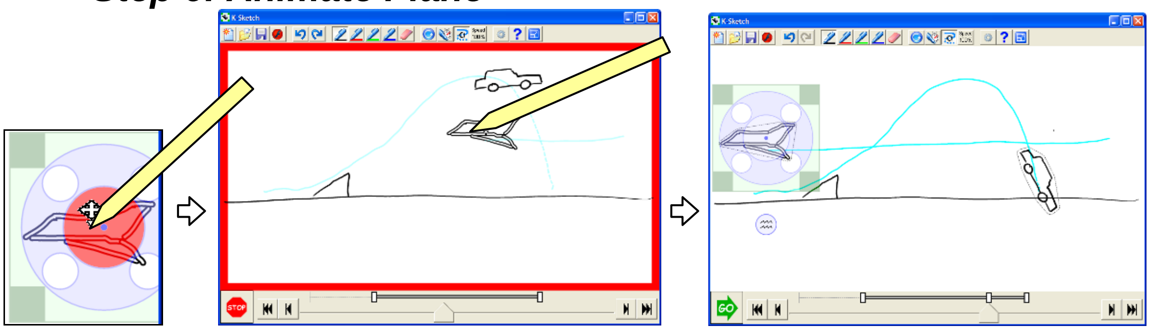


Draw the plane at the moment it appears. Move the time slider to the point shown above, when the car is on the ramp, and then draw the plane. The plane will be visible from this moment forward.



A tic mark appears in the timeline at the moment the plane appears to indicate that an event occurs at this time. You can drag these tic marks to change when events occur.

Step 6: Animate Plane

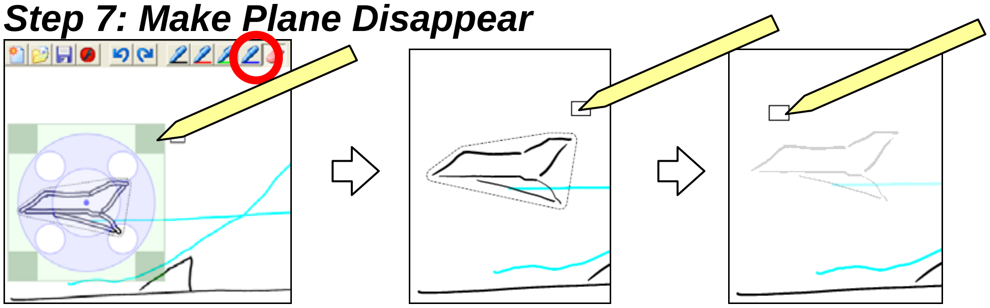


Select the plane (see Step 2), and hold the pen over the "Move" region in the center of the **Object Handle**. Hold an **Alternate** button so that this region is red, as shown above left. Then drag the plane to the left, recording its flying movement. The car moves as you drag so you can coordinate the two motions. Do your best to move the plane steadily and keep it under the jumping car.

When an object is selected, any timeline events related to that object are highlighted in white as shown above. Here we see three events for the plane. The first marks the appearance of the plane and the beginning of its movement. The second marks the end of its movement. The third marks the end of the animation. (***Bug***: you cannot move tic marks if an object is selected.)

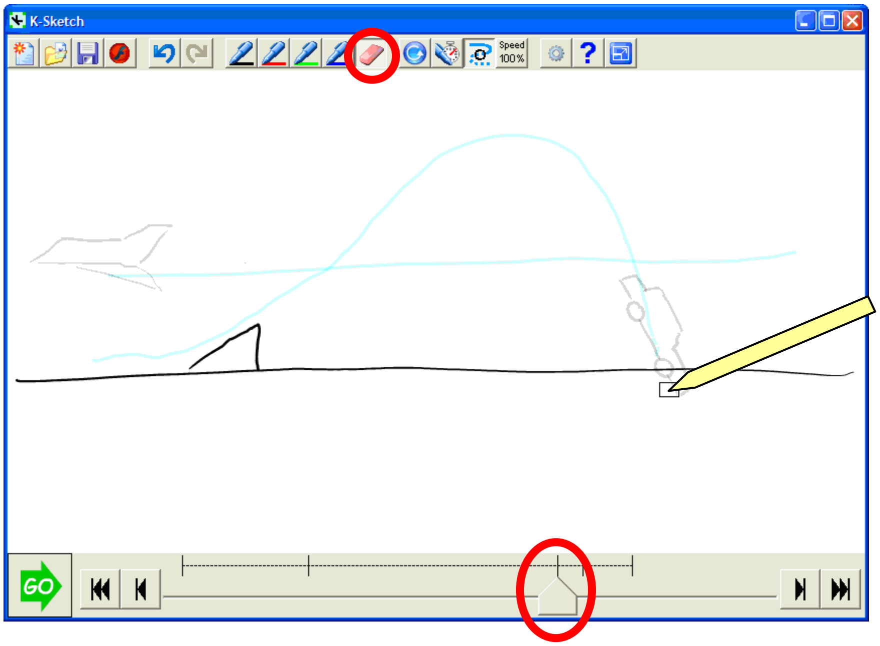
When you finish, your screen should look similar to that shown at the top right of this page. Rewind (see Step 4) and play your animation (using the **Go Button**) to see how you like it. If you are unhappy with the plane's motion, move the **Time Slider Control** to the moment the plane appears and re-record the motion (see Step 4).

Step 7: Make Plane Disappear

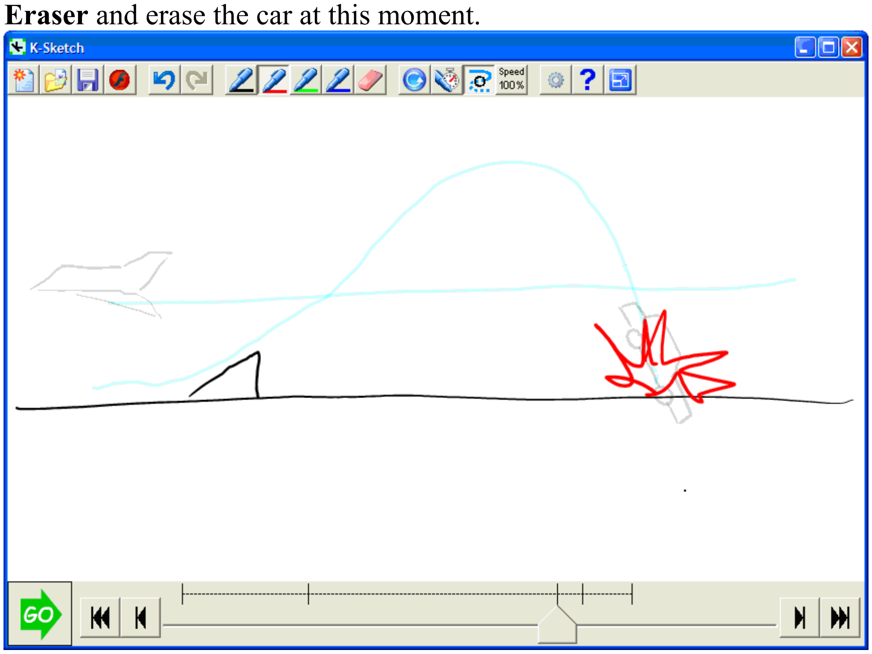


Make the plane disappear when it reaches the far left of the screen so that it appears to fly away. Select the **Eraser** (circled in red above) and erase the plane. If the **Object Handle** is still visible, it will disappear when you tap the eraser outside of it. When you erase the plane, it appears grayed out to indicate that it was visible in the past but is invisible in the future. ***Note***: You cannot erase motion path lines.

Step 8: Animate Explosion



As a final touch, make the car explode when it hits the ground. Move the time slider to the end of the animation or to the moment when the car stops moving. Then select the **Eraser** and erase the car at this moment.



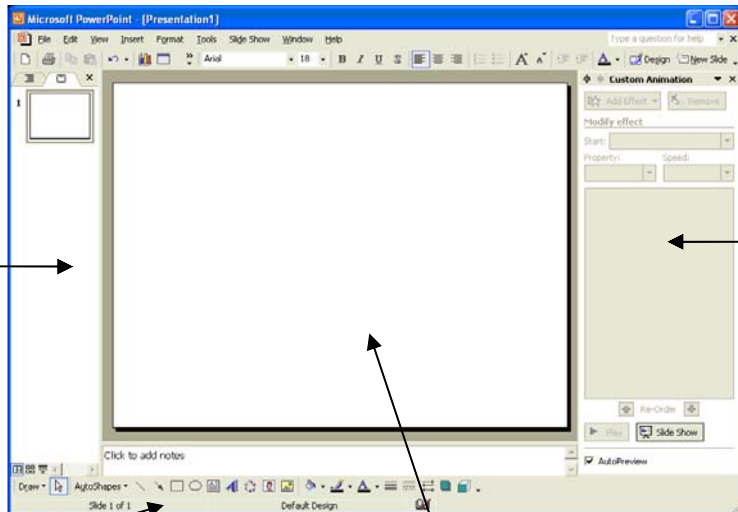
Finally, select the **Red Pen** and draw an explosion on top of where the car was, as above.

That's the final step! Rewind and play the full animation.

E.7 PowerPoint Practice Task Tutorial

Before we begin the practice task, we provide the following figures as a reference. We suggest you study them now and refer to them often in the future.

PowerPoint Overview



Slides Window: Shows all the slides in the presentation. You should only need one.

Task Pane: Currently showing the "Custom Animation" view, where you will access most animation controls. If this ever disappears, you can get it back by selecting "Custom Animation" in the "Slide Show" menu.

Drawing Toolbar: Where you will access most tools related to drawing on the canvas.

Main Window: Where you will design and view your animation.

Drawing Toolbar

PowerPoint switches between tools depending on the cursor location. By selecting one of the following tools, you can override this automatic selection for a single drag operation.

 **Selection tool**: for selecting objects

 **Line, Arrow, Rectangle, Oval, and Text Box tools**: for drawing objects. Controls for some text properties (e.g. font size) are found above the canvas.

AutoShapes menu: access to controls for drawing standard shapes.

Draw menu: where you access drawing controls such as grouping or rotating.

 **Fill Color Button (or menu), Line Color button (or menu)**: When used as buttons, these apply the shown fill color or line color to objects. When used as menus, these allow you to choose between fill and line colors or remove the line or fill.

Task Pane (for Custom Animation)

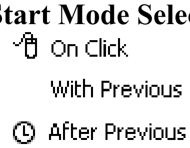
Add Effect: Menu of "effects" that animate drawn objects.

Remove Effect: Deletes the currently selected effect. (Pressing the delete key does the same thing.)

Play Button: For viewing the animation

Reorder Buttons: For changing the order of effects in the list. Dragging and dropping an effect does the same thing.

Start Mode Selector: Indicates when this effect should start

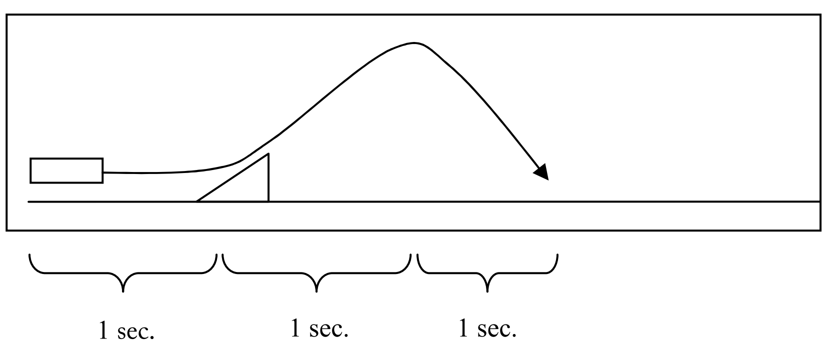


: Start when the user clicks a button
: Start at the same time as the effect listed above this one
: Start when all the effects listed above this one are finished

Speed Selector: Indicates how long this effect should take to run

Very Slow	: 5 seconds
Slow	: 3 seconds
Medium	: 2 seconds
Fast	: 1 second
Very Fast	: ½ second

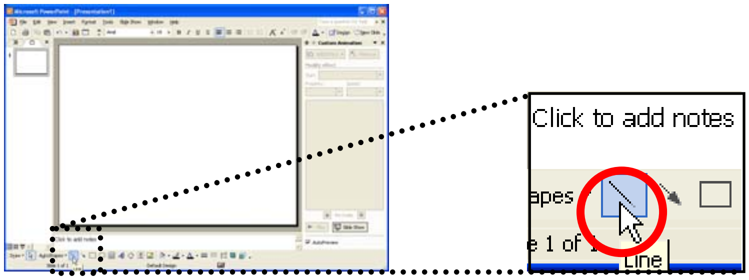
Step 1: Plan Layout and Timing



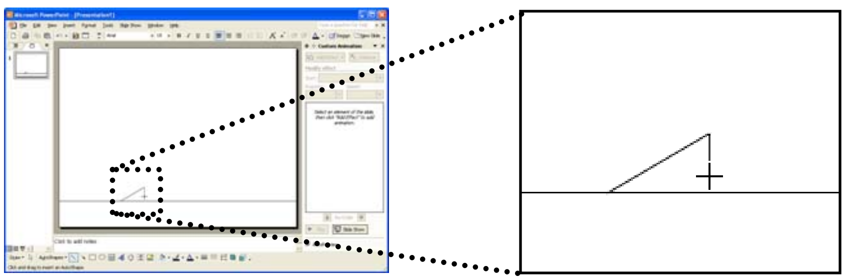
Let's use simple shapes to make this animation (e.g. a box instead of a car) and plan this animation to last three seconds. The 1st second will cover the movement of the car to the ramp. The 2nd second will cover the car's ascent, and the 3rd will cover the descent.

We'll make the plane move during the 2nd and 3rd seconds. We'll also make the highest point of the car's path half way between the left and right sides of the page. This will allow us to break up the plane's movement into two equal parts.

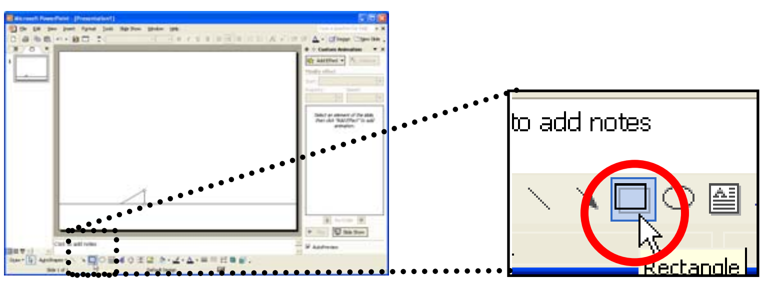
Step 2: Draw Ground and Car



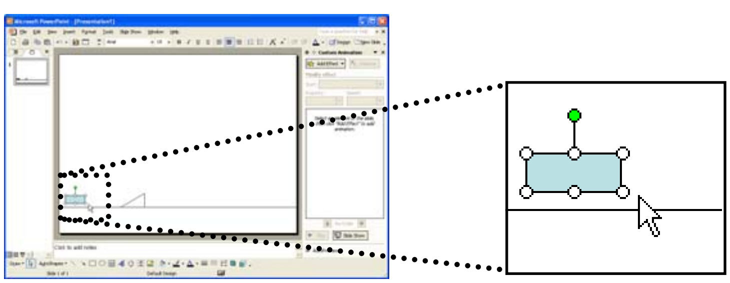
To draw the ground and ramp, select the **Line** tool.



Draw the ramp so it starts roughly ¼ of the page width from the left. Remember, you must re-select the **Line** tool before drawing each line.

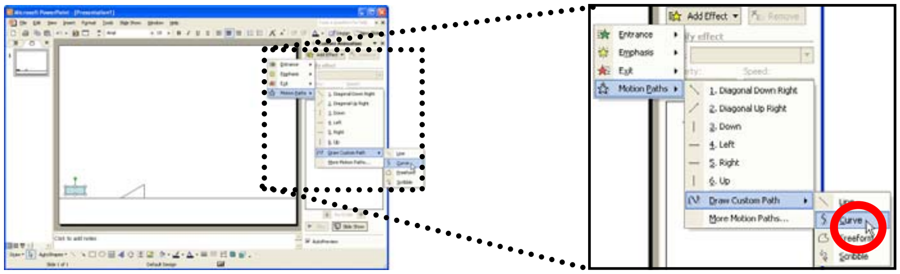


To draw the car, select the **Rectangle** tool.

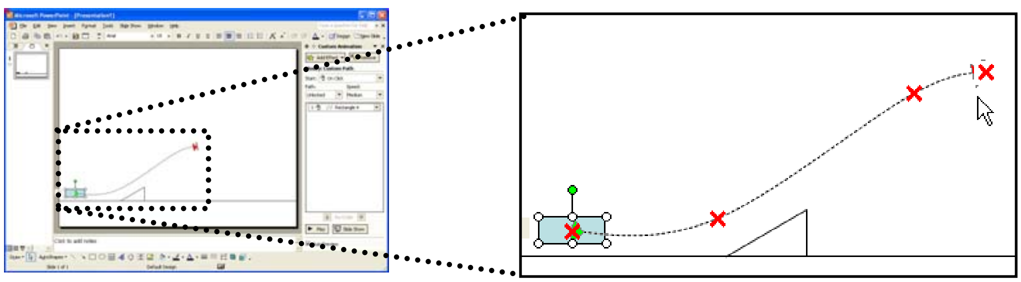


Draw the car to the far left of the page.

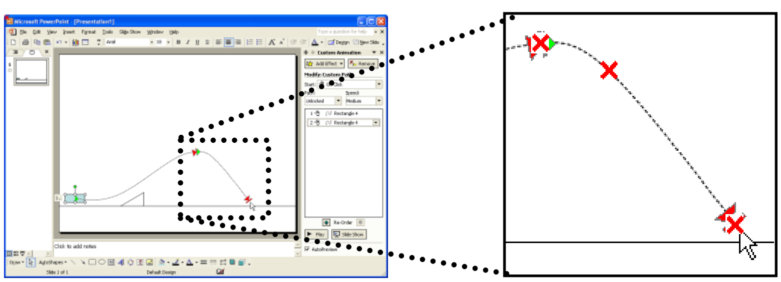
Step 3: Add Car Movement Effect



Let's break up the movement of this car into two chunks. This will make it a little easier to coordinate events when the car reaches its highest point. Make sure the rectangle is selected, and then select **Add Effect** → **Motion Paths** → **Draw Custom Path** → **Curve**.



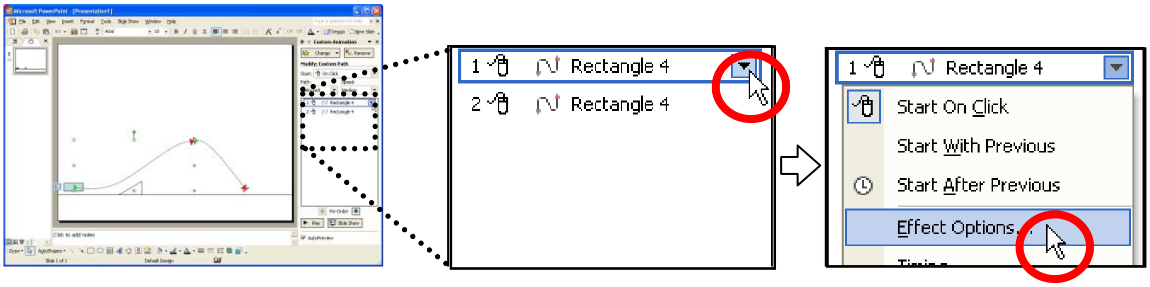
Draw the first part of the path approximately as shown above. The curve above was obtained with four clicks where the four red Xs appear. Press **Escape** to finish drawing. When you finish, the effect you added will play once.



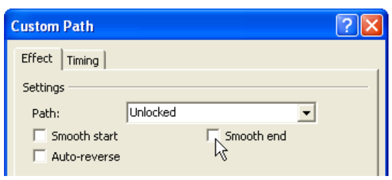
Now select Add **Effect** → **Motion Paths** → **Draw Custom Path** → **Curve** again and draw the last part of the car's path. The curve above was obtained with three clicks where the red Xs appear. Now we have two effects making up the path of our car.

If you make a mistake when drawing any of these paths, you can right-click on the path and select **Edit Points** in the context menu. This will allow you to adjust the points rather than draw a whole new path. **Note:** Try out some of the other effects under **Add Effect** → **Motion Paths** → **Draw Custom Path**. In particular, **Scribble** allows you to draw a path without specifying individual points.

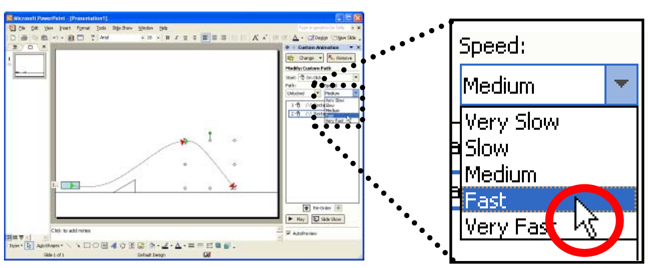
Step 4: Set Car Movement Effect Options



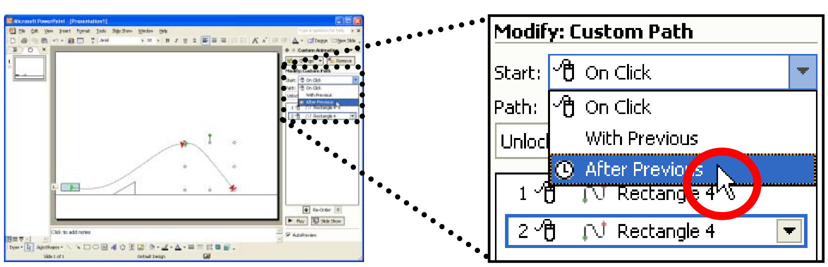
If we played our animation now, it wouldn't look quite right. One thing we'd notice is that the car speeds up and slows down over the course of each effect. To keep our effects moving at constant speed, we'll have to uncheck the "Smooth start" and "Smooth end" options. Start by selecting "Effect Options…" in the first effect's menu as shown above.



In the dialog box that appears, uncheck "Smooth start" and "Smooth end" and click "OK". Repeat this process for the second effect.



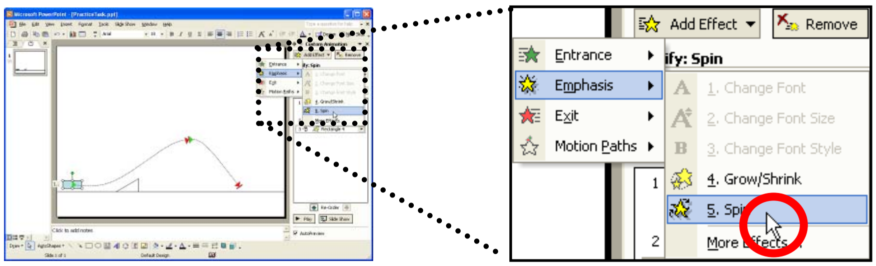
Another thing we'd notice if we played our animation is that the second effect runs too slowly. Select **Speed: Fast** to make this effect run in one second.



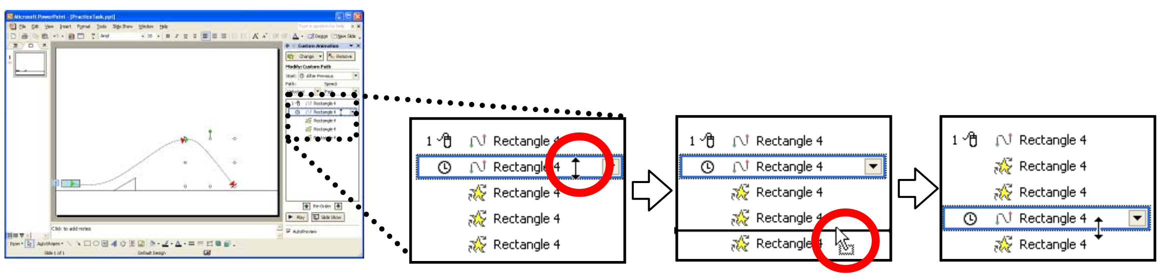
Finally, though it isn't strictly necessary in this case if we don't view a Slide Show, we should configure this effect to run "After Previous" instead of "On Click."

Step 5: Add Car Rotation Effects

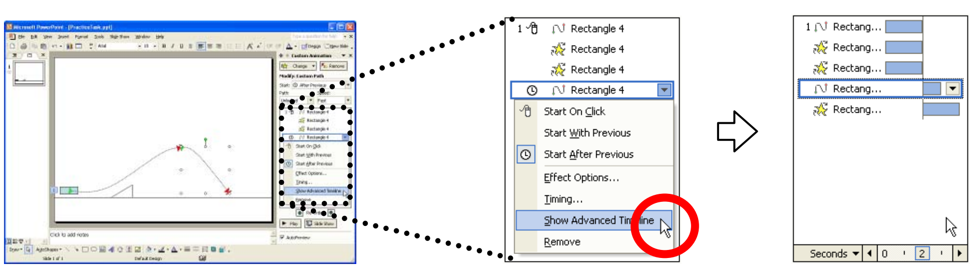
Our car doesn't look very convincing, because it doesn't point itself in the direction it's moving. There's no easy way to do this in PowerPoint, but we have set up the animation so that we can add "Spin" effects to orient the car approximately to its path. We'll add three Spin effects, one that turns the car up as it gets on the ramp, one that turns it down as it reaches its highest point, and one that turns it down again as it descends.



Select the car and add three Spin effects by selecting **Add Effect → Emphasis → Spin** three times. Each time you select it, the car will spin around once.

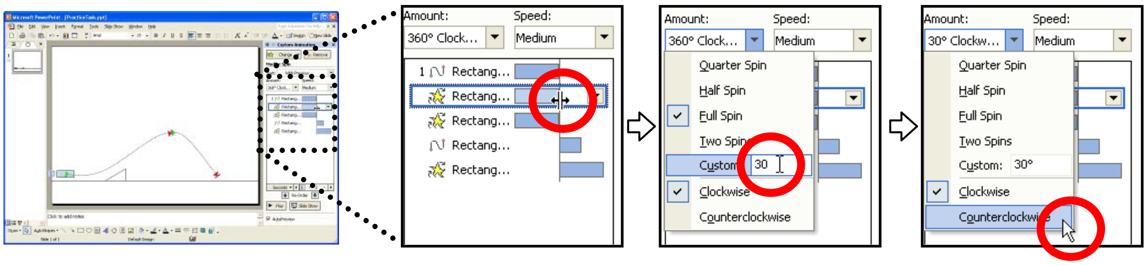


We want the first two of these spin effects to happen during the car's first movement effect, and the third spin effect to happen during the car's second movement effect. Change all three of these effects to move "With Previous" (see Step 4) and then drag the car's second movement effect down to group these effects as shown above.

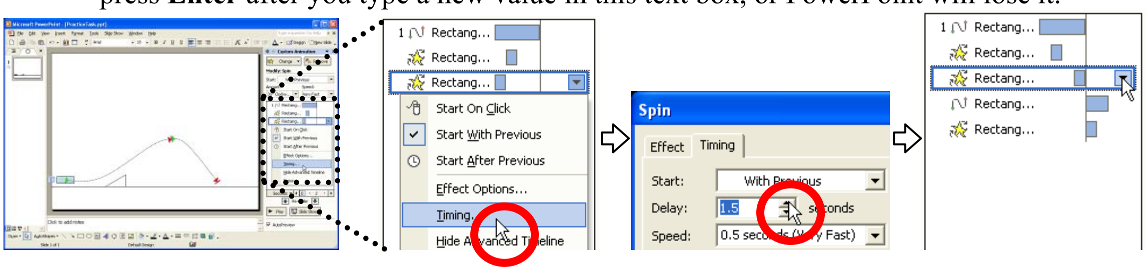


Our timeline is getting crowded, and it's hard to get a sense of what's going on. Click on any effect's menu and select "Show Advanced Timeline" to get a clearer picture of the sequence of events. Now the time and duration of each effect is visible in a little blue (or orange) box when the effect is selected. **Note**: if the advanced timeline is already visible, this menu item will read "Hide Advanced Timeline."

Step 5 (continued)



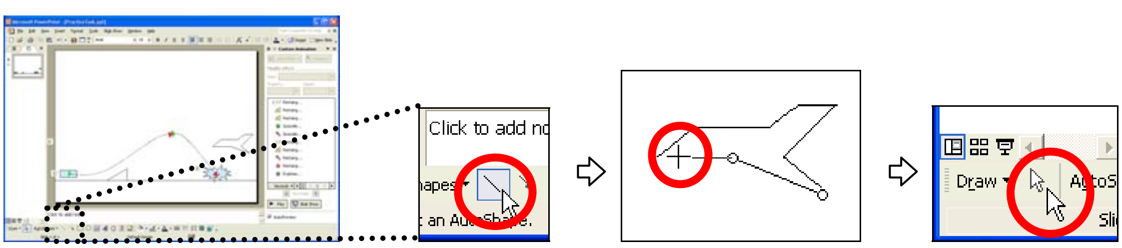
Now let's set the degree of rotation for each of these spin effects. First, let's have the car rotate 30° counter-clockwise as it goes up the ramp. Then we'll have it rotate 30° clockwise as it levels off at the top of its jump. Finally, it will rotate another 40° clockwise when it takes its dive. You can set these options with the **Amount** control that appears above all effects when you select a spin effect, as shown above. **Note**: You must press **Enter** after you type a new value in this text box, or PowerPoint will lose it.



Finally, we need to set the duration of all these spin effects to "Very Fast" (see Step 4) and make sure they start at the correct time. The third should already start at the right time, but the first two do not. Let's add a 0.5 second delay to the start of the first spin effect, and a 1.5 second delay to the start of the second. You can do this through the **Timing** dialog box (as shown above) or by moving and resizing the little blue (or orange) boxes in the timeline. Try doing it both ways.

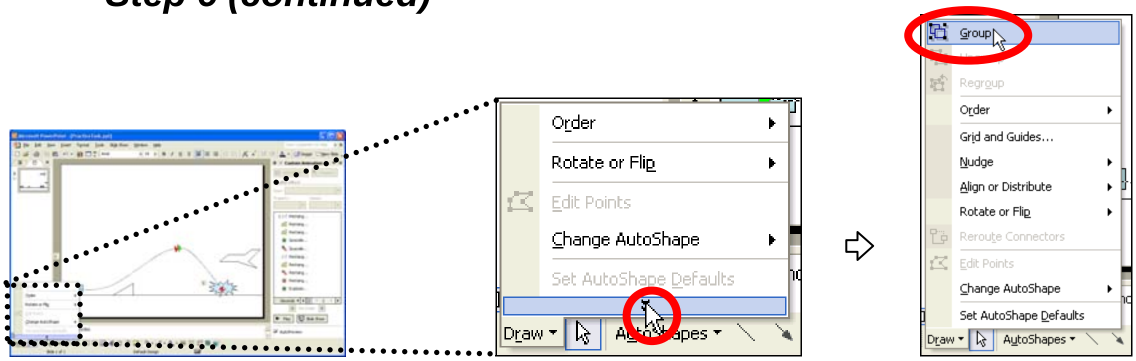
Now, when we play our animation, the car moves more like we would expect.

Step 6: Draw Plane



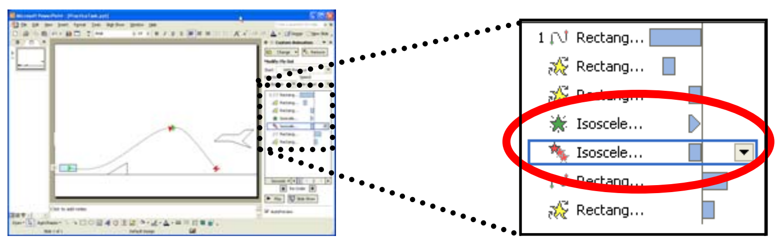
Now let's draw the plane. First, double-click on the **Line** tool. Double-clicking makes it possible to use a tool multiple times with out re-clicking it. Then draw a plane by dragging out several lines in succession. Then press **Escape** to get your pointer back.

Step 6 (continued)

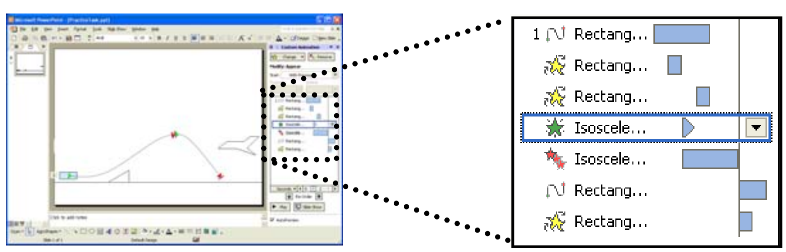


Now group these lines so that you can move them all together. Select all of the lines by dragging a selection box around them. Then select **Draw** → **Group.** You may need to press the small arrows at the bottom of the draw menu to make **Group** visible in the **Draw** menu.

Step 7: Animate Plane

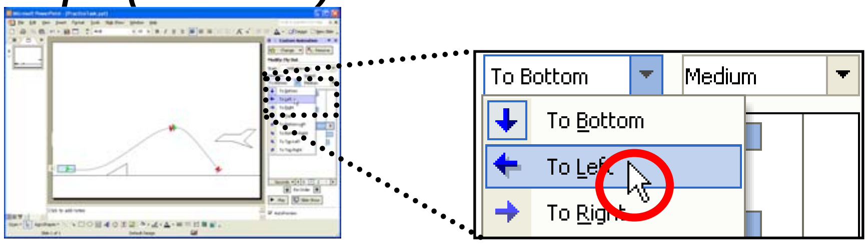


Animate the plane with two effects. Select the plane and then **Add Effect** → **Entrance** → **Appear** and then **Add Effect** → **Exit** → **Fly Out**. Configure both of these to move "With Previous" (see Step 4) and position them under the first two spin effects as shown.

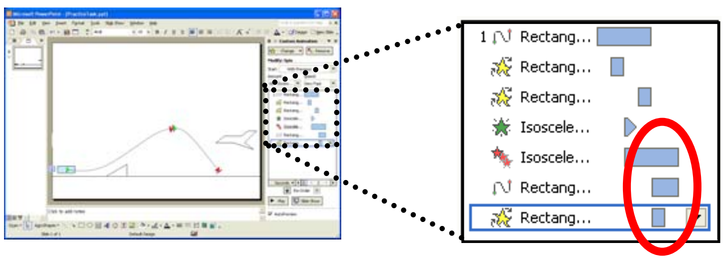


If you did the above step in the order they were listed, these effects will have the delay of the effects listed before them (1.5 seconds). Change the delay for each to 1.0 second (see Step 5), and change the speed of the "Fly Out" effect to "Medium" (see Step 4) so it lasts 2 seconds. Your view should now look as above when the "Appear" effect is selected.

Step 7 (continued)

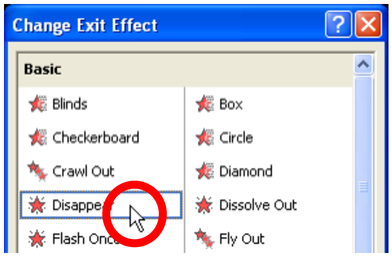


Now select the “Fly Out” effect, and notice that a “Direction” selector has appeared above the effects. Select **Direction → To Left** to make the plane fly off to the left. Play the animation and watch the plane pass under the car.

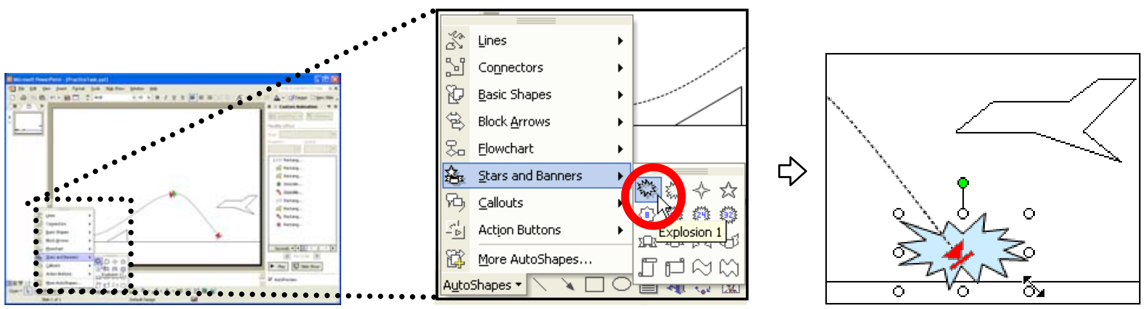


Some of our recent changes created a problem. We configured the car’s second movement effect to start “After Previous,” but the plane’s “Fly Out” effect takes so long that the car pauses in the middle of its motion. Re-configure the car’s second movement effect to occur “With Previous” (see Step 4), and give both it and the last spin effect a delay of 2 seconds (see Step 5). The timeline should now look as it does above, and the animation should look correct when you play it.

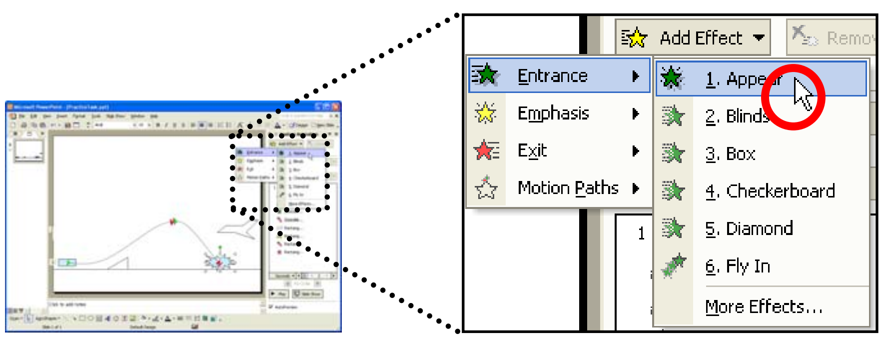
Step 8: Add Explosion



One step remains. Our car needs to explode when it hits the ground. Let's start by making the car disappear. Select the car and then select **Add Effect → Exit → More Effects…**. Then select "Disappear" in the resulting Dialog box as shown above. Configure this effect to start "After Previous" (see Step 4). The car should now disappear when it hits the ground.



Now draw the explosion by selecting **AutoShapes → Stars and Banners → Explosion1** in the Drawing Toolbar and drawing the explosion at the end of the car's path.



Finally, we need to make this animation appear when the car disappears. Select **Add Effect → Entrance → Appear**, and configure this effect to start "After Previous" (see Step 4).

That's the final step! Click Play to test the full movie.

E.8 Comfort-level Questionnaire

This questionnaire was identical to the one found in Section D.1.6.

E.9 Cognitive Load Definitions

Mental Demand	The amount of mental and/or perceptual activity that was required to accomplish these tasks (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.). Were the tasks easy or demanding, simple or complex, exacting or forgiving?
Physical Demand	The amount of physical activity was required to accomplish these tasks (e.g., pushing, pulling, turning, controlling, activating, etc.). Were the tasks easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
Temporal Demand	The amount of time pressure you felt due to the rate or pace at which the tasks or task elements occurred. Was the pace slow and leisurely or rapid and frantic?
Performance	How successful you think you were in accomplishing the goals of the task set by the experimenter (or yourself). How satisfied were you with your performance in accomplishing these goals?
Effort	How hard you had to work (mentally and physically) to accomplish your level of performance.
Frustration Level	How insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed, and complacent you felt while accomplishing these tasks.

E.10 Cognitive Load Worksheet

Place a mark on each scale that represents the magnitude of each factor in the tasks you just performed.

Mental Demand

low high

Physical Demand

low high

Temporal Demand

low high

Performance

good poor

Effort

low high

Frustration Level

low high

E.11 Cognitive Load Weights

For each of the following pairs, circle the item that provided the most significant source of workload variation in these tasks.

Physical Demand	OR	Mental Demand
Temporal Demand	OR	Mental Demand
Performance	OR	Mental Demand
Frustration Level	OR	Mental Demand
Effort	OR	Mental Demand
Temporal Demand	OR	Physical Demand
Performance	OR	Physical Demand
Frustration Level	OR	Physical Demand

Effort	OR	Physical Demand
Temporal Demand	OR	Performance
Temporal Demand	OR	Frustration Level
Temporal Demand	OR	Effort
Performance	OR	Frustration Level
Performance	OR	Effort
Effort	OR	Frustration Level

E.12 Post-test Questionnaire

How easy was it to work with these tools?	Very Easy			Neutral			Very Hard	Don't Know
1) K-Sketch (the pen-based tool)	1	2	3	4	5	6	7	?
2) PowerPoint (the desktop computer tool)	1	2	3	4	5	6	7	?

How fast were you at operating these tools?	Very Fast			Neutral			Very Slow	Don't Know
3) K-Sketch (the pen-based tool)	1	2	3	4	5	6	7	?
4) PowerPoint (the desktop computer tool)	1	2	3	4	5	6	7	?

5) Is there anything you particularly *liked* about K-Sketch?

6) Is there anything you particularly *disliked* about K-Sketch?

7) Is there anything you particularly *liked* about PowerPoint?

8) Is there anything you particularly *disliked* about PowerPoint?

E.13 Video Taping Consent Form

UNIVERSITY OF WASHINGTON
VIDEO, AUDIO, AND PHOTOGRAPHIC RECORDING PUBLICATION CONSENT FORM
Animation Tool Comparison Study

Investigator Name:	**Title**	**Department**	**Contact Information**
Richard C. Davis	Graduate Student	Computer Science & Engineering	206-992-9363 rcdavis@cs.washington.edu
James Landay	Assistant Professor	Computer Science & Engineering	206-685-9139 landay@cs.washington.edu

Researchers' statement

USES OF THE RECORDINGS AND PHOTOGRAPHS

The purpose of this study is to compare animators' speed, mental workload, and satisfaction when producing animations with an experimental tool called "K-Sketch" and existing tools such as Adobe Flash and Microsoft PowerPoint. It is hoped that this research will benefit Human-Computer Interaction researchers in general and help us in particular to make K-Sketch better. We have video recorded you as you used these tools. We would like to keep the tapes for an extended period of time, until December 31, 2008. We may want to use videotape clips for educational presentations or in academic presentations.

It is possible for someone who knows you to recognize your voice or image from the videotape, audiotape, or photograph. We will place a digital mosaic over your face in any photo or video that is used in a publication.

We ask your permission to use the following recordings and photographs in academic public presentations, educational settings and publications such as journals, magazines, newspapers, and online multimedia publications, and web sites:

____
____
____

I give my permission for the research team to use the above videotape or photograph in the following way:

- ❑ Academic public presentations
- ❑ Educational settings
- ❑ Scientific or educational journals, magazines, newspapers
- ❑ Online multimedia publications
- ❑ Web sites
- ❑ Keep the tapes for research purposes for one year.

Printed name of researcher	Signature of researcher	Date

Subject's statement

I have had an opportunity to review the recording and photographs referenced above. I give my permission to the researchers to use the items as I have indicated above in this consent form. I understand that my name will not be published in connection with any such presentation or publication. I will not receive any compensation for the use of the recordings or photographs. I will receive a copy of this consent form.

Printed name of subject	Signature of subject	Date

APPENDIX

F Laboratory Evaluation 1 Data

F.1 Demographic Data

#	Occupation	Sex	Drawing skill (1 = none, 5 = excellent)	How often would you create animations?	Expertise (1 = complete beginner, 7 = expert)		
					Computers	PPT /Word	Tablet PCs
1	(none given)	F	3	1/year	5	2	1
2	illustrator / graphic designer	F	5	not sure	5	1	1
3	grad student: comp. bio.	F	2.5	1/week	6	4	1
4	engineering manager	M	3	not sure	7	4	1
5	artist	M	5	1/year	4	2	1
6	chef, student: culinary arts, microbiology, naturopathic medicine, design	M	3	1/week	5	4	2
7	undergrad student: mech. eng.	M	3	1/year	5	2	2
8	dental assistant	F	3	not sure	5	2	1
9	marketing director	F	2	not sure	4	3	1
10	student: Chinese medicine, former "human performance" consultant	M	2	1/year	4	3	1
11	teacher, software consultant	F	4	1/year	7	6	3
12	baker, undergrad student	F	4	1/week	5	2	1
13	technology manager	F	3	1/month	5	6	1
14	undergrad student: undecided	F	3	1/month	4	3	1
15	photographer	M	3	1/month	6	2	4
16	grad student: electrical eng.	M	3	1/year	6	5	1
	Averages		*3.2*		*5.2*	*3.2*	*1.4*

Study 1 demographic data

#	Animation experience (hours)			Animation tools used	Why not animate more? (1=Strongly Disagree, 7=Strongly Agree)			Tool sequence	Task sequence
	PowerPoint custom animation	PowerPoint other animation	Other animation tools		Can access animation tools?	Complexity discourages?	Time required discourages?		
1	0	0	0		4	3	6	K,P	A,B
2	0	0	0		4	6	6	P,K	A,B
3	0	0	0		6	6	6	K,P	B,A
4	0	40	0					P,K	B,A
5	0	0	0		4	5	3	K,P	A,B
6	1	10	5	2 Flash 3 Dreamweaver	6	2	4	P,K	A,B
7	5	15	10	Autodesk animator	4	4	5	K,P	B,A
8	0	30	0		4	4	4	P,K	B,A
9	5	20	0		6	6	6	P,K	A,B
10	5	0	0		2	7	7	K,P	A,B
11	2	100	30	Keynote	5	1	2	K,P	B,A
12	0	0	0		6	5	6	P,K	B,A
13	0	5	10	AfterEffects	4	2	6	K,P	A,B
14	0	10	0		5	5		P,K	A,B
15	0	0.5	0		7			K,P	B,A
16	0	3	8	8 Solidworks anim. 2 Matlab	3	6	7	P,K	B,A
Avg.	1.1	14.6	3.9		4.7	4.4	5.2		

Study 1 demographic data (continued) and tool/task sequence. PowerPoint "other" animation consisted mainly of animated slide transitions that I did not consider to be meaningful animation experience. Tool sequence legend: K = K-Sketch, P = PowerPoint.

F.2 Experimental Data

	Task time (minutes)				How fast? (1–7, 1 = fastest)	
	K-Sketch		*PowerPoint*		*K-Sketch*	*PowerPoint*
Participant	A	B	A	B		
P1	20.50	15.93	27.35	21.45	3	5
P2	5.10	10.82	40.67	60.00	5	7
P3	4.25	15.53	13.08	15.58	3	4
P4	9.67	3.17	15.83	13.87	2	4
P5	7.10	4.63	18.13	15.40	3	5
P6	3.50	3.97	24.03	12.45	1	3
P7	3.17	5.10	12.90	13.52	2	4
P8	7.48	4.38	15.05	16.25	3	5
P9	6.55	3.78	16.53	13.75	2	5
P10	5.10	5.47	23.90	10.93	2	5
P11	11.12	4.02	14.03	10.97	3	3
P12	4.37	12.18	25.88	59.37	3	6
P13	4.92	9.05	20.90	9.80	4	4
P14	3.17	3.62	15.17	10.22	1	5
P15	3.45	6.85	13.83	20.03	6	4
P16	3.33	5.02	13.40	12.05	1	3
Mean	**6.76**[a]		**19.57**[a]		**2.75**	**4.50**
Std. dev.	**4.30**[a]		**12.24**[a]		**1.39**	**1.10**

[a] Means and standard deviations for task times are shown, but my statistical analysis was conducted on the natural log of task time: K-Sketch *M* = 1.76 (5.81 minutes), *SD* = .526; PowerPoint *M* = 2.85 (17.35 minutes), *SD* = .451.

Speed-related measures. Differences in means are statistically significant.

	Practice time (minutes)		How easy? (1–7) (1 = easiest)		NASA-TLX (1–100)	
Participant	*KSK*	*PPT*	*KSK*	*PPT*	*KSK*	*PPT*
P1	26.17	30.00	1	4	70	68
P2	33.75	54.00	3	5	-[a]	-[a]
P3	31.32	32.72	3	4	35	66
P4	24.25	34.63	2	4	30	39
P5	19.93	43.22	2	4	45	70
P6	26.17	61.33	1	5	6	27
P7	30.57	39.30	1	4	19	54
P8	28.05	48.12	2	5	22	73
P9	20.82	45.35	1	4	22	55
P10	25.75	36.87	2	6	51	91
P11	26.13	30.40	2	4	35	53
P12	21.70	70.42	2	5	11	56
P13	25.88	43.00	4	3	44	54
P14	19.33	34.68	1	4	29	60
P15	36.20	51.45	6	3	52	40
P16	25.70	40.87	1	4	8	44
Mean	**26.36**	**43.52**	**2.13**	**4.25**	**32.0**	**56.7**
Std. dev.	**4.80**	**11.36**	**1.36**	**0.78**	**18.1**	**15.9**

[a] This participant did not complete the final questionnaire necessary to compute cognitive load.

Simplicity-related measures. Differences in means are statistically significant.

Number	Audience
1	No one[a]
2	A single colleague in a private meeting
3	Ten colleagues in a private meeting
4	A student I am tutoring
5	30 students in a class I am teaching
6	300 students in a class I am teaching
7	30 professionals during a business presentation I'm making
8	300 professionals during a business presentation I'm making

[a] For comfort showing, this was listed as "myself (keep in private notebook only)."

Audiences for comfort-level questionnaire.

	Comfort showing (1-7, 1 = extremely uncomfortable)				Comfort creating (1–7, 1 = extremely uncomfortable)			
	K-Sketch		*PowerPoint*		*K-Sketch*		*PowerPoint*	
Participant	A	B	A	B	A	B	A	B
P1	4.75	5.88	5.63	5.50	5.00	4.75	4.75	5.13
P2	6.13	3.25	4.63	3.13	6.13	3.25	4.75	3.00
P3	4.38	3.13	4.88	5.50	4.50	4.25	4.00	4.00
P4	6.25	6.25	6.25	5.88	6.25	6.25	6.25	5.50
P5	5.17	6.00	3.13	4.63	6.57	7.00	3.25	2.75
P6	7.00	7.00	6.50	7.00	7.00	7.00	6.38	7.00
P7	5.63	5.63	5.38	5.75	5.63	5.38	5.38	4.88
P8	7.00	7.00	5.13	3.75	7.00	7.00	2.25	3.13
P9	4.88	4.75	5.13	2.25	5.00	4.75	2.88	2.00
P10	2.38	4.25	2.50	4.88	4.13	4.38	1.88	2.75
P11	5.25	5.50	4.88	6.25	5.50	6.50	3.88	4.25
P12	4.25	3.38	3.13	1.88	4.25	3.75	2.50	3.13
P13	3.13	4.25	5.88	5.88	4.63	4.38	5.13	5.88
P14	5.75	6.38	4.63	4.75	6.25	6.25	3.75	4.75
P15	5.00	4.63	6.00	5.00	4.50	4.13	3.75	3.38
P16	6.13	5.00	6.38	4.63	6.75	6.50	5.38	3.50
Mean	5.17		4.89		**5.46**		**4.10**	
Std. dev.	1.25		1.30		**1.14**		**1.35**	

Comfort showing to others and creating while being watched by others. The differences in means for comfort creating is statistically significant.

			Comfort showing by audience (1 = extremely uncomfortable, 7 = extremely comfortable)								Comfort creating by audience (1 = extremely uncomfortable, 7 = extremely comfortable)							
Participant	Tool	Task	A1	A2	A3	A4	A5	A6	A7	A8	A1	A2	A3	A4	A5	A6	A7	A8
1	K-Sketch	A	6	6	4	7	5	3	4	3	7	7	5	7	4	3	4	3
2	K-Sketch	A	7	6	6	6	6	6	6	6	7	6	6	6	6	6	6	6
3	K-Sketch	A	7	6	6	5	5	4	1	1	7	6	6	5	4	4	2	2
4	K-Sketch	A	7	7	7	7	7	7	4	4	7	7	7	7	7	7	4	4
5	K-Sketch	A	7	5	3	6	5	5	-	-	7	7	7	7	6	-	6	6
6	K-Sketch	A	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	K-Sketch	A	7	6	6	6	6	5	5	4	7	6	6	6	6	5	5	4
8	K-Sketch	A	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	K-Sketch	A	7	7	7	7	4	3	2	2	7	7	7	7	5	3	3	1
10	K-Sketch	A	7	5	2	1	1	1	1	1	7	6	5	3	2	1	6	3

(continued)			Comfort showing by audience (1 = extremely uncomfortable, 7 = extremely comfortable)								Comfort creating by audience (1 = extremely uncomfortable, 7 = extremely comfortable)							
Participant	Tool	Task	A1	A2	A3	A4	A5	A6	A7	A8	A1	A2	A3	A4	A5	A6	A7	A8
11	K-Sketch	A	7	7	6	6	6	4	3	3	7	6	5	6	5	5	5	5
12	K-Sketch	A	7	6	5	5	4	3	2	2	7	6	5	5	4	3	2	2
13	K-Sketch	A	7	6	5	3	1	1	1	1	7	6	6	4	4	4	3	3
14	K-Sketch	A	7	7	6	6	6	5	5	4	7	7	6	7	7	6	5	5
15	K-Sketch	A	7	7	6	6	5	5	2	2	7	7	5	5	4	4	2	2
16	K-Sketch	A	7	7	7	7	7	7	4	3	7	7	7	7	7	7	6	6
1	PowerPoint	A	7	7	6	7	5	4	5	4	7	7	5	7	4	3	3	2
2	PowerPoint	A	7	7	6	5	6	2	2	2	5	6	6	5	5	3	4	4
3	PowerPoint	A	7	6	5	5	4	4	4	4	7	5	4	5	4	3	2	2
4	PowerPoint	A	7	7	7	7	7	7	4	4	7	7	7	7	7	7	4	4
5	PowerPoint	A	6	5	3	3	3	2	2	1	6	4	3	3	3	2	3	2
6	PowerPoint	A	7	7	7	7	7	7	5	5	7	7	7	6	6	7	7	4
7	PowerPoint	A	7	7	5	6	6	5	4	3	7	7	5	6	6	5	4	3
8	PowerPoint	A	7	7	7	6	6	4	2	2	7	3	2	2	1	1	1	1
9	PowerPoint	A	7	7	7	7	7	3	2	1	7	6	2	3	2	1	1	1
10	PowerPoint	A	3	3	2	5	3	1	2	1	5	3	1	2	1	1	1	1
11	PowerPoint	A	7	6	4	5	5	4	4	4	7	6	4	5	3	2	2	2
12	PowerPoint	A	5	5	4	4	3	2	1	1	6	4	3	2	2	1	1	1
13	PowerPoint	A	7	6	6	6	6	6	5	5	7	6	5	5	5	5	4	4
14	PowerPoint	A	7	6	6	7	5	3	2	1	7	5	4	5	4	3	1	1
15	PowerPoint	A	7	7	7	7	5	5	5	5	7	7	4	4	2	2	2	2
16	PowerPoint	A	7	7	6	7	7	7	5	5	7	7	7	7	6	4	3	2
1	K-Sketch	B	6	7	5	7	6	5	6	5	7	7	5	7	4	3	3	2
2	K-Sketch	B	7	5	4	2	2	2	2	2	7	5	4	2	2	2	2	2
3	K-Sketch	B	7	4	3	4	3	2	1	1	7	6	5	5	4	3	2	2
4	K-Sketch	B	7	7	7	7	7	7	4	4	7	7	7	7	7	7	4	4
5	K-Sketch	B	7	6	6	6	5	-	-	-	7	7	7	7	7	-	-	-
6	K-Sketch	B	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	K-Sketch	B	7	7	6	6	6	6	4	3	7	6	6	6	6	5	4	3
8	K-Sketch	B	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	K-Sketch	B	7	7	7	7	4	3	2	1	7	7	7	7	4	3	2	1
10	K-Sketch	B	7	6	6	4	3	1	5	2	7	6	6	5	3	1	5	2
11	K-Sketch	B	7	7	5	7	6	6	3	3	7	7	6	7	7	6	6	6
12	K-Sketch	B	6	5	4	5	3	2	1	1	7	6	5	5	3	2	1	1
13	K-Sketch	B	7	6	6	4	4	3	2	2	7	6	5	4	4	4	3	2
14	K-Sketch	B	7	7	6	7	6	6	6	6	7	7	6	7	6	6	6	5
15	K-Sketch	B	7	7	4	7	5	5	1	1	7	7	5	5	4	3	1	1
16	K-Sketch	B	7	7	7	6	6	5	1	1	7	7	7	7	7	7	5	5
1	PowerPoint	B	7	7	6	7	5	5	4	3	7	7	6	7	5	3	3	3
2	PowerPoint	B	7	5	3	2	2	2	2	2	6	5	3	2	2	2	2	2

(continued)			Comfort showing by audience (1 = extremely uncomfortable, 7 = extremely comfortable)								Comfort creating by audience (1 = extremely uncomfortable, 7 = extremely comfortable)							
Participant	Tool	Task	A1	A2	A3	A4	A5	A6	A7	A8	A1	A2	A3	A4	A5	A6	A7	A8
3	PowerPoint	B	7	6	6	6	6	5	4	4	7	5	5	4	4	3	2	2
4	PowerPoint	B	7	7	7	7	7	7	3	2	7	7	7	7	4	4	4	4
5	PowerPoint	B	7	6	5	6	5	2	4	2	5	4	3	3	3	1	2	1
6	PowerPoint	B	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	PowerPoint	B	7	7	6	7	6	6	4	3	7	6	5	5	5	4	4	3
8	PowerPoint	B	7	7	7	3	2	2	1	1	7	5	5	3	2	1	1	1
9	PowerPoint	B	5	5	2	2	1	1	1	1	5	5	1	1	1	1	1	1
10	PowerPoint	B	6	6	6	6	5	4	3	3	5	4	3	4	2	1	2	1
11	PowerPoint	B	7	6	6	7	6	6	6	6	7	6	3	6	3	3	3	3
12	PowerPoint	B	4	3	2	2	1	1	1	1	6	5	3	4	3	2	1	1
13	PowerPoint	B	7	6	6	6	6	6	5	5	7	6	6	6	6	6	5	5
14	PowerPoint	B	7	7	6	6	4	3	3	2	7	7	5	6	4	3	3	3
15	PowerPoint	B	7	7	6	6	4	4	3	3	7	7	4	5	1	1	1	1
16	PowerPoint	B	7	6	5	6	6	5	1	1	7	6	5	4	2	2	1	1

Comfort showing animations to others and comfort creating them in front of others, separated by audience.

F.3 Incidents

F.3.1 K-Sketch Help Incidents

Incident Count	Participant	Task	Incident
2	1	A	timeline: sequential vs. concurrent motions
1	16	B	can motions be defined with interpolation?
3			*K-Sketch grand total*

F.3.2 PowerPoint Help Incidents

Incident Count	Participant	Task	Incident
3	1	A	timeline: order of effects (with/after previous, speed, delays)
3	1	B	timeline: order of effects (with/after previous, speed, delays)
4	2	B	timeline: order of effects (with/after previous, speed, delays)
2	10	A	timeline: order of effects (with/after previous, speed, delays)
1	10	B	timeline: order of effects (with/after previous, speed, delays)
5	12	B	timeline: order of effects (with/after previous, speed, delays)
18			*total*
4	2	A	effect: confusing "add" and "change"
1	13	A	effect: confusing "add" and "change"
5			*total*
1	1	A	effect: forgot location or operation of disappear
2	5	A	effect: forgot location or operation of disappear
1	10	A	effect: forgot location or operation of disappear
4			*total*
1	10	B	effect: general trouble adding an effect
3	12	B	effect: general trouble adding an effect
4			*total*
1	12	B	problems defining a curve
1	16	B	problems defining a curve
2			*total*
1	5	A	timeline: can i name my layers?
1	9	B	effect: add an effect to a grouped item without ungrouping?
2	12	B	how do I get the custom animation window back?
1	6	A	how to rotate objects?
1	10	A	effect: forgot location or operation of "fly out"
6			*total*
39			*PowerPoint grand total*

F.3.3 K-Sketch Bug Incidents

Incident Count	Participant	Task	Incident
1	2	A	looks like recording when alternate button held too late
1	2	B	looks like recording when alternate button held too late
2	12	B	looks like recording when alternate button held too late
1	13	B	looks like recording when alternate button held too late
5			*total*
1	1	A	timeline: can't move tic marks when selected
1	2	A	timeline: can't move tic marks when selected
1	11	A	timeline: can't move tic marks when selected
1	13	B	timeline: can't move tic marks when selected
4			*total*
1	8	A	performance slow
1	12	B	performance slow
2			*total*
1	8	A	timed change shrunk to zero duration (unpredictable)
1	11	A	timed change shrunk to zero duration (unpredictable)
2			*total*
1	12	B	mistake (not a bug, but i intervened thinking it was)
1	10	A	overwriting part of a motion speeds it up rather than truncating
2			*total*
15			*K-Sketch grand total*

F.3.4 PowerPoint Bug Incidents

Incident Count	Participant	Task	Incident
1	3	B	Objects disappear in preview animations
1	11	B	Objects disappear in preview animations
2			*total*
1	10	A	Two instant effects at same time, 2nd should "win" but doesn't
2	11	A	Two instant effects at same time, 2nd should "win" but doesn't
3			*total*
5			*PowerPoint grand total*

F.3.5 K-Sketch Task Incidents

Incident Count	Participant	Task	Incident
1	1	A	Using QuickTime player
1	2	B	Avoid getting dancers to move in correct order
2			*K-Sketch grand total*

F.3.6 PowerPoint Task Incidents

Incident Count	Participant	Task	Incident
1	1	B	Avoid getting dancers to move in correct order
1	2	B	Do not need to make perfect
1	8	B	How big do the letters need to be?
1	10	A	Don't need to worry about making gamma rays disappear
4			*K-Sketch grand total*

F.4 Other Observations

In addition to the incidents recorded in the previous section, I recorded two other pieces of information: the number of participants who preferred each type of mode switch and the number of participants who manipulated the timeline tic marks.

F.4.1 *Mode switch used*

Number of participants	Preferred Alternate button
10	handheld remote control button
3	tablet PC bezel button
3	alternated between handheld and bezel

F.4.2 *Use of Timeline Tic Marks*

Number of participants	Use of timeline tic marks
6	moved normal tic marks
2	didn't move normal tic marks, but tried to move selection tic marks

F.5 Organized Subjective Feedback

This section organizes the subjective feedback that appears in the raw written responses presented at the end of this appendix, noting the number of participants that made each comment.

F.5.1 *K-Sketch Likes*

Participants	Comment
6	liked pen/drawing or paper-like feel
5	simple or easy to learn
4	felt natural or intuitive
3	timeline
2	more control over timing
2	seeing things move simultaneously while animating
1	fun to use
1	visual nature of interface
1	easy selection/grouping
1	fewer steps
1	orient to path feature (practice task needed it)

F.5.2 K-Sketch Dislikes

Participants	Comment
11	needs drawing/editing tools
3	roughness of motion/drawings
2	use of both hands at once
1	needs import
1	needs ability to change speed (feature wasn't introduced in study)
1	context menu inaccessible when off screen
1	needs "standard movements"

F.5.3 PowerPoint Likes

Participants	Comment
4	similarity to known tools
4	more graphical tools
3	formality/precision of graphics
2	logical or structured
2	more control of timing
1	existence of ready-made effects

F.5.4 PowerPoint Dislikes

Participants	Comment
7	complicated, technical, "too many options," "high mental effort"
4	time consuming or tedious or too many steps
3	inflexible or "too structured" not "user-friendly"
2	having to plan ahead of time
2	timeline
1	poor drawing tools
1	hard to coordinate objects
1	no "layering" of images (as in Photoshop)
1	no naming of objects
1	loss of effects when grouping
1	absence of orient to path (practice task needed it)

F.6 Written Responses

This section lists written responses to short answer questions. The responses are organized by question, with all responses for a question numbered by participant number. If a participant does not appear in the list, it means that the participant did not respond to the question.

F.6.1 Demographic Survey

5) Briefly describe the subject matter and/or purpose of the animations you would like to create.

2. Like to animate my signature character—a cartoon seagull.
3. Some small cartoons just for fun or put on my website
4. No idea, probably just random tinkering.
5. Recreational/experimental animation of existing images
6. Art and stimulation. No real purpose.
7. Short clips/humor.
8. Cartoon character reading a book.
9. Probably flash for my company's interactive website…we market to elementary aged children & their parents.
10. Something showing an animated form (stick figures, moving shapes) to enliven my website.
12. The purpose of the animations I would like to create is to provoke a thought or feeling that the individual wouldn't necessarily have in a regular period of time. I want people to think outside of themselves.
13. Animations for web and video.
14. I think simply making something move across the screen would make me happy (like a dog walking…).
15. User interface.
16. (1) "Eye candy" for the purpose of making research presentation more interesting. This is to (A) keep/focus the attention of the audience (B) communicate difficult concepts quickly and effectively (C) make a certain idea or proposal seem "impressive." (2) In my robotics work, I'd like to create an on-the-fly animator to run through multiple candidate trajectories on robotic arms. This is for the sake of debugging & developing good trajectories by means of safe & quick evaluation *and* for the sake of collision detection/avoidance.

F.6.2 Post-test Questionnaire

5) Is there anything you particularly *liked* about K-Sketch?

1. Easy to use. Fun in that it was like drawing on paper. Visually oriented. WYSIWYG. Good portable size.
2. I liked working on the Tablet PC because the sketching is much faster and closer to feel of hand drawing images.
3. The flow is more natural, easy to think of. Unlike in Power Point, users are expect to think more logically.
4. Simpler interface.
5. The pen interface is more intuitive than mouse. Also, visualizing the time bar was easier.
6. How you can see things as they move. More control. I like the pen strokes and the speed.
7. Swift learning curve, yet functional.
8. Being able to draw the figures, being able to move them however I liked, easy selection & grouping, easy addition of other figures & movements.
9. So easy to use…pretty intuitive. Much friendlier interface for users.
10. It was familiar (pen/paper).
11. The time slider was great—being able to see the motions at the same time I was drawing also was really nice.
12. I felt like I had more control over when things were happening over time.
13. Fewer steps, ability to have freehand drawings.
14. It's much easier to create the animations.
15. Ability to control position & rotation at the same time. Some more realistic motion since the path and rotation was done free hand.
16. The timeline! Also, as a whole it was very intuitive and the pen-based interaction made the large array of possible commands seem very workable (i.e., I didn't have to look through several menus & submenus to hunt for commands to do what I wanted. I could just *do* what I wanted.

6) Is there anything you particularly ***disliked*** about K-Sketch?

1. No. But I could see where an easel type stand might come in handy at times for desk work or some sort of pillow like backing for working on your lap in a vehicle/train (commuters) or even some sort of glove or handle on the back of the tablet so it could be held easily and securely while standing. I realize you're still working on the software, but more options (colors, line/eraser size, etc.). Would be nice to copy and repeat or copy and mirror (flip/reverse) the picture and/or the animated part to save time.
2. Two tools means working faster with both sides of the brain. Not enough editing tools to correct little errors. Would be easier animating simple objects that are scanned in or inserted from illustration programs.
3. A bit slow, no way for me to specify the speed and the time of the animation other than actually recording it. You know, sometimes I want it to go faster than my hand can draw. ☺
4. Motion paths being freeform were a bit rough at times.
5. Using both hands for selection.
6. Maybe a lack of tools.
7. A bit light on features/tools (compared to PowerPoint).
8. Not having the tools necessary to create standard looking shapes (ready-made circles, squares, etc.) & straight lines. Also, not being able to select standard movements.
9. Not having AutoShapes for cleaner looking drawings. Also no copy & paste for duplicating objects.
10. I was actually worried about scratching the screen. Nothing unique to K-Sketch but more to using the notebook screen interface itself.
11. Not being able to re-edit the look of the objects. No “power tools,” like aligning. Although I liked writing “boom,” I would like to be able to type, too.
12. There wasn’t anything obvious to dislike.
13. Not as precise, lack of smooth shapes/lines, inability to tweak.

14. Having the menu hang off the screen if your object was too low. Not having a "ruler" or "stencil" for circles or straight lines.
15. Too much drawing. It was hard to make straight lines when making animation paths. Couldn't edit paths or timing.
16. That I couldn't program a std. linear trajectory or w/constant speed, but had to rely on how well I could emulate that w/ the mouse.

7) Is there anything you particularly ***liked*** about PowerPoint?

1. Just seemed like another Microsoft product. It was nice to be able to use shortcut keys that are in every other MS product.
2. I like the accuracy of the tools and the similarity to Visio that I had learned for doing flow charts and some mechanical drafting. I'll experiment more with the version of PowerPoint I do have on our home PC. Thank you for showing me what this powerful program can do.
3. Very logic. More ready-built effect to use.
4. More structured.
5. More familiar commands. More options for image/text manipulation.
6. No.
7. Potential for complex animations w/good control over sequencing…
8. The standard shapes, lines, and movements from above. Also, easy grouping.
9. Many similar commands as Photoshop which I have familiarity with.
10. The copy/paste feature.
11. Just the "organization" features—align, group, etc.
12. Not particularly.
13. Crispness of animation—shapes, fonts, etc.
14. Very precise and organized.
15. Precise control: over timing, smooth movements, smooth curves.
16. In the end, the animations looked neat b.c. fonts & pure geometric figures were used. In K-Sketch, they were all sketches. As a result, I would prefer PPT for a professional/business presentation, but would certainly favor the

speed of K-Sketch for presentations to colleagues & students where polish & elegance of the results is not as important.

8) Is there anything you particularly ***disliked*** about PowerPoint?

1. I preferred the timeline bar that K-Sketch had along the bottom. Being a more visual person, I liked seeing where the events were.
2. Lack of layering of images as far as I could see from yesterday's experience.
3. Need a lot of mental effort (at least for the first few times to get use to it). Need to think logically. Not very flexible.
4. Almost too structured.
5. Complicated editing options. Inability to name shapes. Options appeared in many ways, but some were difficult to find.
6. Too many options for fewer results, too many sub-menus, default options.
7. …But at the cost of user-friendliness and ***time***.
8. Difficulty w/drawing figures (no free-hand), difficulty moving them & coordinating them w/other moving figures, the number of steps involved, having to plan out animation ahead of time.
9. Laborious and tedious.
10. The "technical" feel vs. K-Sketch. I felt a bit "dumb" compared to using K-Sketch (like I was entering the technical real of "programming" vs. "animation").
11. The animation features—very complex for what it does. You have to think more about the animation before doing the animation.
12. I didn't like the fact that it's time consuming. I didn't care for it as an animation tool.
13. Many steps, a box for a car.
14. Too many technical things to change. When grouping things, takes off the animation that might have been previously there.
15. Movement & rotation together did not look very good.

16. Time consuming. And the fact that there is such an array of commands & features buried in submenus. Also, very much disliked the fact there was no timeline one could scroll/drag through to preview *or* that one could not move frame-by-frame.

APPENDIX

G Laboratory Evaluation 2 Documents

This appendix contains most of the documents used in my second summative laboratory evaluation. The documents that do not appear here were identical to others that were used in the first summative laboratory evaluation (e.g., the recruiting poster). Some others (e.g., the K-Sketch User Guide) were so similar to previous documents that I present only those parts that changed. In all cases where a document is omitted in part or in full, I tell where to find the full document.

G.1 Recruiting Poster

The poster for the study was identical to the poster used for the first summative laboratory study. It appears in Section E.1.

G.2 Study Script

Materials

1 Study packet
2 Ball-point pens
1 Tutorials/Task Instructions
1 K-Sketch User Guide
1 The TAB Lite User Guide
1 Drink for participant
2 Clipboards
5 Sheets blank paper for participant notes
1 Info sheet for reimbursement

Procedure

21) Participant fills out Consent Form
22) Give copy of consent form
23) Participant fills out Demographics/Screening Form
24) Repeat twice (Once for K-Sketch, and once for the TAB Lite)
    a. Repeat twice (Once for each session)
        i. Start Video (mark Recording Start Time)
        ii. Set up animation tool (Check that the TAB Lite has pen width 2)
        iii. Repeat for each set of tasks
            1. Deliver verbal tutorial
            2. Complete Practice Tasks
            3. Repeat for each experimental task
                a. Show participant task animation
                b. Read task instructions aloud
                c. Observe as performed recording time/critical incidents
                d. Participant fills out Comfort-level/Post Task Questionnaire
        iv. Participant takes 10 minute break half way through
        v. Doodle time with tool (last session only)
        vi. Participant fills out Post-Study Questionnaire (last session only)
        vii. Videotaping consent form
        viii. Give copy of video consent form
        ix. Read Reimbursement information form and collect information.

G.3 Informed Consent

UNIVERSITY OF WASHINGTON CONSENT FORM

Animation Tool Comparison Study

Investigators:

James A. Landay | Associate Professor | Computer Science & Engineering
Telephone: (206) 685-9139 | e-mail*: landay@cs.washington.edu

Richard C. Davis | Graduate Student | Computer Science & Engineering
Telephone: (206) 992-9363 | e-mail*: rcdavis@cs.washington.edu

**Please note that we cannot ensure the confidentiality of information sent via e-mail.*

<u>Investigators' statement</u>

We are asking you to be in a research study. The purpose of this consent form is to give you the information you will need to help you decide whether or not to be in the study. Please read the form carefully. You may ask questions about the purpose of the research, what we would ask you to do, the possible risks and benefits, your rights as a volunteer, and anything else about the research or this form that is not clear. When all your questions have been answered, you can decide if you want to be in the study or not. This process is called 'informed consent.'

PURPOSE OF THE STUDY

We want to better understand how our experimental animation tool, "K-Sketch," compares with existing animation tools, such as the TAB Lite. The goal of this study is to compare animators' speed, mental workload, and satisfaction when producing animations with these tools.

PROCEDURES

If you choose to participate, we would like you to attend four sessions of about three hours each. In each session, we would like you to fill out some questionnaires, practice using an animation tool, and to create 4 or 5 animations using these tools while we observe and time you. You will be allowed to take breaks so you don't become too tired.

The first questionnaire will briefly ask you questions about your background, such as your age and occupation. We will also ask you about your drawing skill and your interest in animation. You do not have to answer every question.

Next, we would like you to create 4 or 5 animations using K-Sketch or the TAB Lite. We'll train you to use each tool before you do the animation tasks. We'll time these tasks, and observe you as you perform them. After each timed task, we'd like you to fill out another questionnaire describing your experience. You will get a 10 minute break halfway through each session. You may opt out of any request at any time.

Finally, we will ask you to compare your experience using the two programs.

We would like to videotape you using these animation tools. Video taping is voluntary and will not affect the experiment. Only the research team will have access to the videotapes; you are welcome to view the video tapes if you wish, and you may ask us to destroy them at any time after the study. Videos will be destroyed on or before December 31, 2008. I will ask for your permission, on a separate form, to use video taped material in any publication.

RISKS, STRESS, OR DISCOMFORT

Some people feel that providing information for research is an invasion of privacy. Some people feel a little self-conscious when they are video taped. You may ask me to stop video taping at any time. Also, you may review the video and asked for it to be deleted.

BENEFITS OF THE STUDY

It is hoped that this research will benefit Human-Computer Interaction researchers in general and help us in particular to make K-Sketch better. You may not directly benefit from taking part in this study.

OTHER INFORMATION

Taking part in this research is voluntary. You can stop at any time. Information about you is confidential. We will link study information with your name using a study code so that we can keep records between sessions and so that you can withdraw any video recordings at any time. We will store this link in a locked filing cabinet separate from the data and destroy it on December 31, 2008. No one will be able to tell which animation is yours. The following groups may review study records for this project: University of California at Berkeley researchers; Institutional oversight review offices at the University of Washington, or the University of California Berkeley; and federal regulators. We may publish or present animation created during this study. If we publish or present the results of this study, we will not be able to use your name.

You will receive $50 worth of Amazon.com gift certificates for each session of this study that you participate in.

Signature of investigator Printed Name Date

<u>Subject's statement</u>

This study has been explained to me. I volunteer to take part in this research. I have had a chance to ask questions. I agree to let the researcher present my computer work publicly as described above in the consent form. If I have questions later on about the research I can ask one of the investigators listed above. If I have questions about my rights as a research subject, I can call the University of Washington Human Subjects Division at (206) 543-0098. I will receive a copy of this consent form.

Signature of subject Printed name Date

Copies to: Investigator's file
Subject

G.4 Demographic Survey

The demographic survey used in this study is nearly identical to the one used in the first summative laboratory evaluation (see Section E.4). There are only two differences. First, the top of this survey asks for participants' age, rather than simply asking if they are over 18 years old. Second, this survey presents new choices for question two. The new version follows:

2) When did you last create an animation? (Circle one)

less than one month ago	less than six months ago	less than two years ago	more than two years ago	never

G.5 Tutorial and Task Phases

This experiment progressed in phases. Each phase included instruction, practice tasks, and experimental tasks. The following document lists the topics covered and tasks used in each phase.

G.5.1 Phase 1

K-Sketch Points to Cover

1) Manual
2) Interface Parts
3) Drawing/Erasing/Cut-Copy-Paste
4) Undo/Redo
5) Selecting/Using the manipulator

6) Connection between drawing and Timeline
7) Animating (with widget)
8) Moving Two objects simultaneously
9) Timeline/Play controls/Editing Timeline tics
10) Overwriting old stuff, truncating
11) Selection of oddly shaped stuff

TAB Lite Points to Cover

1) Manual
2) Interface Parts
3) Drawing/Erasing/Cut-Copy-Paste
4) Undo/Redo
5) Selecting/Shift-Select/Using the manipulator
6) Connection between drawing and Timeline
7) Animating (in betweening)
8) Timeline/Play controls
9) Bad in-betweens

Practice Animations

1) A1-Moving Objects
2) A2-Particles Growing

Tasks

3) A-Cobalt
4) B-Man Watching Sun

G.5.2 Phase 2

K-Sketch Points to Cover

1) Rotations > 180 deg
2) Selection troubles
3) Moving time tics

4) Copying Objects

TAB Lite Points to Cover

1) Rotations > 180 deg
2) Re-in-betweening
3) Copying Objects
4) Copying Frames

Practice Animations

5) C1-Bicycle

Tasks

6) C-Gear Reduction

G.5.3 Phase 3

K-Sketch Points to Cover

1) Moving Center of Rotation
2) Copying Objects and Motions
3) Moving selected motions

TAB Lite Points to Cover

1) Onion Skinning
2) Swing

Practice Animations

7) D1-Robot Arm

Tasks

8) D-Ant Walking

G.5.4 Phase 3b

K-Sketch Points to Cover

1) Planning/Rehearsing
2) Speed Control
3) Fix dialog

TAB Lite Points to Cover

1) Planning

Practice Animations

8b) E1-Throw Catch

Tasks

10) E-The Trickster Mummy

G.5.5 Phase 4

K-Sketch Points to Cover

1) Overwriting vs. adding – correction window
2) Rotate Handle
3) Walking man, rolling wheel, stretching diamond

TAB Lite Points to Cover

1) Correlations between morphed objects
2) Incremental Work

Practice Animations

11) F1-Ship

Tasks

12) F-Lady With Torch Tumbling

G.5.6 Phase 5

K-Sketch Points to Cover

1) Record Drawings (warning)
2) Speed Control

TAB Lite Points to Cover

1) Correlations between morphed objects
2) Incremental Work

Practice Animations

13) G1-Ship With Cannon

Tasks

14) G-Plane Story

G.5.7 Phase 6

K-Sketch Points to Cover

1) Practice Simultaneous motions

TAB Lite Points to Cover

none

Practice Animations

15) H1-Flower Zoom

Tasks

16) H-Faerie Adventures

G.5.8 Phase 7

K-Sketch Points to Cover

1) Copying Motions

TAB Lite Points to Cover

none

Practice Animations

17) I1-Electrons

Tasks

18) I-Detection of Distant Planets

G.6 Task Instructions

In the following task, you will be shown an animation and asked to make a similar animation using a particular tool. Try to represent the important features of this animation as fast as you can. **You do not need to make this animation look perfect.** It is more important for you to work quickly than it is for you reproduce the objects or their motions precisely. However, please try to keep the sequence of events the same and the speed of moving objects within about 50% of what you see.

As you work, please "think aloud," that is, verbalize what you are thinking. You can ask any question about the task, but avoid asking for help using the tool. If you get stuck, however, please ask for assistance.

Notes (not read to participant):

1) If participant makes a mistake in interpretation of the task, do not intervene until they say that they are done. (This forces on use of tool editing capabilities.)

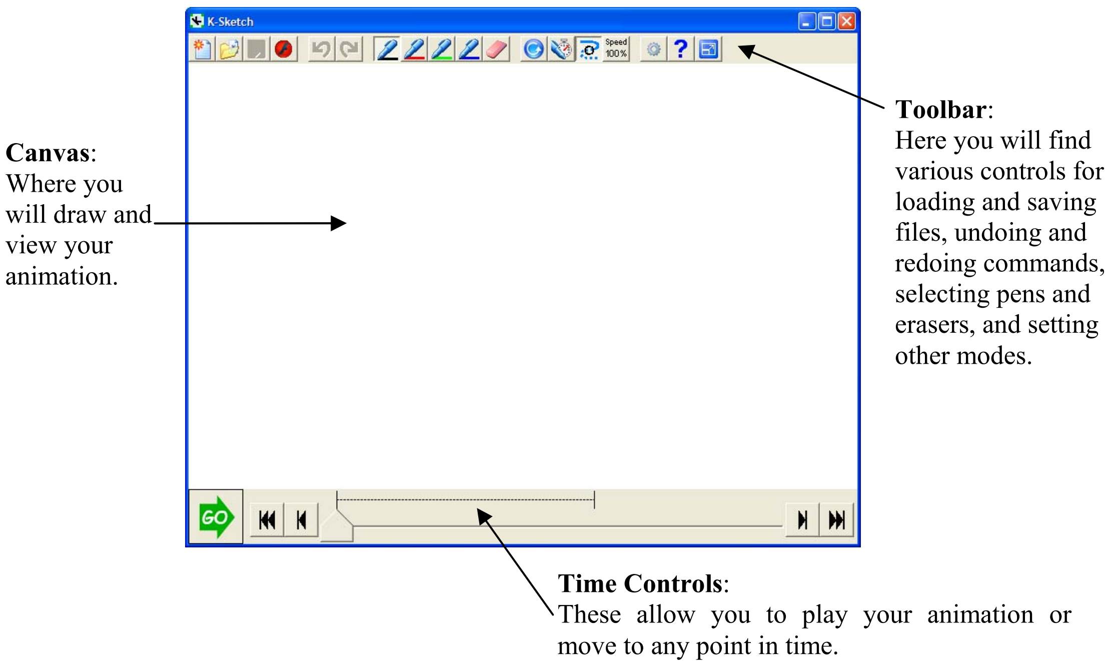
G.7 K-Sketch User Guide

Participants were allowed to refer to software user guides at any time during this study. The TAB Lite software came with a user guide (see http://www.tablite.com). The K-Sketch user guide was nearly identical to the practice task tutorial used in the first summative laboratory evaluation (see Section E.6). There were three differences. First, the opening pages were revised to describe the buttons that had been added since the previous tutorial was made. Second, the body of the previous tutorial (steps 1–8) was grouped under the heading "Tutorial 1 – The Car Jumped Over the Plane." Third, a second tutorial was added, entitled "Tutorial 2 – Adding vs. Overwriting Motions." The following sections present the new opening pages and the second tutorial. Refer to Section E.6 for the first tutorial.

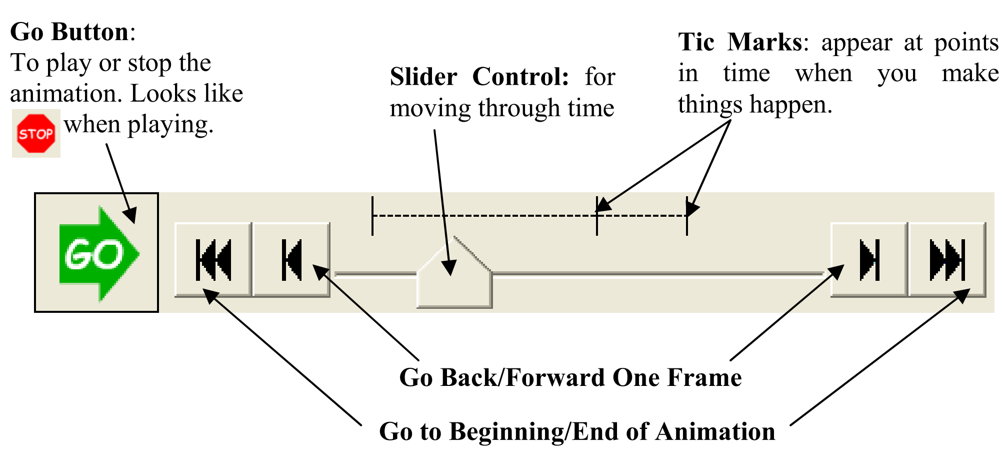
G.7.1 *Opening Pages*

This user guide gives a basic introduction to K-Sketch, the “kinetic” sketch pad for novice animators. Pages 1-3 give a quick reference for the various parts of the interface, and the remaining pages present tutorials that walk you through the animation process.

K-Sketch Overview



Time Controls

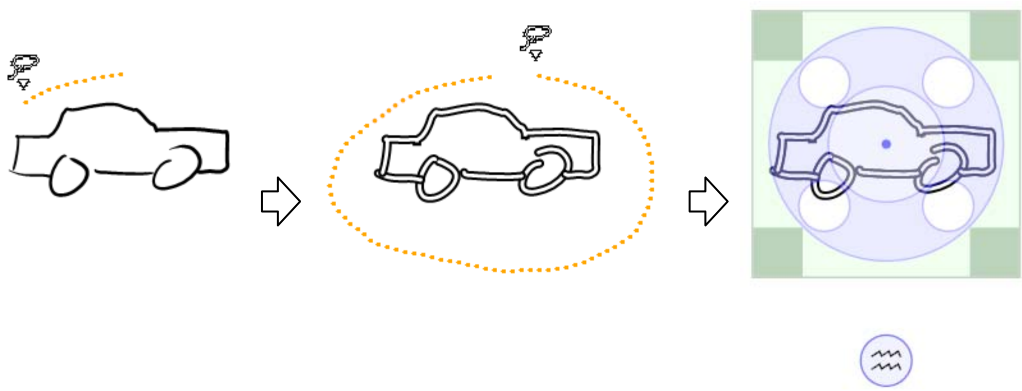


Selecting Objects

To select objects, you must hold down an "Alternate" button and draw a loop around them or tap on them. Any keyboard key (except TAB) will work as an Alternate button.

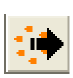
When using a TabletPC, we suggest finding a bezel button that is mapped to a keyboard key. Most Tablet PCs allow you to map these buttons to keys through a "Tablet Buttons" tab in the Control Panel's "Tablet and Pen Settings" dialog. Handheld remote controls often have buttons mapped to keyboard keys as well.

When you select an object, the Object Handle appears, as shown below.



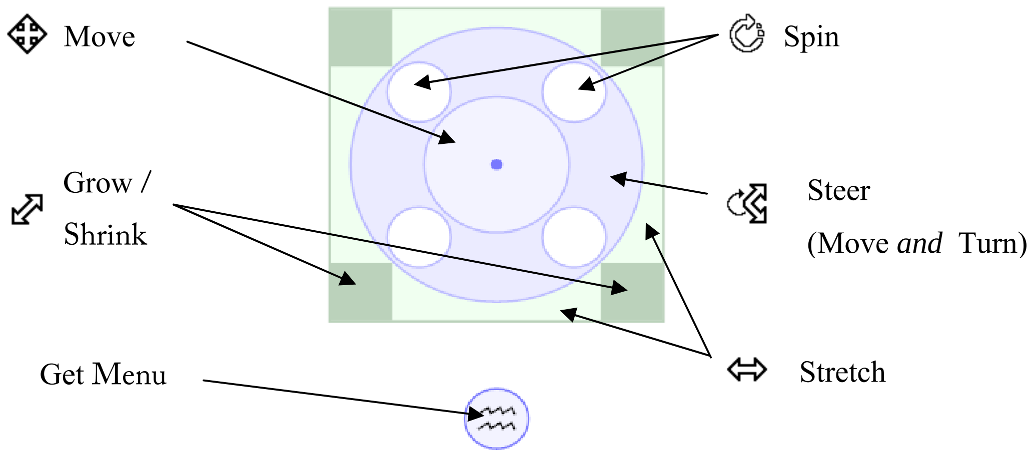
You can also tap on strokes to select objects. This will attempt to select strokes that move together.

Tapping on a motion path will select it. We suggest you use the "Select Next Guess" button described below to select motions, however, because this will make sure you get all of them.

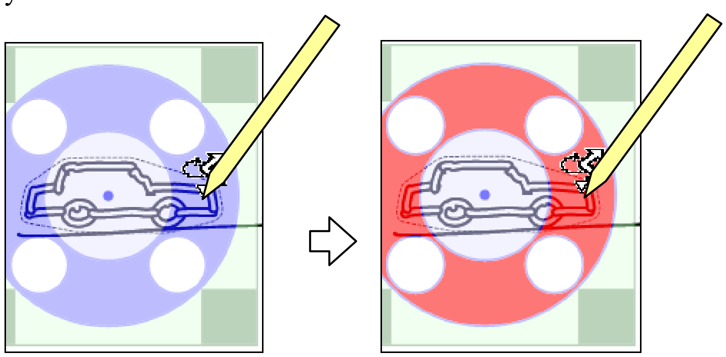
 You can use the "Select Next Guess" buttons shown here to cycle between groups of strokes as well as the motions applied to them. The first set of motions includes all of them, and later ones will be subsets of the motions.

Object Handle

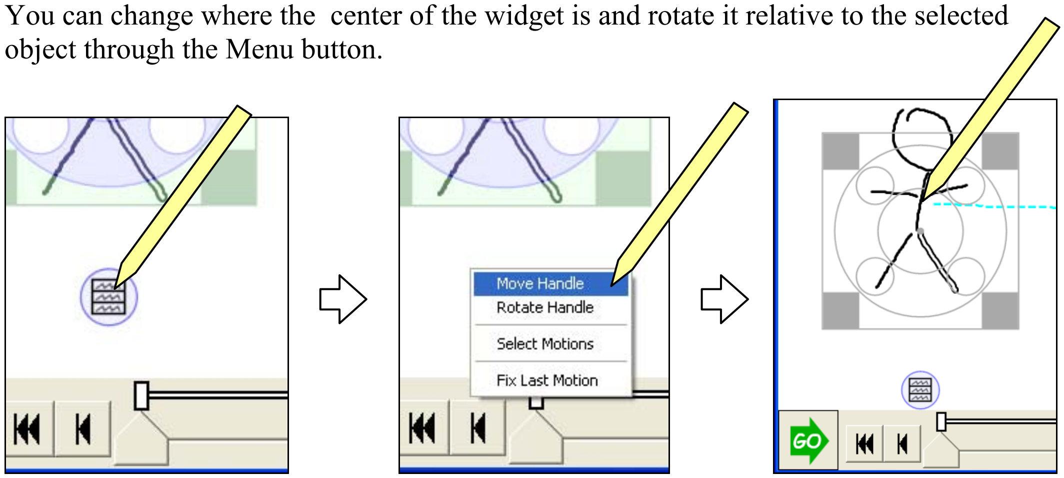
This control appears on top objects after you select them. It will move objects in various ways, depending on where you touch it with your pen.



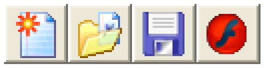
When you hold down the Alternate button, control regions turn red. This indicates that your next motion will be animated.



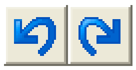
You can change where the center of the widget is and rotate it relative to the selected object through the Menu button.

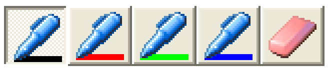


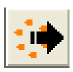
Toolbar Buttons

 **New Animation, Open Animation, Save Animation,** and **Export**

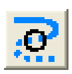
 **Cut, Copy,** or **Paste** selected objects or motions.

 **Undo** or **Redo** previous commands

 Choose **Pen Color** or **Eraser**

 **Select Next Guess:** Cycles through likely selections of objects and their motions.

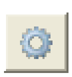
 **Fix Last Motion:** For adjusting how new motions overwrite or add to each other.

 **Show Motion Paths:** When pressed, the paths of animated movements are visible.

 **Record Drawing:** When pressed, every line you draw will be animated in time.

 **Loop Playback:** When pressed, the animation will repeat when it plays to the end.

Adjust Speed: Shows or hides a slider for making the animation slower or faster.

 **Options:** Show available configuration options.

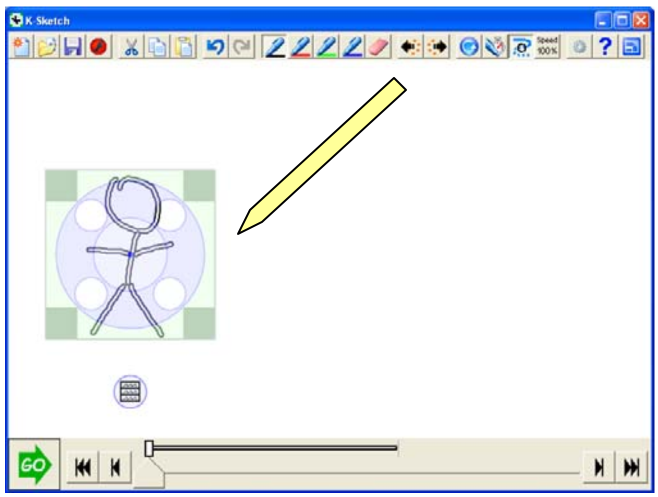
 **Help:** Show available help resources.

 **Full Screen Mode:** Toggle between filling the screen or one window only.

G.7.2 Tutorial 2 – Adding vs. Overwriting Motions

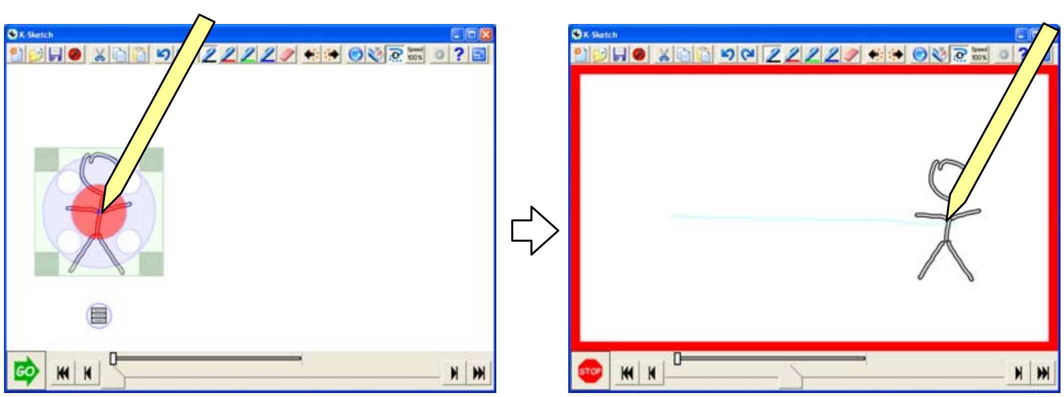
If you animate an object that is already moving, K-Sketch will attempt to guess whether the new motion should replace the old motion or be added on top of it. This tutorial will teach you how K-Sketch decides and tell you what you can do if K-Sketch guesses incorrectly.

Step 1: Draw and Select a Stick Figure



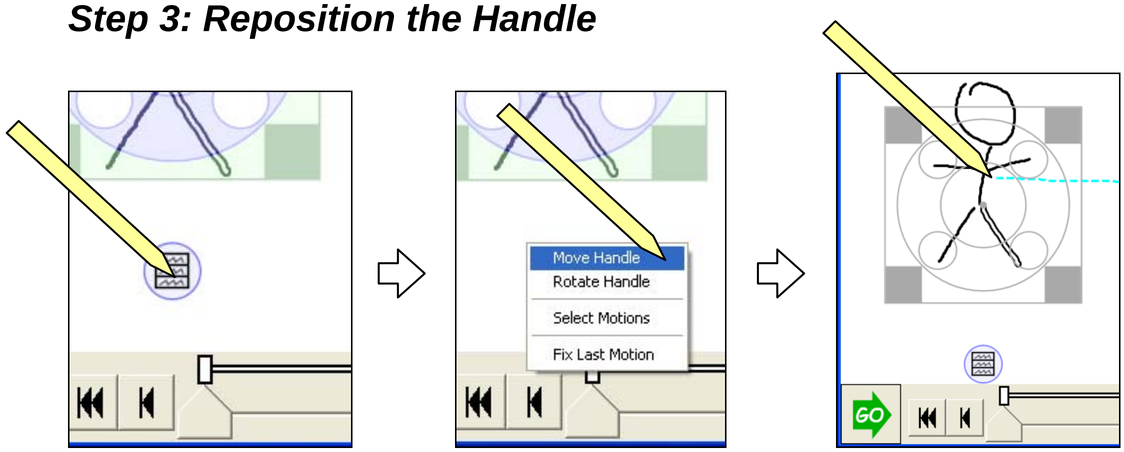
Start by drawing and selecting a stick figure. Make sure to use separate strokes for each leg.

Step 2: Move the Figure Across the Screen



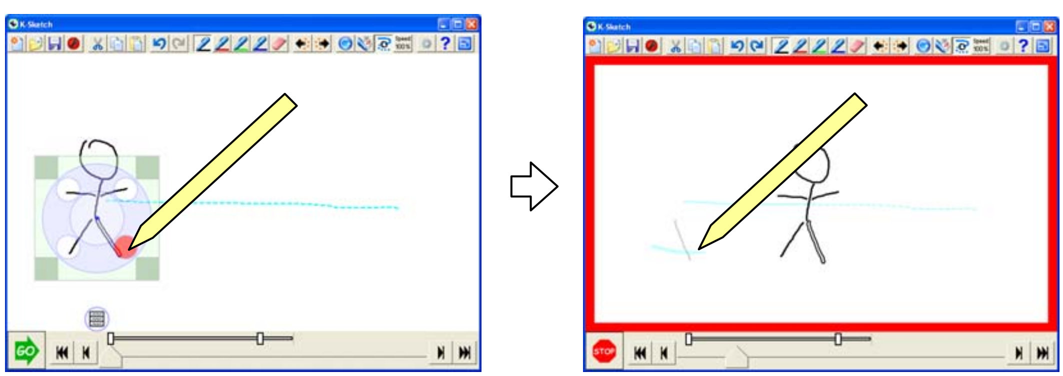
Make the figure run across the screen. Animate a movement to the right using the center (translate) region.

Step 3: Reposition the Handle



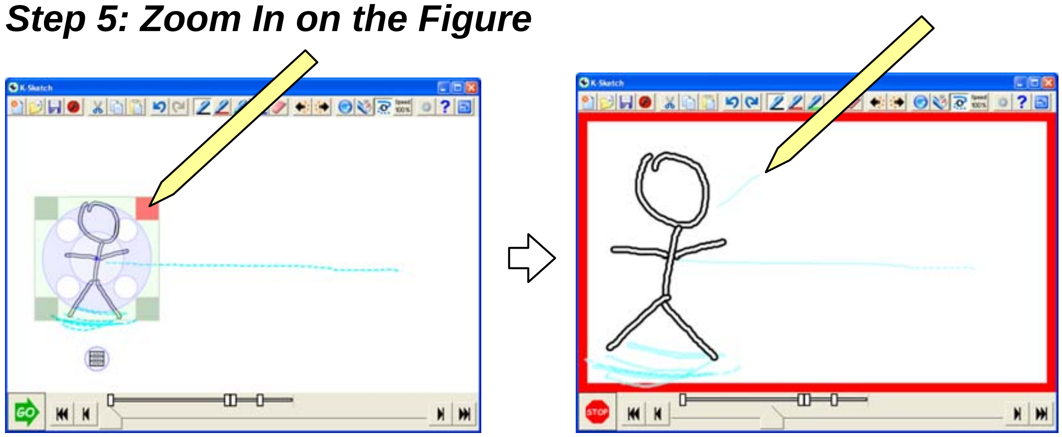
Our figure looks rather stiff as he runs. Rewind to the beginning of the animation and select one of his legs by drawing a loop around the leg only. The handle is in the same place, but you can tell that the rest of the body is not selected, because only the leg is drawn as an outline. Once you have done this, select “Move Handle” in the context menu and drag it until the center dot is at the top of the leg.

Step 4: Move the Figures Legs

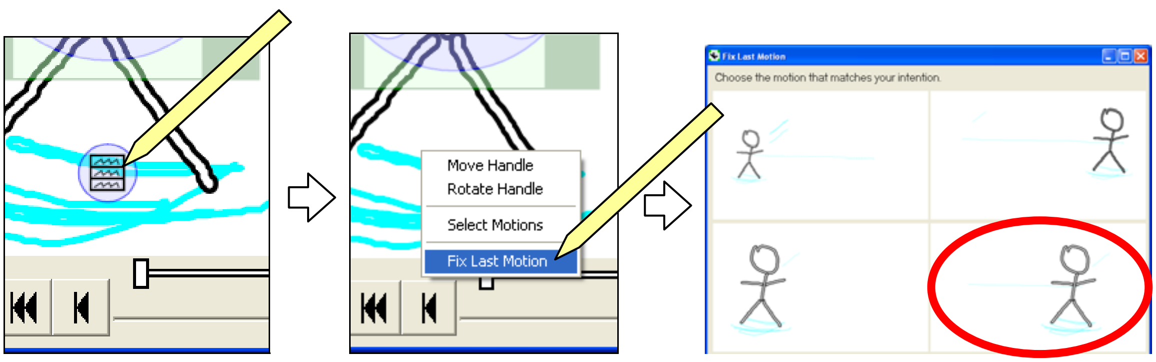


Now animate a rotation motion back and forth (using one of the four small disks on the “donut” part of the handle). Since you selected a part of a moving object, K-Sketch guessed that you wanted to add this rotation on top of fist movement. Repeat steps 3 and 4 for the other leg, and your figure will really look like he is running!

Step 5: Zoom In on the Figure

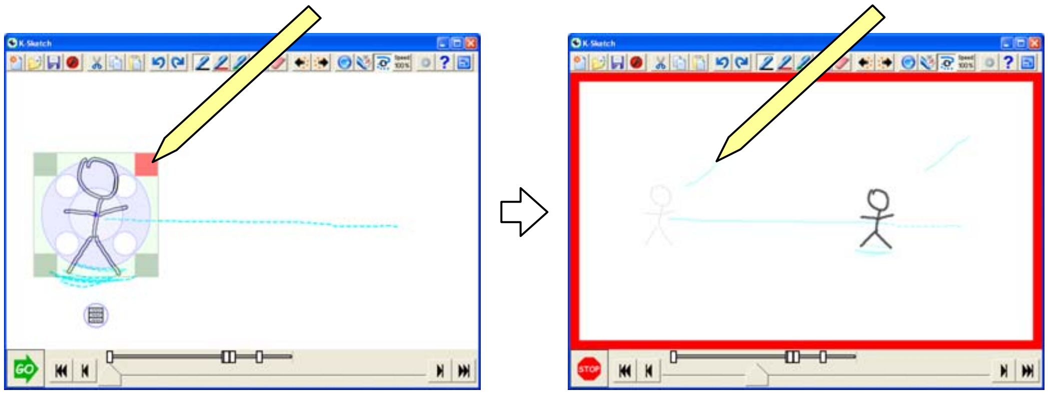


What if we tried to zoom in on our runner as he is running? Rewind to the beginning of the animation, select the runner, and animate a scale using one of the handle's four corners. Because we are moving an object that has already been moved, K-Sketch assumes that we wanted to overwrite the existing motion.



To get the motion we want, we need to "fix" the motion right after we do it. Select "Fix last motion" in the context menu to bring up a list of all the motions you might have wanted. The one you're looking for will usually be near the top of the list. Tap on it to select it. Then rewind and play the animation.

Step 6: Zoom Away From the Figure



Let's suppose that we later decided to zoom out from the runner rather than zooming in on him. Rewind to the beginning and re-animate the zoom in the opposite direction. This time we do not have to fix the motion. When K-Sketch overwrites motions, it will attempt to overwrite motions of the same type.

That's the final step! Rewind and play the full animation.

G.8 Comfort-level Questionnaire

	Extremely Unsatisfied			Neutral			Extremely Satisfied	Don't know
1) How satisfied were you with the animation you produced?	1	2	3	4	5	6	7	?

Please let us know how comfortable you would be *showing* the animation you just created to the following groups of people. Circle a response to each of the following.

	Extremely Uncomfortable			Neutral			Extremely Comfortable	Don't know
2) To a single colleague in a private meeting	1	2	3	4	5	6	7	?

3) To ten colleagues in a private meeting	1	2	3	4	5	6	7	?
4) To a student I am tutoring	1	2	3	4	5	6	7	?
5) To 30 students in a class I am teaching	1	2	3	4	5	6	7	?
6) To 300 students in a class I am teaching	1	2	3	4	5	6	7	?
7) To 30 professionals watching a business presentation I'm making	1	2	3	4	5	6	7	?
8) To 300 professionals watching a business presentation I'm making	1	2	3	4	5	6	7	?

Please let us know how comfortable you would be ***creating*** the animation as you just created it ***while being watched by*** the following groups of people. Circle a response to each of the following.

	Extremely Uncomfortable			Neutral			Extremely Comfortable	Don't know
9) Watched by a single colleague in a private meeting	1	2	3	4	5	6	7	?
10) Watched by ten colleagues in a private meeting	1	2	3	4	5	6	7	?
11) Watched by a student I am tutoring	1	2	3	4	5	6	7	?
12) Watched by 30 students in a class I am teaching	1	2	3	4	5	6	7	?
13) Watched by 300 students in a class I am teaching	1	2	3	4	5	6	7	?
14) Watched by 30 professionals during a business presentation I'm making	1	2	3	4	5	6	7	?
15) Watched by 300 professionals during a business presentation I'm making	1	2	3	4	5	6	7	?

G.9 Post-task Questionnaire

1) How long do you think it took to complete this task? (Draw a line across the timeline below.)

0 min.	5 min.	10 min.	15 min.	20 min.	25 min.	30 min.	35 min.	40 min.	45 min.	50 min.

	Very Easy			Neutral			Very Hard	Don't Know
2) How easy was this task?	1	2	3	4	5	6	7	?

	Very Fast			Neutral			Very Slow	Don't Know
3) How fast did you work?	1	2	3	4	5	6	7	?

4) Is there anything that made this task particularly ***easy***?

5) Is there anything that made this task particularly ***hard***?

6) Is there anything you particularly ***liked*** about this tool while completing this task?

7) Is there anything you particularly ***disliked*** about this tool while completing this task?

G.10 Post-study Questionnaire

Please circle a response indicating how much you agree or disagree with the following statements.

	Strongly Disagree			Neutral			Strongly Agree	Don't Know
1) I enjoyed using K-Sketch.	1	2	3	4	5	6	7	?
2) I enjoyed using the TAB Lite.	1	2	3	4	5	6	7	?

	Very Unlikely			Neutral			Very Likely	Don't Know
3) How likely are you to use K-Sketch again?	1	2	3	4	5	6	7	?

4) What factors influenced your answer to the preceding question?

	Very Unlikely			Neutral			Very Likely	Don't Know
5) How likely are you to use the TAB Lite again?	1	2	3	4	5	6	7	?

6) What factors influenced your answer to the preceding question?

7) Are there any comments you'd like to make about K-Sketch or you experience of using it during this study?

8) Are there any comments you'd like to make about the TAB Lite or you experience of using it during this study?

G.11 Video Taping Consent Form

This document was nearly identical to the version used in the first summative laboratory evaluation (see Section E.13). The only difference was that the type of recording, "Video Recordings made during study," was filled in before the study

APPENDIX

H Laboratory Evaluation 2 Data

H.1 Demographic data

#	Occupation	Age	Sex	Drawing skill (1 = none, 5 = excellent)	How often would you create animations?	Expertise (1 = complete beginner, 7 = expert) Computers	PowerPoint	Tablet PCs
1	grad student: technical comm.	22	F	2	not sure	5	5	1
2	law student	39	F	4	1/month	6	5	1
3	teacher, grad student: technical comm.	33	M	2	1/month	3	3	1
4	grad student: east Asia studies	37	M	3	not sure	4	4	2
5	undergrad student: technical comm./ linguistics	21	F	3	1/month	4	4	1
6	software developer, grad student: technical comm.	53	M	2	not sure	6	3	1
7	grad student: technical comm.	27	F	3	not sure	6	4	1
	Averages	*33.1*		*2.7*		*4.9*	*4.0*	*1.1*

Study 2 demographic data

#	Animation experience (hours): PowerPoint custom animation	PowerPoint other animation	Other animation tools	Animation tools used	Why not animate more? (1=Strongly Disagree, 7=Strongly Agree): Can access animation tools?	Complexity discourages?	Time required discourages?	Tool sequence
1	0	50	30	HyperCard	5	5	?	T,K
2	0	0	0		4	2	4	K,T
3	1	0	0		6	4	6	T,K
4	0	0	0		3	3	5	K,T
5	0	0	16	Keynote, Flash	5	6	6	T,K
6	0	0	0		2	?	?	K,T
7	30	30	0		4	6	5	K,T
Avg.	*4.43*	*11.43*	*6.57*		*4.14*	*4.33*	*5.20*	

Study 2 demographic data (continued) and tool sequence. PowerPoint "other" animation consisted mainly of animated slide transitions that I did not consider to be meaningful animation experience. Tool sequence legend: K = K-Sketch, T = The TAB Lite. PowerPoint.

H.2 Experimental Data

Task	Based on	Description	Operations used
A	39–*Cobalt*	Face with mouth opening while stars move behind	translate: straight, scale, appear, disappear
B	15–*Man watching sun*	Sun moves in arc as man turns to watch	appear, disappear, translate: any[k]
C	66–*Gear reduction*	Gears turn each other (8 times)	copy motion[k], rotate: any[k], interpolate[t], repeat motion
D	25–*Ant walking*	Ant swings his legs back and forth	rotate: < 180°, set timing[k], swing[t], move limb
E	33–*The trickster mummy*	Juggler coordinates tossing of two balls	rotate: < 180°, translate: any[k], physically simulate, move limb
F	26–*Lady with torch tumbling*	Tumbling figure holding a flickering torch	rotate: < 180°, set timing[k], move relative[k], orient to path[k], translate: any[k], interpolate[t], morph[t], deform[t], repeat motion, define cels
G	46–*Plane story*	Plane flies around and destroys another plane	translate: straight, rotate: < 180°, scale, appear, disappear, move relative[k], trace[k]
H	30–*Faerie adventures*	Figure moves & flaps its wings, camera zooms	rotate: < 180°, scale, set timing[k], move relative[k], translate: any[k], repeat motion
I	51–*Detection of distant planets*	Star emits waves as a planet revolves around it	appear, disappear, copy motion[k], translate: any[k], repeat motion

[k] This operation was provide by K-Sketch only.
[t] This operation was provided by the TAB Lite only.

Study 2 experimental tasks

		Participants completing	Participants using paper notes	Time to task incident (mean, in minutes)	Full task time (mean, in minutes)	How fast? (mean, 1–7, 1=fastest)	How easy? (mean, 1–7, 1=easiest)	Comfort showing (mean, 1–7)	Comfort creating (mean, 1-7)	Help incidents (total)	Bug incidents (total)	Task incidents (total)
Task A (face)	*KSK*	7	0	3.22	3.22	3.00	2.57	5.29	5.69	0	0	0
	TAB	7	0	4.64	4.64	3.43	3.14	5.32	5.24	1	0	0
Task B (sun)	*KSK*	7	1	**4.90**	**4.90**	3.17	2.86	5.55	5.41	0	0	0
	TAB	7	1	**12.41**	**14.14**	4.17	4.50	4.43	4.22	2	0	2
Task C (gears)	*KSK*	7	1	**5.96**	**6.43**	4.00	4.57	3.66	4.18	3	1	2
	TAB	7	1	**13.49**	**16.86**	4.86	4.86	4.41	4.10	9	5	4
Task D (ant)	*KSK*	7	0	6.64	6.64	4.00	3.71	5.18	5.02	3	1	0
	TAB	7	1	5.45	5.45	3.71	3.14	5.20	5.08	1	0	0
Task E (juggle)	*KSK*	7	1	**11.30**	18.67	5.43	5.43	4.45	3.96	6	3	9
	TAB	7	6	**22.55**	23.42	5.14	5.14	5.36	4.51	0	1	2
Task F (torch)	*KSK*	7	0	**5.01**	**5.21**	3.33	2.33	5.90	5.50	0	3	2
	TAB	6	5	**17.88**	**17.95**	4.50	4.17	4.28	3.76	0	0	1
Task G (plane)	*KSK*	7	0	18.59	19.72	4.80	4.80	6.14	5.17	8	5	4
	TAB	5	4	24.80	24.67	4.40	4.70	6.13	5.14	0	0	1
Task H (faerie)	*KSK*	6	0	**7.83**	7.85	3.00	2.67	6.00	5.33	0	6	0
	TAB	3	2	**14.90**	14.70	3.33	2.67	6.08	5.29	0	0	0
Task I (waves)	*KSK*	5	1	7.64	7.74	2.50	2.50	6.19	6.21	0	0	2
	TAB	2	1	15.50	16.29	3.00	2.50	6.94	6.50	0	0	1

Study 2 summary results. Statistically significant differences are shown in bold.

Time to First Task Incident

		P1	P2	P3	P4	P5	P6	P7	Mean[a]
Task A	*KSK*	3.17	5.18	5.75	4.15	1.40	2.95	2.22	3.22
(face)	*TAB*	3.13	4.93	9.47	11.25	2.20	5.18	2.45	4.64
Task B	*KSK*	4.08	4.10	8.10	5.00	2.52	7.28	5.42	**4.90**
(sun)	*TAB*	9.62	13.57	*14.13*	21.03	*6.40*	13.78	13.27	**12.41**
Task C	*KSK*	4.05	8.13	13.65	8.23	2.48	*5.50*	*5.25*	**5.96**
(gears)	*TAB*	5.22	28.28	30.77	*41.70*	6.20	10.97	*6.30*	**13.49**
Task D	*KSK*	4.77	6.67	5.60	7.32	3.43	15.97	7.95	6.64
(ant)	*TAB*	4.35	4.68	4.78	15.17	2.52	9.43	4.07	5.45
Task E	*KSK*	7.43	*10.10*	9.95	23.52	*5.37*	*13.53*	*18.48*	**11.30**
(juggle)	*TAB*	11.05	23.83	27.00	36.65	14.70	32.75	*23.60*	**22.55**
Task F	*KSK*	4.55	7.05	*6.15*	6.93	2.07	5.58	*5.00*	**5.01**
(torch)	*TAB*	18.35	13.90	28.70		8.17	20.37	*16.40*	**17.88**
Task G	*KSK*	11.15	*11.65*	*15.35*	*40.68*	10.98	30.15	28.57	18.59
(plane)	*TAB*	22.50	*25.82*			9.68	32.05	23.78	24.80
Task H	*KSK*	6.15	7.98	15.87	8.95	2.58		10.23	**7.83**
(faerie)	*TAB*	14.95				4.78		15.83	**14.90**
Task I	*KSK*	4.82	9.27	7.45		3.28		*11.72*	7.64
(waves)	*TAB*	10.72				*7.78*			15.50

[a] These means are the inverse log of means computed through a regression-based statistical analysis of log time and may not exactly match a directly computed mean. Standard deviations are not shown, because their units are hard to interpret.

Statistically significant differences are in bold.

Full Time

		P1	P2	P3	P4	P5	P6	P7	Mean[a]
Task A	*KSK*	3.17	5.18	5.75	4.15	1.40	2.95	2.22	3.22
(face)	*TAB*	3.13	4.93	9.47	11.25	2.20	5.18	2.45	4.64
Task B	*KSK*	4.08	4.10	8.10	5.00	2.52	7.28	5.42	**4.90**
(sun)	*TAB*	9.62	13.57	*25.28*	21.03	*8.87*	13.78	13.27	**14.14**
Task C	*KSK*	4.05	8.13	13.65	8.23	2.48	*7.93*	*6.20*	**6.43**
(gears)	*TAB*	5.22	28.28	30.77	*56.45*	6.20	10.97	*22.22*	**16.86**
Task D	*KSK*	4.77	6.67	5.60	7.32	3.43	15.97	7.95	6.64
(ant)	*TAB*	4.35	4.68	4.78	15.17	2.52	9.43	4.07	5.45
Task E	*KSK*	7.43	*28.30*	9.95	23.52	*17.58*	*38.95*	*23.33*	18.67
(juggle)	*TAB*	11.05	23.83	27.00	36.65	14.70	32.75	*30.73*	23.42
Task F	*KSK*	4.55	7.05	*6.70*	6.93	2.07	5.58	*6.02*	**5.21**
(torch)	*TAB*	18.35	13.90	28.70		8.17	20.37	*17.72*	**17.95**
Task G	*KSK*	11.15	*13.92*	*18.82*	*41.97*	10.98	30.15	28.57	19.72
(plane)	*TAB*	22.50	*27.07*			9.68	32.05	23.78	24.67
Task H	*KSK*	6.15	7.98	15.87	8.95	2.58		10.23	7.85
(faerie)	*TAB*	14.95				4.78		15.83	14.70
Task I	*KSK*	4.82	9.27	7.45		3.28		*12.92*	7.74
(waves)	*TAB*	10.72				*8.35*			16.29

[a] These means are the inverse log of means computed through a regression-based statistical analysis of log time and may not exactly match a directly computed mean. Standard deviations are not shown, because their units are hard to interpret.

How Fast?

		P1	P2	P3	P4	P5	P6	P7	Mean	St.Dev.
Task A (face)	*KSK*	1	4	4	4	1	4	3	3.00	1.41
	TAB	2	6	4	5	2	3	2	3.43	1.62
Task B (sun)	*KSK*	1	5	4	4	2	4	4	3.17	1.33
	TAB	4	-[a]	4	4	4	4	5	4.17	0.41
Task C (gears)	*KSK*	4	4	5	3	3	5	4	4.00	0.82
	TAB	3	6	6	6	3	4	6	4.86	1.46
Task D (ant)	*KSK*	2	6	5	4	3	5	3	4.00	1.41
	TAB	3	4	4	4	2	6	3	3.71	1.25
Task E (juggle)	*KSK*	5	7	3	5	5	7	6	5.43	1.40
	TAB	4	6	5	4	4	6	7	5.14	1.22
Task F (torch)	*KSK*	3	6	3	4	2	2	4	3.33	1.51
	TAB	4	7	5		2	4	5	4.50	1.64
Task G (plane)	*KSK*	4	6	4	6	3	5	6	4.80	1.30
	TAB	5	6			3	4	4	4.40	1.14
Task H (faerie)	*KSK*	3	5	5	4	2		4	3.00	1.00
	TAB	4				1		5	3.33	2.08
Task I (waves)	*KSK*	2	3	2		3		5	2.50	0.71
	TAB	3				3		3	3.00	0.00

[a] Participant 2 omitted this question.

How Easy?

		P1	P2	P3	P4	P5	P6	P7	Mean	St.Dev.
Task A (face)	*KSK*	1	5	3	4	1	2	2	2.57	1.51
	TAB	2	6	3	5	2	2	2	3.14	1.68
Task B (sun)	*KSK*	1	3	2	4	2	4	4	2.86	1.22
	TAB	5	6½	4	4	4	4	4	4.50	0.96
Task C (gears)	*KSK*	6	6	3	4	3	5	5	4.57	1.27
	TAB	3	7	6	6	3	3	6	4.86	1.77
Task D (ant)	*KSK*	2	6	2	4	2	6	4	3.71	1.80
	TAB	2	5	2	4	2	5	2	3.14	1.46
Task E (juggle)	*KSK*	5	7	3	5	4	7	7	5.43	1.62
	TAB	5	7	6	4	4	5	5	5.14	1.07
Task F (torch)	*KSK*	2	5	2	4	1	1	3	2.33	1.51
	TAB	5	5	4		2	4	5	4.17	1.17
Task G (plane)	*KSK*	3	6	5	6	3	6	6	4.80	1.64
	TAB	6	6			3	4½	4	4.70	1.30
Task H (faerie)	*KSK*	3	6	5	4	1		4	2.67	1.53
	TAB	4				1		3	2.67	1.53
Task I (waves)	*KSK*	2	5	2		3		5	2.50	0.71
	TAB	3				2			2.50	0.71

Number	Audience
1	None (how satisfied were you with the animation?)[a]
2	A single colleague in a private meeting
3	Ten colleagues in a private meeting
4	A student I am tutoring
5	30 students in a class I am teaching
6	300 students in a class I am teaching
7	30 professionals during a business presentation I'm making
8	300 professionals during a business presentation I'm making

[a] This audience was not used for comfort creating animations.

Audiences for comfort-level questionnaire

Comfort Showing Animations to Others

		P1	P2	P3	P4	P5	P6	P7	Mean	St.Dev.
Task A (face)	*KSK*	7.00	3.13	3.75	6.63	6.75	4.75	5.00	5.29	1.54
	TAB	6.13	4.38	3.38	6.63	6.25	6.00	4.50	5.32	1.23
Task B (sun)	*KSK*	6.50	4.63	3.75	6.63	7.00	6.00	4.38	5.55	1.28
	TAB	3.38	3.88	3.13	6.63	4.75	5.00	4.25	4.43	1.18
Task C (gears)	*KSK*	1.13	3.75	3.00	6.50	4.75	3.00	3.50	3.66	1.66
	TAB	2.75	4.63	2.25	6.50	5.75	6.00	3.00	4.41	1.74
Task D (ant)	*KSK*	7.00	4.75	3.63	6.63	6.00	4.00	4.25	5.18	1.35
	TAB	6.88	4.75	3.38	6.63	6.63	4.00	4.13	5.20	1.47
Task E (juggle)	*KSK*	6.13	2.63	4.13	6.50	4.00	4.00	3.75	4.45	1.37
	TAB	7.00	4.25	3.25	6.63	5.88	6.38	4.13	5.36	1.46
Task F (torch)	*KSK*	6.88	5.88	4.00	6.63	7.00	7.00	4.63	5.90	1.31
	TAB	5.25	3.75	3.00		5.88	4.81	3.00	4.28	1.21
Task G (plane)	*KSK*	7.00	6.38	3.50	6.38	6.75	5.94	4.63	6.14	0.94
	TAB	7.00	5.38			6.63	6.88	4.75	6.13	1.00
Task H (faerie)	*KSK*	6.75	3.25	3.13	6.63	6.13		5.13	6.00	0.82
	TAB	7.00				7.00		4.25	6.08	1.59
Task I (waves)	*KSK*	6.63	6.00	4.75		5.75		4.88	6.19	0.62
	TAB	7.00				6.88			6.94	0.09

Comfort Creating Animations in Front of Others

		P1	P2	P3	P4	P5	P6	P7	Mean	St.Dev.
Task A (face)	*KSK*	6.71	4.43	3.14	7.00	7.00	5.14	6.43	5.69	1.49
	TAB	5.86	3.57	3.00	7.00	5.86	6.00	5.43	5.24	1.43
Task B (sun)	*KSK*	6.57	3.71	3.29	7.00	7.00	6.00	4.29	5.41	1.60
	TAB	3.71	2.43	3.00	7.00	5.14	5.00	3.29	4.22	1.58
Task C (gears)	*KSK*	1.43	3.29	2.57	7.00	5.14	4.29	5.57	4.18	1.91
	TAB	4.86	1.43	2.29	7.00	4.86	6.00	2.29	4.10	2.12
Task D (ant)	*KSK*	6.86	4.00	3.43	7.00	5.00	4.00	4.86	5.02	1.41
	TAB	7.00	2.86	3.57	7.00	5.86	4.00	5.29	5.08	1.65
Task E (juggle)	*KSK*	6.29	2.00	3.71	7.00	3.71	3.00	2.00	3.96	1.97
	TAB	5.57	2.86	2.86	7.00	4.57	6.43	2.29	4.51	1.89
Task F (torch)	*KSK*	6.43	3.86	4.00	7.00	6.43	7.00	5.29	5.50	1.34
	TAB	4.57	2.71	2.57		5.14	5.00	2.57	3.76	1.27
Task G (plane)	*KSK*	6.57	3.14	3.71	7.00	6.71	6.00	3.43	5.17	1.74
	TAB	6.00	2.57			6.14	6.00	5.00	5.14	1.51
Task H (faerie)	*KSK*	6.43	4.71	2.86	7.00	6.00		3.57	5.33	1.54
	TAB	5.86				6.43		3.57	5.29	1.51
Task I (waves)	*KSK*	7.00	4.14	4.43		5.43		3.71	6.21	1.11
	TAB	6.00				7.00			6.50	0.71

			Comfort showing by audience (1 = extremely uncomfortable, 7 = extremely comfortable)								Comfort creating by audience (1 = extremely uncomfortable, 7 = extremely comfortable)							Time estimate (minutes)
Participant	Tool	Task	A1	A2	A3	A4	A5	A6	A7	A8	A2	A3	A4	A5	A6	A7	A8	
1	K-Sketch	A	7	7	7	7	7	7	7	7	7	7	7	7	6	7	6	3
2	K-Sketch	A	3	6	5	4	3	2	1	1	6	5	6	4	4	3	3	4
3	K-Sketch	A	5	5	4	5	4	3	2	2	5	3	5	3	2	2	2	15
4	K-Sketch	A	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	15
5	K-Sketch	A	7	7	7	7	7	7	6	6	7	7	7	7	7	7	7	1
6	K-Sketch	A	6	5	5	5	5	5	4	3	6	6	6	5	5	4	4	5
7	K-Sketch	A	6	7	5	7	5	4	3	3	7	7	7	7	7	5	5	5
1	TAB Lite	A	6	7	6	6	6	6	6	6	7	6	7	6	5	5	5	4
2	TAB Lite	A	4	7	6	6	4	3	3	2	6	5	5	3	2	2	2	4
3	TAB Lite	A	5	6	3	5	2	2	2	2	5	3	5	2	2	2	2	8
4	TAB Lite	A	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	13.5
5	TAB Lite	A	6	7	7	6	7	7	5	5	6	6	7	6	6	5	5	3
6	TAB Lite	A	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	9.5
7	TAB Lite	A	6	6	6	6	4	3	3	2	7	7	6	6	5	4	3	4
1	K-Sketch	B	6	7	7	7	7	6	6	6	7	7	7	6	6	7	6	3.25
2	K-Sketch	B	5	7	6	6	4	3	3	3	6	5	5	3	3	2	2	4
3	K-Sketch	B	5	5	4	5	4	3	2	2	5	4	5	3	2	2	2	15
4	K-Sketch	B	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	10
5	K-Sketch	B	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	3
6	K-Sketch	B	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	7.75
7	K-Sketch	B	5	5	4	6	5	3	4	3	6	5	5	4	3	4	3	10
1	TAB Lite	B	3	5	5	4	3	3	2	2	6	5	5	4	3	2	1	10
2	TAB Lite	B	5	5	5	4	4	3	3	2	5	4	3	2	1	1	1	10
3	TAB Lite	B	4	5	3	5	2	2	2	2	5	3	5	2	2	2	2	20
4	TAB Lite	B	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	19
5	TAB Lite	B	5	5	5	5	5	5	4	4	5	5	6	5	5	5	5	8
6	TAB Lite	B	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	16.5
7	TAB Lite	B	6	6	5	5	4	3	3	2	6	5	5	3	2	1	1	10
1	K-Sketch	C	1	2	1	1	1	1	1	1	3	1	2	1	1	1	1	4.75
2	K-Sketch	C	2	4	4	4	4	4	4	4	6	4	5	2	2	2	2	7
3	K-Sketch	C	4	4	3	4	3	2	2	2	4	2	4	2	2	2	2	25
4	K-Sketch	C	3	7	7	7	7	7	7	7	7	7	7	7	7	7	7	17.5
5	K-Sketch	C	5	5	5	6	5	4	4	4	5	5	6	5	5	5	5	3
6	K-Sketch	C	4	4	3	3	3	3	2	2	5	5	5	5	4	3	3	11.75
7	K-Sketch	C	3	6	4	6	3	2	2	2	7	6	7	6	6	4	3	13
1	TAB Lite	C	2	6	4	3	3	2	1	1	7	6	6	5	4	4	2	9.25
2	TAB Lite	C	5	5	5	5	5	5	4	3	2	2	2	1	1	1	1	20
3	TAB Lite	C	2	3	2	3	2	2	2	2	3	2	3	2	2	2	2	40
4	TAB Lite	C	3	7	7	7	7	7	7	7	7	7	7	7	7	7	7	45
5	TAB Lite	C	5	6	6	7	6	6	5	5	5	5	6	5	5	4	4	6.75

(continued)

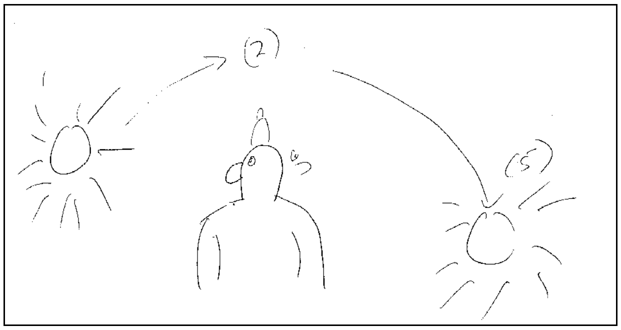
			Comfort showing by audience (1 = extremely uncomfortable, 7 = extremely comfortable)								Comfort creating by audience (1 = extremely uncomfortable, 7 = extremely comfortable)							Time estimate (minutes)
Participant	Tool	Task	A1	A2	A3	A4	A5	A6	A7	A8	A2	A3	A4	A5	A6	A7	A8	
6	TAB Lite	C	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	13
7	TAB Lite	C	4	6	4	4	3	1	1	1	5	3	4	1	1	1	1	20
1	K-Sketch	D	7	7	7	7	7	7	7	7	7	7	7	7	7	7	6	5
2	K-Sketch	D	4	6	5	5	5	5	4	4	4	4	4	4	4	4	4	7
3	K-Sketch	D	5	5	4	5	3	3	2	2	5	4	5	4	2	2	2	15
4	K-Sketch	D	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	18
5	K-Sketch	D	6	6	6	6	6	6	6	6	5	5	5	5	5	5	5	3
6	K-Sketch	D	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	17
7	K-Sketch	D	6	6	4	5	5	3	3	2	7	6	6	5	3	4	3	14
1	TAB Lite	D	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	9
2	TAB Lite	D	6	6	6	6	5	4	3	2	4	4	4	3	2	2	1	7
3	TAB Lite	D	4	5	4	5	3	2	2	2	5	4	5	4	3	2	2	10
4	TAB Lite	D	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	19
5	TAB Lite	D	6	7	7	7	7	7	6	6	6	6	7	6	6	5	5	4
6	TAB Lite	D	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	17.5
7	TAB Lite	D	6	6	5	5	4	3	2	2	7	6	6	6	5	4	3	13
1	K-Sketch	E	5	7	7	7	6	6	6	5	7	7	7	6	6	6	5	9
2	K-Sketch	E	5	5	4	3	1	1	1	1	2	2	2	2	2	2	2	10
3	K-Sketch	E	6	5	4	5	4	3	3	3	5	5	5	4	3	2	2	15
4	K-Sketch	E	3	7	7	7	7	7	7	7	7	7	7	7	7	7	7	23
5	K-Sketch	E	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	12
6	K-Sketch	E	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	18.25
7	K-Sketch	E	6	6	3	4	4	3	2	2	5	2	2	2	1	1	1	23
1	TAB Lite	E	7	7	7	7	7	7	7	7	7	6	6	6	5	5	4	11.25
2	TAB Lite	E	5	5	5	5	4	4	3	3	4	4	4	3	2	2	1	13
3	TAB Lite	E	4	5	3	5	3	2	2	2	4	3	4	3	2	2	2	25
4	TAB Lite	E	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	39
5	TAB Lite	E	6	6	6	7	6	6	5	5	5	5	6	4	4	4	4	10
6	TAB Lite	E	6	7	7	7	6	6	6	6	7	7	7	6	6	6	6	18.5
7	TAB Lite	E	5	6	5	5	4	3	3	2	5	3	3	2	1	1	1	35
1	K-Sketch	F	6	7	7	7	7	7	7	7	7	7	7	6	6	6	6	3.5
2	K-Sketch	F	3	7	7	7	6	6	6	5	5	5	4	4	3	3	3	8
3	K-Sketch	F	5	5	5	5	4	3	3	2	6	5	6	4	3	2	2	12.5
4	K-Sketch	F	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	13
5	K-Sketch	F	7	7	7	7	7	7	7	7	7	7	7	6	6	6	6	3
6	K-Sketch	F	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	8
7	K-Sketch	F	6	7	5	6	5	3	3	2	7	7	7	6	4	3	3	10
1	TAB Lite	F	5	6	5	6	5	5	5	5	7	5	6	4	3	4	3	15
2	TAB Lite	F	5	5	4	4	4	3	3	2	4	3	3	3	2	2	2	15
3	TAB Lite	F	4	4	3	4	3	2	2	2	4	2	4	2	2	2	2	30

(continued)

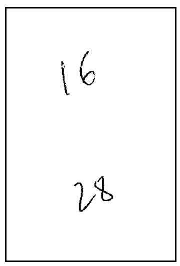
Participant	Tool	Task	Comfort showing by audience (1 = extremely uncomfortable, 7 = extremely comfortable) A1	A2	A3	A4	A5	A6	A7	A8	Comfort creating by audience (1 = extremely uncomfortable, 7 = extremely comfortable) A2	A3	A4	A5	A6	A7	A8	Time estimate (minutes)
5	TAB Lite	F	6	6	6	7	6	6	5	5	5	5	6	5	5	5	5	8
6	TAB Lite	F	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	18.5
7	TAB Lite	F	3	5	4	4	3	2	2	1	6	4	3	2	1	1	1	23
1	K-Sketch	G	7	7	7	7	7	7	7	7	7	7	7	7	6	6	6	12
2	K-Sketch	G	6	7	7	7	6	6	6	6	4	4	4	3	3	2	2	10
3	K-Sketch	G	5	5	3	5	3	3	2	2	5	4	5	4	3	3	2	22
4	K-Sketch	G	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	27
5	K-Sketch	G	7	7	7	7	7	7	6	6	7	7	7	7	7	6	6	8
6	K-Sketch	G	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	20
7	K-Sketch	G	5	6	5	6	4	4	4	3	6	4	6	3	2	2	1	20
1	TAB Lite	G	7	7	7	7	7	7	7	7	7	6	7	6	5	6	5	18
2	TAB Lite	G	6	7	6	6	5	5	4	4	4	3	3	3	2	2	1	16
5	TAB Lite	G	6	7	7	7	7	7	6	6	6	6	7	6	6	6	6	10
6	TAB Lite	G	6	7	7	7	7	7	7	7	6	6	6	6	6	6	6	18
7	TAB Lite	G	7	6	6	6	4	3	4	2	7	6	6	5	4	4	3	25
1	K-Sketch	H	6	7	7	7	7	7	7	6	7	7	7	6	6	6	6	7
2	K-Sketch	H	5	4	4	4	3	2	2	2	6	6	6	5	4	3	3	10
3	K-Sketch	H	4	4	3	4	3	3	2	2	4	3	4	3	2	2	2	25
4	K-Sketch	H	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	15
5	K-Sketch	H	6	6	6	7	6	6	6	6	6	6	6	6	6	6	6	3
7	K-Sketch	H	6	6	6	6	5	4	4	4	6	6	5	3	2	2	1	13
1	TAB Lite	H	7	7	7	7	7	7	7	7	7	6	7	6	5	5	5	20
5	TAB Lite	H	7	7	7	7	7	7	7	7	7	7	7	6	6	6	6	5
7	TAB Lite	H	6	6	5	5	4	3	3	2	6	5	5	3	2	2	2	15
1	K-Sketch	I	6	7	7	7	7	6	7	6	7	7	7	7	7	7	7	5
2	K-Sketch	I	6	7	7	7	6	5	5	5	5	5	5	4	4	3	3	12
3	K-Sketch	I	6	6	5	6	5	4	3	3	6	5	6	4	4	3	3	10
5	K-Sketch	I	3	6	6	7	6	6	6	6	6	6	6	5	5	5	5	4
7	K-Sketch	I	5	6	6	6	5	4	4	3	7	5	4	3	2	3	2	17
1	TAB Lite	I	7	7	7	7	7	7	7	7	7	6	7	6	5	6	5	11
5	TAB Lite	I	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

H.3 Sketches and Notes

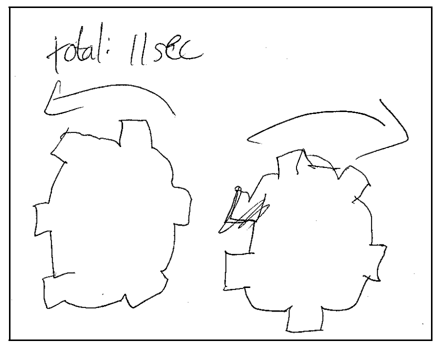
This section contains all sketches and notes made by participants during the second summative laboratory evaluation.



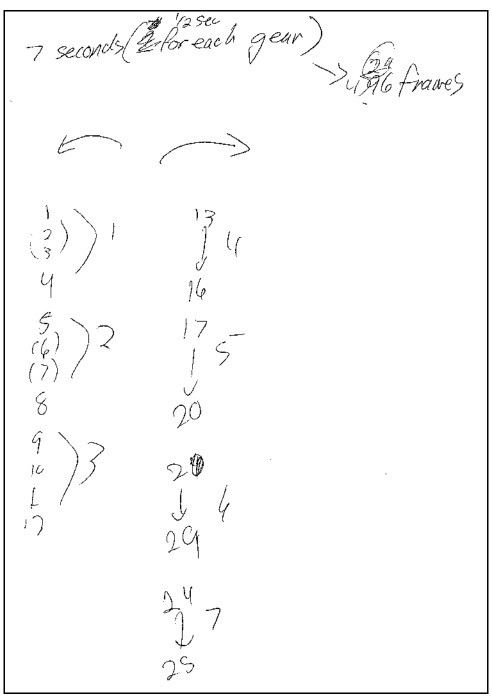
Task B – Participant 3 – K-Sketch



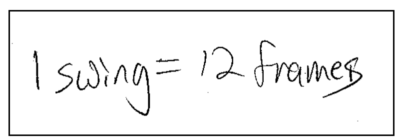
Task B – Participant 5 – The TAB Lite



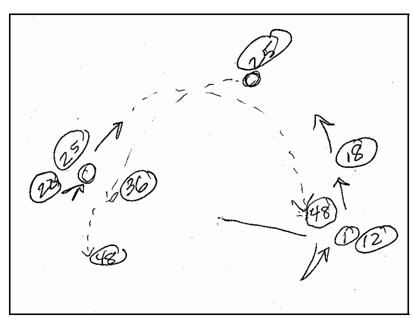
Task C – Participant 3 – K-Sketch

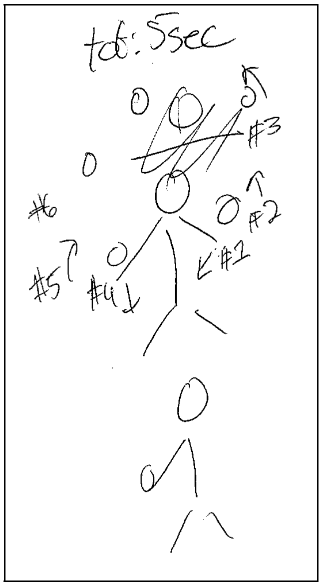


Task C – Participant 3 – The TAB Lite

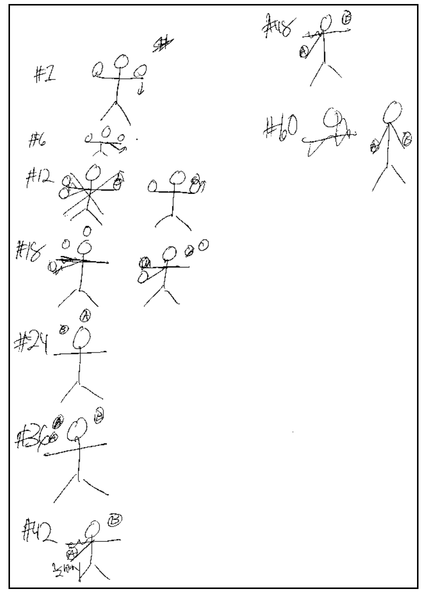


Task D – Participant 3 – The TAB Lite

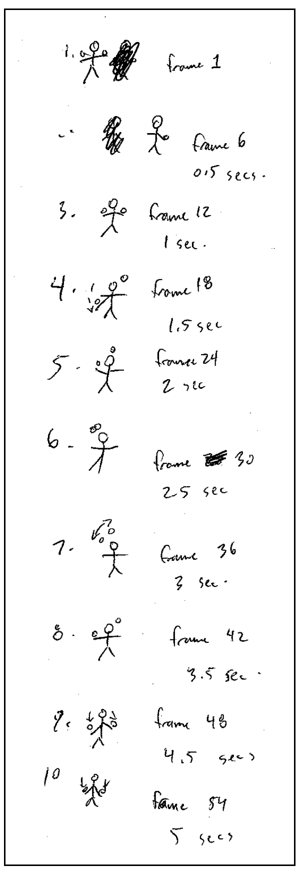

Task E – Participant 2 – The TAB Lite



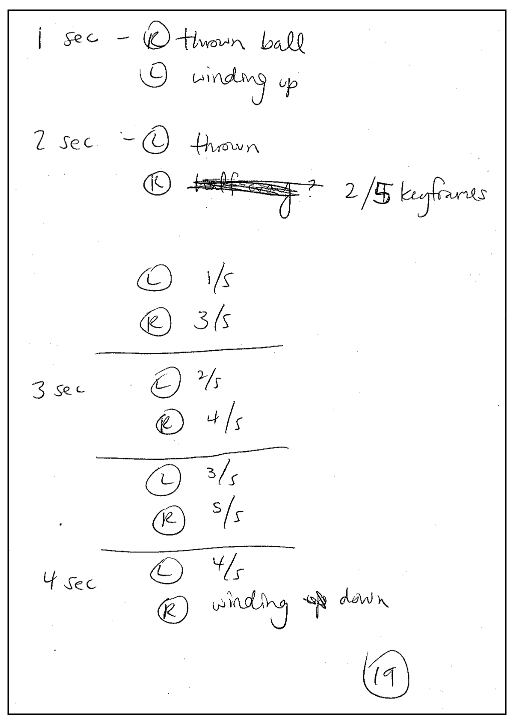
Task E – Participant 3 – K-Sketch



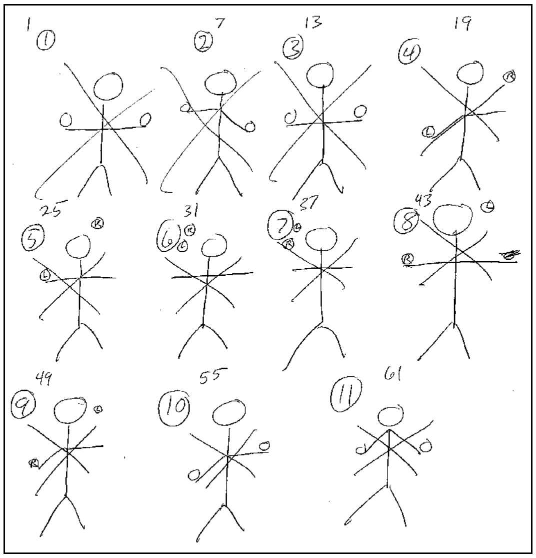
Task E – Participant 3 – The TAB Lite



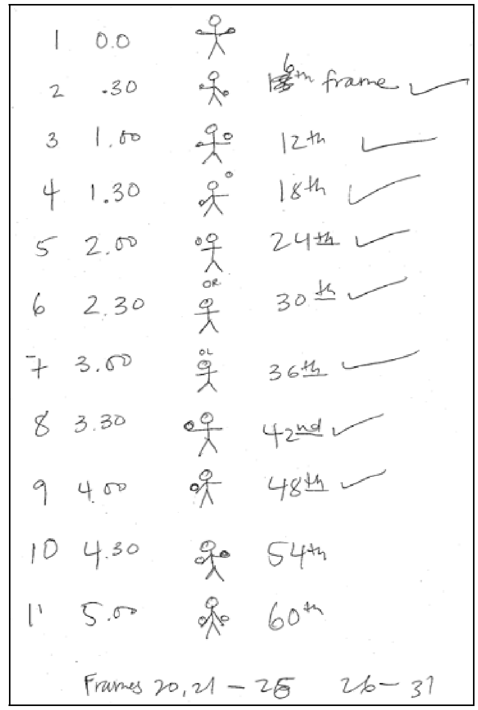
Task E – Participant 4 – The TAB Lite



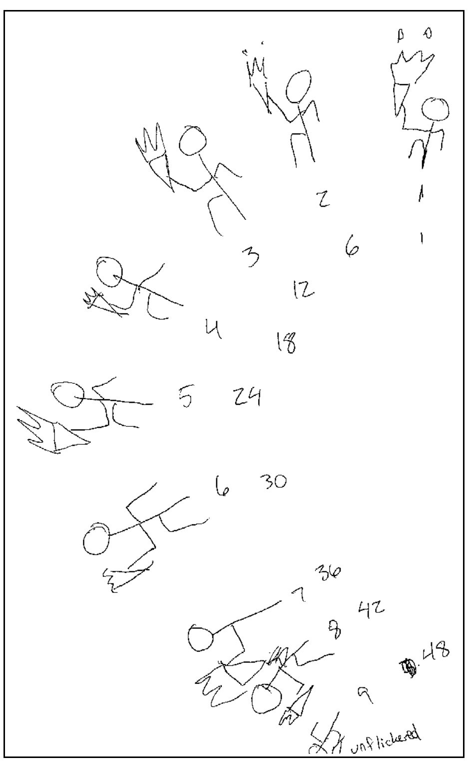
Task E – Participant 5 – The TAB Lite



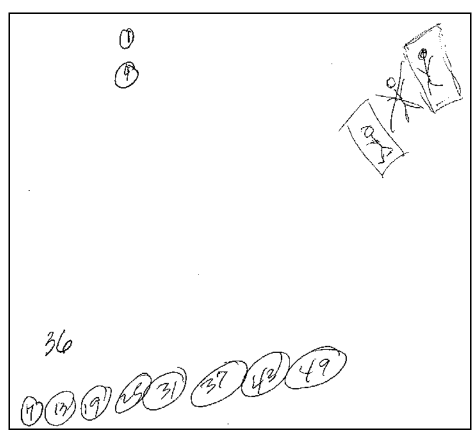
Task E – Participant 6 – The TAB Lite



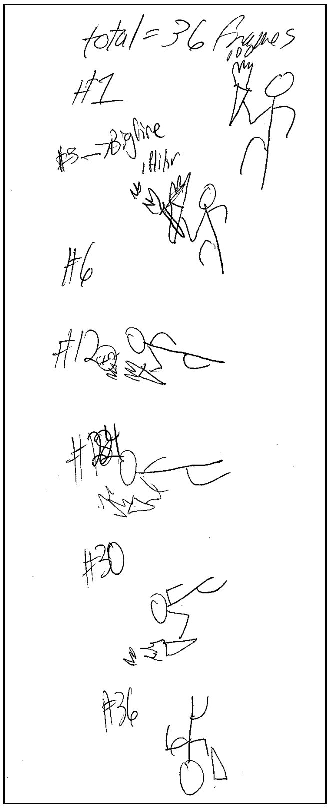
Task E – Participant 7 – The TAB Lite



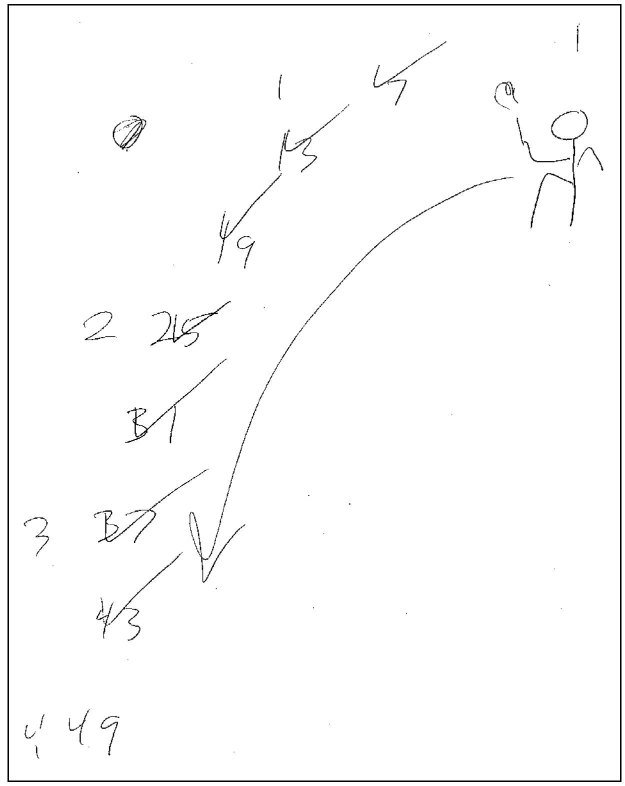
Task F – Participant 1 – The TAB Lite



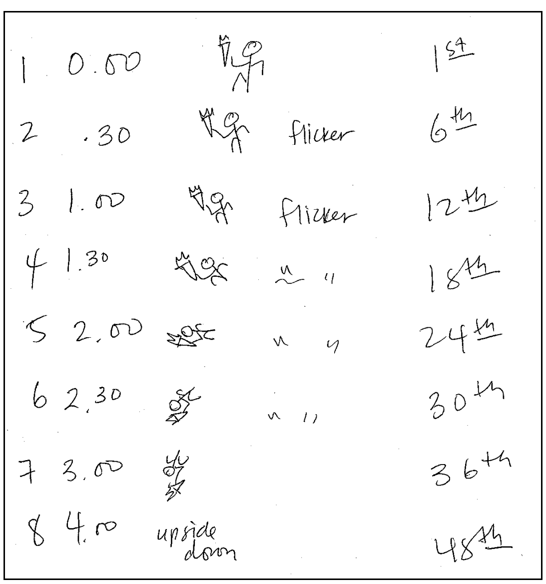
Task F – Participant 2 – The TAB Lite



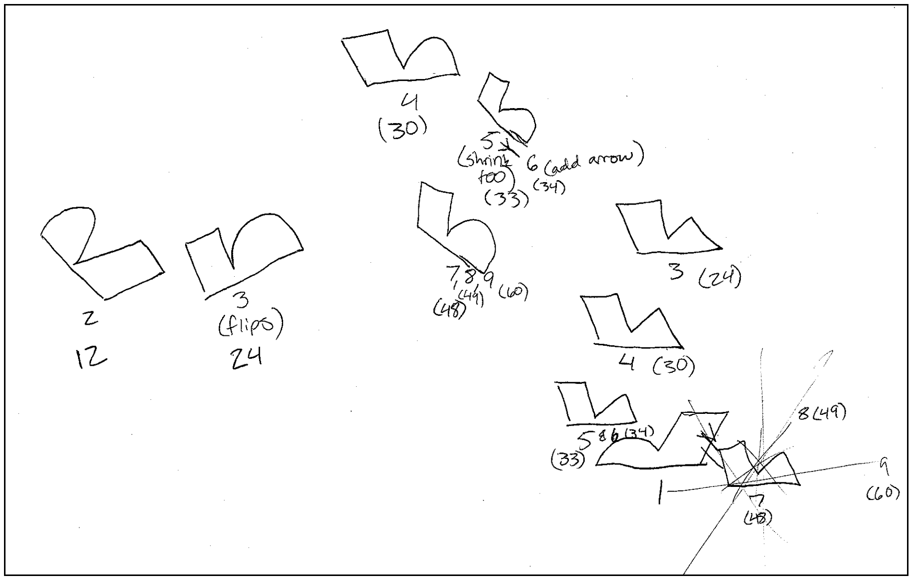
Task F – Participant 3
The TAB Lite



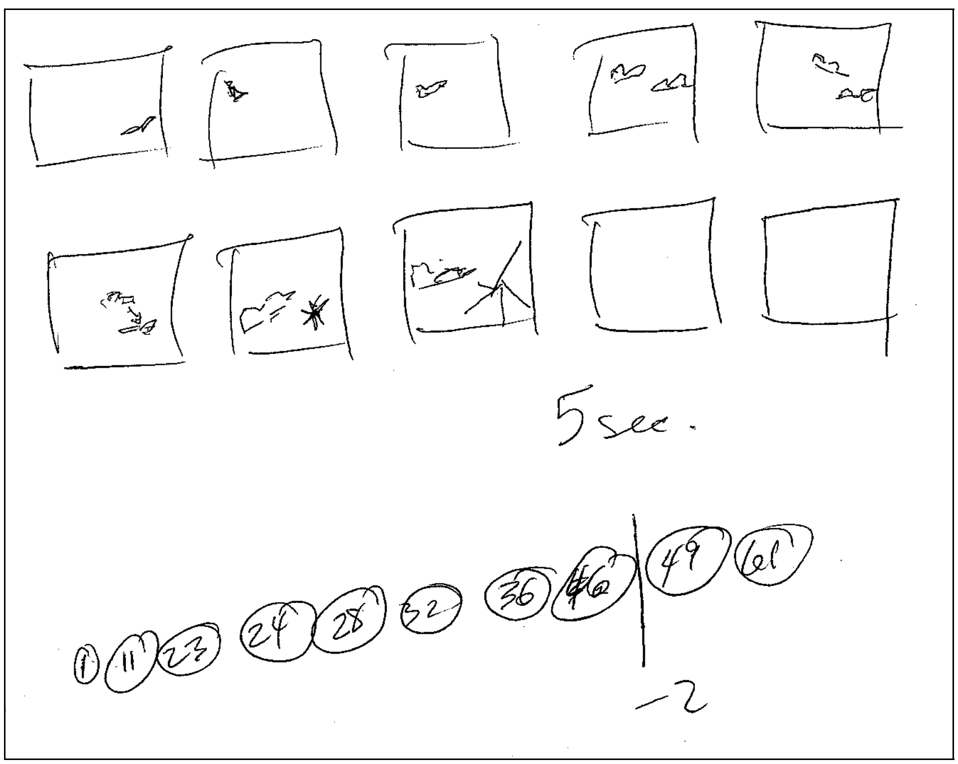
Task F – Participant 6 – The TAB Lite



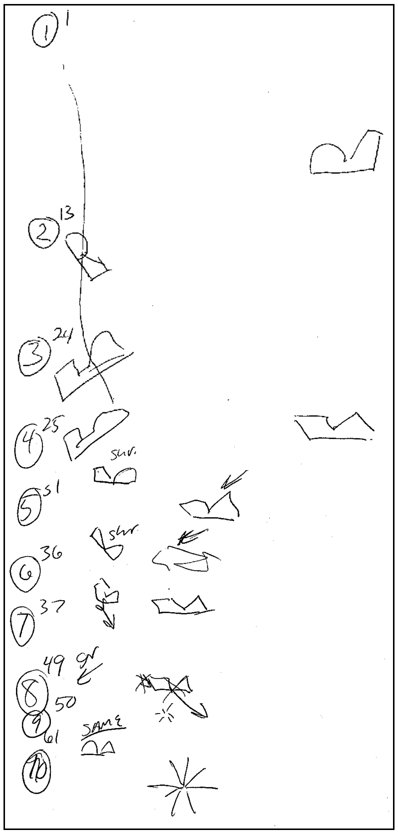
Task F – Participant 7 – The TAB Lite



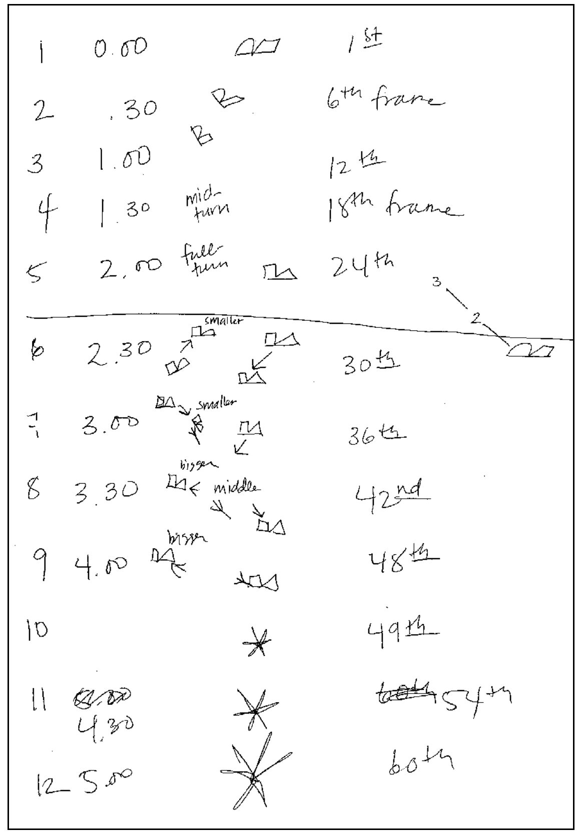
Task G – Participant 1 – The TAB Lite



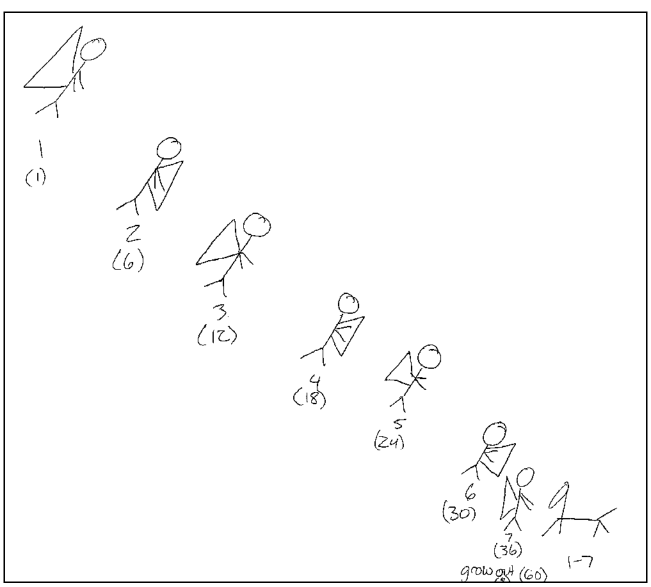
Task G – Participant 2 – The TAB Lite



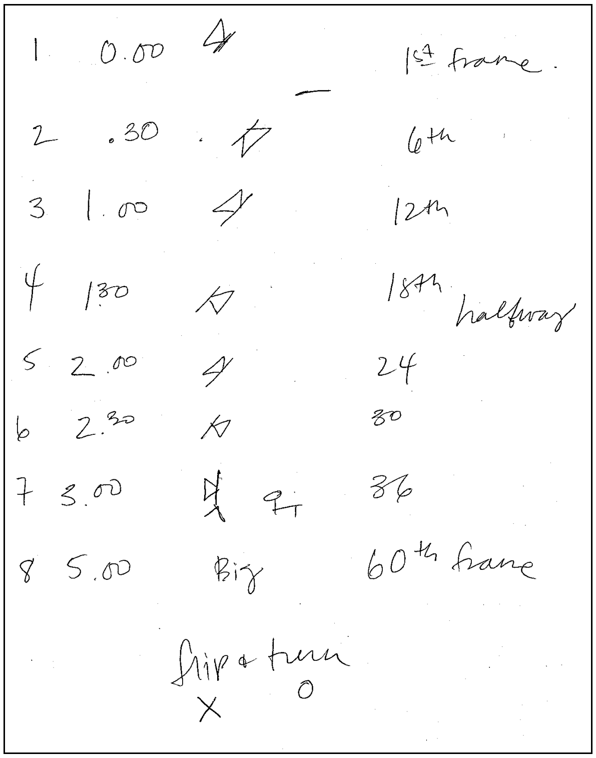
Task G – Participant 6
The TAB Lite



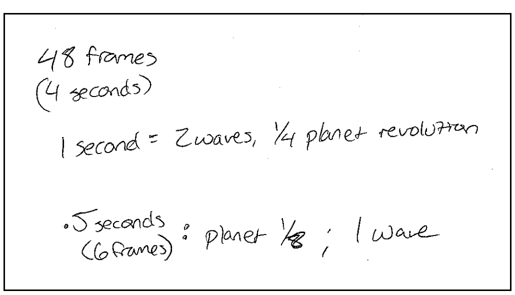
Task G – Participant 7 – The TAB Lite



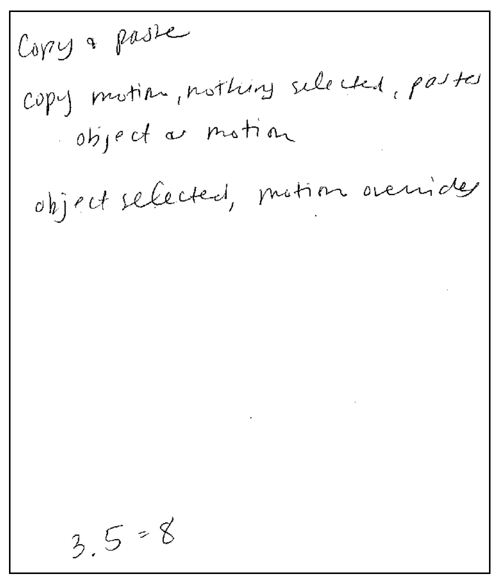
Task H – Participant 1 – The TAB Lite



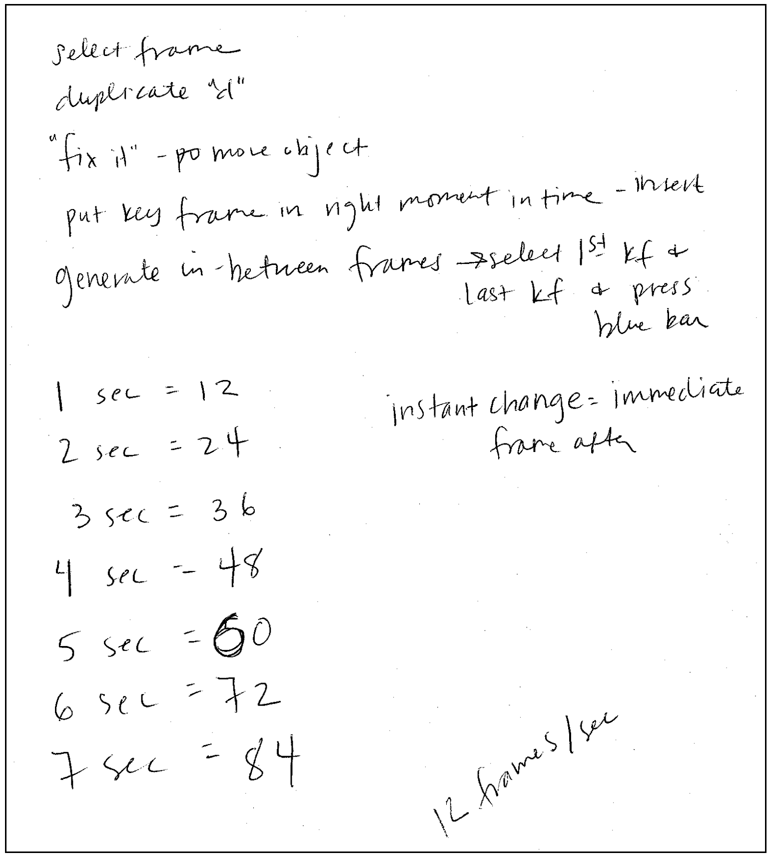
Task H – Participant 7 – The TAB Lite



Task I – Participant 1 – The TAB Lite



Task I – Participant 7 – K-Sketch

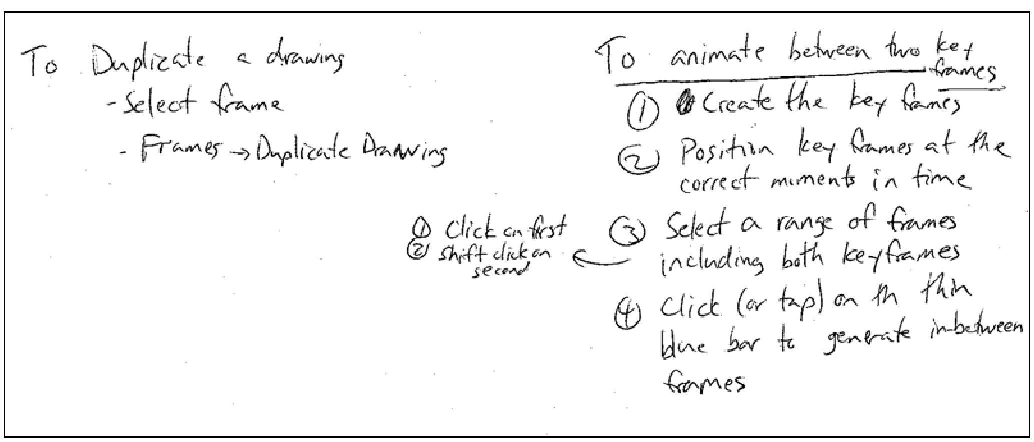

select frame
duplicate "d"
"fix it" - po move object
put key frame in right moment in time - insert
generate in-between frames → select 1st kf & last kf & press blue bar

1 sec = 12
2 sec = 24
3 sec = 36
4 sec = 48
5 sec = 60
6 sec = 72
7 sec = 84

instant change = immediate frame after

12 frames/sec

Notes created by Participant 7 to help when using The TAB Lite


To Duplicate a drawing
- Select frame
- Frames → Duplicate Drawing

To animate between two key frames
① Create the key frames
② Position key frames at the correct moments in time
③ Select a range of frames including both keyframes
④ Click (or tap) on the thin blue bar to generate in-between frames

① Click on first
② shift click on second

Notes created by me to help Participant 4 when using The TAB Lite

H.4 Incidents

H.4.1 K-Sketch Help Incidents

Incident Count	Participant	Task	Incident
6	4	G	general confusion coordinating motions w/fix dialog
4	6	E	general confusion coordinating motions w/fix dialog
10			*total*
3	4	C	forgot how to copy/paste objects
3	4	D	forgot how to flip an object (i.e., get mirror image)
1	4	G	forgot which region of the handle has which behavior
1	4	E	general confusion coordinating motions (overwhelmed?)
1	6	G	need to rotate handle to get flip along correct axis.
1	7	E	strategies for recovering from confusion in coordinating motions
10			*total*
20			*K-Sketch grand total*

H.4.2 The TAB Lite Help Incidents

Incident Count	Participant	Task	Incident
1	2	C	general confusion generating in-betweens
4	4	C	general confusion generating in-betweens
1	7	C	general confusion generating in-betweens
6			*total*
2	4	B	unknown cause of bad in-between
1	2	A	why does the wheel look like it's spinning backwards?
1	4	A	must tap canvas after selecting frame to paste ink
1	3	C	forget that shift-click adds to selection
1	4	C	forgot how to advance frame-by-frame
1	4	D	forgot how to "swing"
7			*total*
13			*The TAB Lite grand total*

H.4.3 K-Sketch Bug Incidents

Incident Count	Participant	Task	Incident
2	4	F	handle jumps after moving center point
1	6	D	handle jumps after moving center point
3			*total*
1	3	F	looks like it's animating, but it's not
1	3	H	looks like it's animating, but it's not
1	4	E	looks like it's animating, but it's not
3			*total*
1	3	H	bad center point or reference frame chosen for interaction
1	4	G	bad center point or reference frame chosen for interaction
2			*total*
1	3	C	crash - data recovered
1	7	G	crash - data recovered
2			*total*
4	3	H	corrupted references frames causing confusing motions
2	4	G	rotation spins opposite direction
1	5	E	confusing feedback after moving timeline tic
1	4	E	speed slider preview plays when pen removed
1	5	G	appeared to draw ink, but disappeared on screen refresh.
9			*total*
19			*K-Sketch grand total*

H.4.4 The TAB Lite Bug Incidents

Incident Count	Participant	Task	Incident
3	4	C	fill color expands into a region it shouldn't
2	4	C	filled region doesn't get in-betweened
1	3	E	crash: minimal data lost
6			*The TAB Lite grand total*

H.4.5 K-Sketch Task Incidents

Incident Count	Participant	Task	Incident
1	2	G	trouble understanding what parts are missing
1	3	F	trouble understanding what parts are missing
1	3	G	trouble understanding what parts are missing
1	4	G	trouble understanding what parts are missing
2	5	E	trouble understanding what parts are missing
1	6	C	trouble understanding what parts are missing
4	6	E	trouble understanding what parts are missing
1	7	C	trouble understanding what parts are missing
1	7	E	trouble understanding what parts are missing
1	7	F	trouble understanding what parts are missing
2	7	I	trouble understanding what parts are missing
16			*total*
1	2	E	trying hard to fix something that doesn't need fixing
1	4	G	trying hard to fix something that doesn't need fixing
2			*total*
1	2	E	confused and wants to stop, but i ask to continue
19			*K-Sketch grand total*

H.4.6 The TAB Lite Task Incidents

Incident Count	Participant	Task	Incident
1	2	G	trouble understanding what parts are missing
1	3	G	trouble understanding what parts are missing
2	4	C	trouble understanding what parts are missing
1	5	B	trouble understanding what parts are missing
1	5	I	trouble understanding what parts are missing
1	7	C	trouble understanding what parts are missing
2	7	E	trouble understanding what parts are missing
2	7	F	trouble understanding what parts are missing
11			*The TAB Lite grand total*

H.5 Post-study Data

	Tasks finished		Enjoyed? (7 = strongly agree)		Use again? (7=very likely)	
Participant	K-Sketch	TAB Lite	K-Sketch	TAB Lite	K-Sketch	TAB Lite
1	9	9	7	6	7	5
2	9	7	7	6	7	4
3	9	6	6	4	6	3
4	8	5	5	2	5.5	1
5	9	9	5	6	5	6
6	7	7	6	6	3	5
7	9	8	7	5	7	3
Averages	*8.6*	*7.3*	*6.1*	*5.0*	*5.8*	*3.9*

Total number of tasks finished and responses to post-study questionnaire questions

H.6 Other Observations

I recorded an *incident* whenever under very specific circumstances, usually when I had to intervene. However, I observed participants experiencing difficulty at many times when no incident was record. This section lists the problems I observed both with K-Sketch and The TAB Lite. Each table lists the number of participants who had the difficulty as well as a concise list of all participants and tasks where the difficulty was observed.

H.6.1 K-Sketch Difficulty

# participants	Difficulty	Participant:Tasks
2	Needed plans/sketches	3:BCE, 7:I
2	Trouble remembering h regions (more?)	3:F, 4:A
2	Orient to Path when rotate desired	1:C, 7:E
1	Orient to Path when Stretch desired	7:H
5	Expect “proportional” shrink of timeline tics	1:E, 2:DE, 5:DE, 6:G, 7:E
2	Can’t access “menu” button (off screen)	1:H, 6:D
6	Putting events at wrong time (more?)	1:I, 3:D, 4:EG, 5:E, 6:G, 7:BEG
2	Confused by Timeline feedback	5:DE, 6:EG
1	Confused by Relative Motions	7:H
1	Run out of space for new objs. or moving objs.	2:A
2	Touch too lightly when animating (disconnect)	2:C(many), 7:(several)
4	Accidentally overwrite motion	2:DE, 3:C, 4:F, 6:BG
2	Accidentally add motion	3:F, 7:E
2	Trouble connecting timeline tics to events	2:E, 7:H
1	Expect to add graphic to existing motion	2:F
7	Expects relative motion to be in world ref. frame	1:PracF, 2:PracF, 3:PracF, 4:H, 5:PracE, 6:PracF, 7:PracF
3	Select wrong item in fix dialog (hard to see)	2:H, 6:G, 7:F
1	Using fix for non-fixable problem (more?)	7:E
3	Forget to Fix	4:G, 6:E, 7:E
1	Paste motion on selection, see nothing (more?)	2:I
1	Paste motion on object with wrong pivot point	6:D
1	Cut vs. Erase confusion (more?)	3:B
1	Select before erasing (more?)	6:B
2	Speed preview animation won’t stop (more?)	3:G, 6:G
1	Hard to truncate a motion (more?)	5:E
1	Grab moving time bar, expect to stop (more?)	4:B
1	Move handle at one time, but needed it earlier	7:D
2	Trouble coordinating motions	4:E, 7:I
1	Trouble using Alternate button to animate	4:E
2	Trouble choosing pivot points	4:F, 7:EG
3	Trouble Selecting	5:E, 6:C, 7:C

H.6.2 The TAB Lite Difficulty

# participants	Difficulty	Participant:Tasks
7	Needed plans/sketches	1:FGHI, 2:EFG, 3:CDEF, 4:E, 5:BE, 6:EFG, 7:EFGH
1	Run out of space for new/moving objects	6:F
5	Getting lost in timeline/sequence of events	2:E, 3:C, 4:BCE, 5:I, 7:E
5	Overwriting frames without realizing	2:C, 3:B, 4:CE, 6:E , 7:CF
6	Bad in-between	1:BGH, 2:BG, 3:A, 4:BC, 5:B, 7:BH
4	Problems using in between	2:B, 3:B, 4:E, 6:G
3	Problems in betweening one object independently of others	4:BC, 5:B, 7:B
2	Problems getting direction of motion correct	4:C, 5:C
2	Problems with Planning/Math	3:BC, 4:C
6	Problems selecting (objects or frames)	1:E, 2:AB, 3:BCF, 4:ACD, 5:C, 6:C
3	Problems using "onion skin"	2:B, 3:F, 5:D
3	Problems cutting/copying/pasting frames	2:BC, 4:A, 5:C
5	Problems using manipulator	2:C, 4:CD, 5:D, 6:G, 7:F
1	Problems using "swing"	4:D

H.7 Organized Subjective Feedback

This section organizes the subjective feedback that appears in the raw written responses presented at the end of this appendix. Each table lists the number of participants who made each comment, as well as a concise list of all participants and tasks where the difficulty was observed.

H.7.1 K-Sketch Comments Made During Study

# participants	Comment	Participant:Tasks
1	Like it in general	1:G
1	Like WYSIWYG / Immediate feedback	6:AB
3	Like animating by demonstration	2:B, 3:AB, 5:B
1	Like animating one part separately from others	2:E
1	Like making rough motion before timing	1:A
6	Like timeline tic marks (seeing, moving, snapping)	1:AEI, 2:BCD, 3:CE, 5:AG, 6:G, 7:I
3	Like simple timeline	1:AC, 2:A, 3:A
1	Like time slider bar	7:E
1	Like curved motion paths	1:B
1	Like moving without key frames	1:F
1	Like overwriting motion	2:AD
4	Like cut, copy, paste, undo, or redo	2:C, 4:D, 6:CG, 7:BCG
4	Like copying motion	1:DI, 2:I, 3:I, 5:I
1	Like manipulating motions	3:I
2	Like selection method	3:B, 7:A
4	Like object handle	2:FG, 3:H, 4:D, 6:F
4	Like orient to path	1:H, 2:G, 3:B, 5:FH
3	Like move handle feature	1:DF, 3:DE, 5:DH
2	Like rotate handle feature	3:F, 5:H
4	Like "Fix Last Operation"	2:H, 3:G, 6:EFG, 7:F
1	Like relative motions	3:F
1	Like "ghost" after erasing	1:G
1	Like global speed control	1:E
1	Like record drawings	5:G
2	Drawing hard (some want shapes)	1:C, 1:D
1	Moving hard (want precision positioning)	1:G
1	Flipping along desired axis hard	6:G
2	Hard to select (want add/rem. from selection)	1:A, 5:A
1	Hard to copy motions	3:C
1	Hard to see dependencies between motions	1:B
1	Hard to avoid orient to path when want rotate	1:C
7	Hard to coordinate motions	1:EG, 2:CDEGHI, 3:D, 4:E, 5:CE, 6:ACE, 7:CEI

(continued)

# participants	Comment	Participant:Tasks
1	Hard to coordinate relative motions	3:G
1	Hard to avoid blank time at end of animation	1:G
1	Hard to access context menu btn. (off screen)	1:F
1	Hard to remember how to move handle	7:D
2	Hard to read timeline feedback	5:E, 6:E
1	Dislike roughness of global speed control	1:?
1	Dislike using stretch tool to flip	2:G
1	Dislike eraser (erases whole stroke)	5:B
1	Dislike Context Menu button moving off screen	6:D

H.7.2 The TAB Lite Comments Made During Study

# participants	Comment	Participant:Tasks
5	Like in-between tool	1:AB, 2:ABCG, 5:I, 6:ABEFG, 7:A
3	Like cut/copy/paste/undo/redo	2:CDF, 6:CFG, 7:D
2	Like keyboard shortcuts	6:A, 7:CD
4	Like Onion-skinning	1:D, 2:F, 5:F, 6:E
6	Like Swing	1:D, 2:D, 3:D, 5:D, 6:D, 7:D
1	Like Shift key for rotate	1:C
2	Like frame duplication process	1:H, 6:E
2	Like manipulator	3:B, 5:CG
3	Drawing Easy	1:B, 3:AE, 5:A
2	Drawing Hard (some want shapes)	1:C, 6:CG
2	Interface straightforward	5:A, 6:B
1	Interface inefficient	1:BEF
1	Too little screen real estate	1:GH
1	Hard to determine order of animation steps	1:B
4	Hard to use manipulator	2:AG, 3:A, 5:C, 7:H
1	Hard to select	3:F
1	Hard to switch between pen modes	2:G

(continued)

# participants	Comment	Participant:Tasks
2	Hard to use "onion skin"	2:D, 3:E
1	Hard to make motion fluid	1:F
4	Hard to do "math"	1:FG, 2:CDF, 3:C, 7:C
1	Hard to create a good in-between	7:BE
4	Hard to in between one object without breaking others	2:B, 3:BF, 4:C, 5:B
4	Hard to coordinate/keep track of events	1:EG, 2:EF, 5:E, 7:E
3	Hard to navigate timeline	2:E, 5:E, 7:C
2	Hard to keep track of timing	1:E, 5:I
1	Hard to break up into key frames	3:CE
1	Hard to get direction of motion correct	2:C
1	Hard to tell which way an object is rotating	7:C
1	Dislike how easy it is to erase large number of frames with in-between tool	7:C
1	Dislike bad in-betweens	1:BH
2	Dislike having to insert frames with key	3:A, 5:A
1	Dislike absence of global speed control	7:C

H.7.3 Final K-Sketch Comments

Comment	Participants
intuitive	4, 6, 7
easy to use	2, 5, 6
easy to learn	2, 4, 7
easy to navigate	2, 3, 4
pleasant, friendly	7
efficient	7
tasks were easier	1
like absence of frames	3
imprecise	5
overly simple	5
coordination difficult	6
timeline feedback unintuitive	6

H.7.4 Final TAB Lite Comments

Comment	Participants
complicated, "quirky", or "overwhelming"	1, 2, 5, 7
need to be methodical/linear/plan ahead	2, 4, 5, 7
too much math	2, 7
animations look better	1
managing key frames tedious	3, 7
navigating through time difficult	3
key frames familiar/natural	5, 6
like precision	5

H.8 Written Responses

This section lists written responses to short answer questions. Responses to post-task questionnaires are organized first by task, then by tool, and then by question. Responses to other questions are organized by question. All responses are numbered by participant number. If a participant does not appear in a list, it means that the participant did not respond to the question.

H.8.1 Demographic Survey

5) Briefly describe the subject matter and/or purpose of the animations you would like to create.

1. Not sure.
2. Political cartoons and miscellaneous silliness.
3. I'd like to produce animations derived from scenes of novels that I've recently read.
4. I love 2-D animation—I grew up on Warner Bros., MGM classics. Lately "Invader Zimm" is one of my favorites. If I could emulate those I could promote my "pitch" ideas more effectively.
5. I'm interested in creating animation for use on the web (like as an introductory Flash video for a website).
6. I would like to create presentations to assist with corporate church prayer and worship.

7. For use in user-centered design methods.

H.8.2 Post-task Questionnaire: Task A

K-Sketch

[K-Sketch, A] 4) Is there anything that made this task particularly ***easy***?

1. Making the animation *then* manipulating the timeline.
2. Elementary shapes.
3. Not having to worry about individual frames for changing & manipulating the animation.
6. Only two things were happening.
7. Yes. I only animated two things.

[K-Sketch, A] 5) Is there anything that made this task particularly ***hard***?

1. Selecting the diamonds—couldn't shift-multiple select. (*Note: loop select appeared to grab center piece.*)
2. Relative placement of objects.
3. No.
6. Getting the timing right.
7. No.

[K-Sketch, A] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. The second tick marks on the timeline. Didn't get lost & buried in too many frames—the timeline is all right there w/no scrolling!
2. Pretty easy timeline tools. Easy to erase/do over.
3. I liked just having the entire animation on one screen & not having to switch between different frames.
5. The ease of adjusting timelines of individual items.
6. WYSIWIG.
7. Selecting all diamonds @ once was useful.

[K-Sketch, A] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Nope.
2. Nope.
3. No.
5. Having to draw a lasso around every item I want to select rather than being able to click on them individually.
6. No.
7. No.

The TAB Lite

[The TAB Lite, A] 4) Is there anything that made this task particularly ***easy***?

1. Everything happened at the same time, so I only had to make the first & last slide, then let the program fill in the "in between."
2. Just having to do beginning and end frames instead of movement by hand.
3. Positioning the individual shapes was easy and intuitive, since all I had to do was click on them & move them.
5. The animation was short and the movements of many elements (the diamonds) were repetitive.
6. The animate bar.
7. N

[The TAB Lite, A] 5) Is there anything that made this task particularly ***hard***?

1. Not really.
2. Movement of individual diamonds while keeping their alignment with each other (of course now I know I don't have to do them indiv.—can use "shift," but…). Estimating space so that group of diamonds didn't run into face object.
3. It was a little difficult to enlarge the mouth, because I thought that either the widen or enlarge features would've worked.

5. Not that I can think of.
6. No.
7. N

[The TAB Lite, A] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. The in between tool!
2. "In between" or fill-in-motion function. Helps with straight movement—Round or arched (or arced) movement, not so much.
3. The drawings were simple to create.
5. The tablet made things easier and the tool itself has a very straightforward interface.
6. Intuitive keyboard controls.
7. Fill in the frames feature.

[The TAB Lite, A] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Nope.
2. Difficulty with selecting more than one object to move at once…tendency to accidentally resize objects while trying to just move it.
3. Having to duplicate frames by hitting the "insert" key numerous times.
5. Having to add in so many between-frames one at a time.
6. No.
7. N

H.8.3 Post-task Questionnaire: Task B

K-Sketch

[K-Sketch, B] 4) Is there anything that made this task particularly ***easy***?

1. Not really.
2. No rotation / fancy movement.

3. It was easy to select objects to manipulate; I didn't have to worry much about stray pieces.
6. Motions were either fluid or static and were not hard to coordinate.
7. No.

[K-Sketch, B] 5) Is there anything that made this task particularly ***hard***?

1. Not really.
2. No.
3. I'd forgotten the difference between the erase & cut function & that caused some difficulty.
6. No.
7. No.

[K-Sketch, B] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Not having the animation assume straight lines between key frames—made for more smooth lines (except for hand error).
2. Easy click and drag action. Easy time manipulation / adjustment of placement in time.
3. Movement of the sun was very quick & easy to control; also the "steer" function made movement while rotating easy to do.
5. The ability to record in real-time.
6. Task feedback was immediate.
7. Undo helped a lot at the beginning. I was able to recover from my mistake.

[K-Sketch, B] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. (*It bothered this participant that there are hidden dependencies between frames, which prevented her from thinking about key frames independently. She did not know how to express this in words.*)

2. Nope.
3. No.
5. The imprecision of the eraser tool.
6. No.
7. No.

The TAB Lite

[The TAB Lite, B] 4) Is there anything that made this task particularly ***easy***?

1. Not complicated to draw the figures.
2. Movement of the sun was easy with "in between" or fill-in function using key points/frames along path.
3. The sun's movement was pretty easy to animate.
6. Animate function. (*Note, "animate function" means "in-between" function.*)
7. N

[The TAB Lite, B] 5) Is there anything that made this task particularly ***hard***?

1. Not knowing the most productive order in which to do the creation steps—I think there was a more productive/efficient/accurate order than the one I did!
2. Moving the sun while keeping the watching person stationary was tricky → Note being able to "in between" one object without "in betweening" the other(s) and getting crazy Picasso results on my little watcher guy's features.
3. Having to integrate the gradual movement of the sun with the rapid changes in the man's head & facial features.
4. Remembering to keep figure's shape consistent through key frame changes.
5. The fact that each item was changing across different timelines confused me.
6. No.
7. Figuring out how to change the animation. Had to start over, because had a hard time with that.

[The TAB Lite, B] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Still, the in between feature!
2. In-between feature (see this participant's response to (4) above).
3. It was pretty easy to adjust the positioning of the head and nose by tilting them 45°.
5. It was easy to identify where I made mistakes. (*Note, this participant is referring to the fact that many frames are visible at once.*)
6. Easy to erase and redo.
7. N

[The TAB Lite, B] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. When it makes your expected animation go all wonky by assuming it needs to animate pieces it shouldn't in ways it shouldn't!
2. In-between feature (see this participant's response to (5) above).
3. It seemed as if I was not able to rotate the man's head and nose in a way that was consistent. (*Note: this participant forgot how to shift-click to select multiple items at once.*)
6. No.
7. N

H.8.4 Post-task Questionnaire: Task C

K-Sketch

[K-Sketch, C] 4) Is there anything that made this task particularly ***easy***?

1. Nope.
2. Rotation path was clean (not affected by my shaky drawing of circles) using tool.
3. Producing the rotation of the individual gears.
6. Copy paste saved some time.
7. Copy feature. (*Note, this participant is referring to* copy object.)

[K-Sketch, C] 5) Is there anything that made this task particularly ***hard***?

1. The graphic was much harder to draw than any of the others.
2. Synchronization of rotation was difficult.
3. Trying to copy & flip the movement of the gears.
4. Me not remembering simple copy/paste function → user problem → not software.
5. Synching up the teeth of the gears during rotations.
6. Coordinating opposite motions.
7. Getting it to look accurate during the animation—each spoke falling into each other like real gears turning.

[K-Sketch, C] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. The timeline!
2. Copy/paste to get identical gear shapes. Undo, undo, undo! Easy timeline adjustment in terms of start finish.
3. Time & movement manipulation.
6. Undo.
7. No.

[K-Sketch, C] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Grabbing the wrong section of the rotator tool. (*Note: she grabbed* orient to path *accidentally.*)
2. Not exactly.
3. No, just a bit of a learning curve to copy & flip motions.
6. No.
7. No.

The TAB Lite

[The TAB Lite, C] 4) Is there anything that made this task particularly ***easy***?

1. Holding the shift key as I rotated the shape, because that let me know *exactly* when I hit a quarter-turn, which made the animation smoother overall & also saved time in the creation process.
2. "In between" function—once rotation wasn't an issue…(but it was!)
3. Drawing the gears.
6. Copy and paste.
7. N

[The TAB Lite, C] 5) Is there anything that made this task particularly ***hard***?

1. Drawing the shape—I didn't like the end result (& was consequently uncomfortable with it) because my drawing was disproportionate & didn't animate well!
2. Rotation—synchronizing rotation—my own misperception of which direction things were rotating in. The *math* of calculating how many frames per rotation were needed for end time.
3. Positioning the gears, calculating the number of frames & how to break up the rotation of the gears to fit that calculation.
6. Freehand drawing.
7. Y—I missed being able to drag the animation through time at a speed of my choice, where I could comprehend what was happening (like I could with K-Sketch, for example).

[The TAB Lite, C] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Nothing new!
2. In between, cut-n-paste, and undo!
2. Nothing.
5. The rotation tool is easy to use but…

6. Copy and paste.
7. N

[The TAB Lite, C] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Not having pre-existing shapes that would've fit together better (maybe there are these).
2. Math yuck. ☹ (Also see this participant's response to (5) above.)
3. Again, figuring out the frames & different segments.
4. Cannot copy motion of individual objects for replication purposes → have to individually animate everything. (*Note, this is a reference to the fact that adding a second gear that moves at the same time as the first requires changing every key frame.)*
5. …It's difficult to see how far you've rotated unless you pay attention to where the rotational handle began at. (*Note, this participant would start pulling the control at the top right, but would forget her starting point during a drag.*)
6. Freehand drawing.
7. Y—Poor error prevention!! And recovery is very difficult.

H.8.5 Post-task Questionnaire: Task D

K-Sketch

[K-Sketch, D] 4) Is there anything that made this task particularly ***easy***?

1. The legs were all moving w/one kind of motion, which made copying & pasting easy!
2. Understanding the "move handle" function.
3. Using the "move handle" feature made it easy to keep movement along a specific axis.
4. Short cutting w/stretch tool to replace legs in foreground.
6. No.
7. Very straightforward—2 animations, 2 copy/pastes. That made it easy for me.

[K-Sketch, D] 5) Is there anything that made this task particularly ***hard***?

1. Nope.
2. Hard to synchronize motion of legs.
3. Making sure the movements were coordinated by keeping track of the time slider.
6. The example was coordinated in a way I hadn't learned yet.
7. At first, remembering to "move handle," and when to actually move it. But I feel I've got it now.

[K-Sketch, D] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. The handle moving features—very intuitive.
2. Pretty easy to do over/adjust timeline.
3. I liked the "move handle" feature.
5. Being able to move the "handle" of an object when trying to rotate it.
6. No.
7. N

[K-Sketch, D] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Nope.
2. My own drawing! ☺
3. No.
6. The move menu moved off screen.
7. N

The TAB Lite

[The TAB Lite, D] 4) Is there anything that made this task particularly ***easy***?

1. The bug itself didn't move, so I could just concentrate on its legs.
2. Swing and copy/paste functions.

3. The “swing” feature made it easy to produce reverse movement.
5. The movement of the legs was repetitive and simple.
6. Swing.
7. “Swing” feature. Copy/paste shortcuts.

[The TAB Lite, D] 5) Is there anything that made this task particularly ***hard***?

1. Not really.
2. Working w/ “onion skin” function not easily/visually distinctive (e.g., I almost lost track of which “skin” I was looking at among closely placed objects).
3. Nothing.
6. Coordinating multiple elements.
7. N

[The TAB Lite, D] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Having the swing option & the onion skin. The former saved time & the latter helped with accuracy & was fun (I liked being able to see the grayed-out lines even when playing the animation—that was kind of fun).
2. Easy to repeat motions.
3. Using the “swing” feature.
5. The swing function works very intuitively and made this task very easy to complete.
6. Swing.
7. “Swing” feature.

[The TAB Lite, D] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Nope.
2. Nah.
3. No.

6. Rotating and moving an object in one motion. (*Note, this is a task issue, not a tool issue. This participant did not like having to do both a rotate and a move.*)
7. N

H.8.6 Post-task Questionnaire: Task E

K-Sketch

[K-Sketch, E] 4) Is there anything that made this task particularly ***easy***?

1. Ø
2. Being able to isolate parts.
3. Changing the axis & the timeline.
6. No.
7. N

[K-Sketch, E] 5) Is there anything that made this task particularly ***hard***?

1. Getting everything (all the pieces) to move on its own but in relation to everything else. (There was a lot going on—at least to me!)
2. Synchronizing flying objects—throwing and catching.
3. No.
4. Coordination of balls/arms and not knowing shortcuts. (*Note, this participant uses "shortcuts" to refer to the speed that comes from experience with the system.*)
5. The unsynchronized timelines of the different elements.
6. Coordinating opposing motions simultaneously.
7. Coordinating arms & balls together.

[K-Sketch, E] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Slowing down/speeding up time to get it to the appropriate speed.
2. Nothing stood out…
3. Manipulating the speed & the axis of movement.
6. The fix function.

7. Manually going back & forth in time with marker, at my preferred speed. (*Note, this participant is referring to the ability to preview an animation by dragging the time slider bar.*)

[K-Sketch, E] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Not being able to lengthen/shorten the entire animation while keeping everything proportionate. (*Note: this participant could have used the speed control instead of speeding up or slowing down the whole animation, but she did not think it was precise enough.*)
2. Nope.
3. No.
5. The many tick marks on the timeline that appeared (because I had so many different events going on) were confusing and distracting.
6. Finding the right point in the sequence to fix.
7. N

The TAB Lite

[The TAB Lite, E] 4) Is there anything that made this task particularly ***easy***?

1. Nope.
2. Not really…onion skin functionality, maybe.
3. Drawing the pictures.
6. Onion skin.
7. N

[The TAB Lite, E] 5) Is there anything that made this task particularly ***hard***?

1. Keeping track of what needed to move & when/where it needed to move, with respect to the other moving pieces. Timing the animation (I did it by feel).
2. Timing/synchronizing movement of balls in the air and movement of arms.

3. Having to plan the different movements and making sure that the onion skin feature was set on the correct range.
5. The timing of the two balls—synching them and coordinating their movements.
6. Many intermediate states.
7. Synching the balls in my animation with the example.

[The TAB Lite, E] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Not particularly—I didn't use many features. The onion skin was very handy, though. And I was pleased w/how the animation turned out!
2. Neh.
3. No.
6. Duplicate and between functions.
7. N

[The TAB Lite, E] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Could've used a timeline, to be able to time it better/more quickly.
2. Can't see timeline in one screen / difficult navigation of timeline.
3. Using the onion skin…having to set it.
5. Not having the keyframes visually distinctive is sometimes frustrating when making changes.
6. No.
7. I don't trust the "fill-in" feature. It saves time if it does it right, but if it doesn't it becomes a huge time-burner.

H.8.7 Post-task Questionnaire: Task F

K-Sketch

[K-Sketch, F] 4) Is there anything that made this task particularly ***easy***?

1. Not having to worry about key points, really. The animation was very fluid.

2. Selection and motions all combined in one tool.
3. Being able to rotate the axis to keep the flame consistent in its movement; being able to cause the flame to "flicker" by manipulating the transparent image.
6. Fix function.
7. Just 3 things to do that didn't require much coordination.

[K-Sketch, F] 5) Is there anything that made this task particularly ***hard***?

1. Ø
2. Difficult to do inversion/flickering without also changing size of object in motion. (*Note: this participant wanted a more subtle flicker than she could make.*)
3. No.
6. No.
7. N

[K-Sketch, F] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. How there is no in-betweening. Moving the "handle" of an object.
2. Not really.
3. (*See this participant's response to (4) above.*)
5. The "steering" tool.
6. The stretch function felt intuitive.
7. Fix option with hammer/wrench icon.

[K-Sketch, F] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Ø
2. Nope.
3. Not really.
6. No.

7. N

The TAB Lite

[The TAB Lite, F] 4) Is there anything that made this task particularly ***easy***?

1. Having mapped out the key points graphically ahead of time (with what # slide they corresponded to).
2. Copy and paste.
3. Getting the flame to "flicker" by making it larger & smaller.
5. The movements were very repetitive.
6. Copy & paste.
7. N

[The TAB Lite, F] 5) Is there anything that made this task particularly ***hard***?

1. Trying to get the movement fluid.
2. Having to keep track of key frames. "Delete" key too close to "insert" button!! ☹ Copy and paste doesn't carry over next onion skin, which makes it a slightly flawed strategy for extension of movement.
3. Making sure that each part had been "shift-clicked," making sure the torch stayed w/ the man.
6. Keeping inside the drawing space.
7. Flame flickering.

[The TAB Lite, F] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. It didn't make any goofy assumptions while filling in the in-between this time!
2. Once you get used to onion skin it's very helpful at least for movement of one object. Pretty easy to line up key frames of repetitive motions/positions.
3. No.
5. Onion skinning.
6. Between function.

7. N

[The TAB Lite, F] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. The timing isn't very intuitive for me—I wish there were 10 slides per minute, just for easier math (but I understand now having 12 is graphically better than 10!)
2. Framing / Timing calculations.
3. Having to click individual pieces of the drawing & making sure I had done so.
6. No.
7. N

H.8.8 Post-task Questionnaire: Task G

K-Sketch

[K-Sketch, G] 4) Is there anything that made this task particularly ***easy***?

1. Ø
2. Nah.
3. Making the objects appear suddenly; having the system help me to configure the motions.
6. Undo, fix, timing adjustments.
7. N

[K-Sketch, G] 5) Is there anything that made this task particularly ***hard***?

1. Having multiple actions happen @ the same time both between & within objects.
2. Lots of movement at once—added challenge of growing/shrinking while in motion and other stuff happening in accord.
3. I had a hard time making the first plane rotate & move at the same time.
4. Me not remembering how to short-cut processes versus working in a linear fashion.
5. The fact that there were multiple moving objects that had varying movements.

6. The stretch function was much less intuitive this time. (*Note, this time was different, because the participant needed to rotate the handle before stretching.*)
7. Figuring out the explosion.

[K-Sketch, G] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Just generally—I liked using it.
2. Sizing and movement functions in one tool.
3. I liked the "guess" function, because it allowed me to do what I wanted without going back and forth.
5. The "record drawing" tool; being able to adjust multiple timelines to end/begin at the same time.
6. Making the explosion was fun.
7. Recovering from mistakes = great!

[K-Sketch, G] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. How the end time of the animation kept jumping outward to where no animation was going on.
2. Difficulty of inverting and avoiding re-sizing. A little awkward trying to "orient" to path separate from motion/transversion/rotation.
3. No.
6. Sizing and stretching felt hard to do, sizing was intimidating, but the fix function made it easier than expected.
7. N

The TAB Lite

[The TAB Lite, G] 4) Is there anything that made this task particularly ***easy***?

1. Again, planning it out first & being able to look @ the orininal & my drawn paper plan together.

2. "In between" function.
5. The keyframes for each element matched up with the other elements.
6. Betweening.
7. N

[The TAB Lite, G] 5) Is there anything that made this task particularly ***hard***?

1. Lots of things happening all together.
2. Simultaneous movements and changes in size of objects.
6. Drawing.
7. Lots of steps.

[The TAB Lite, G] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Nothing particular.
2. "In between" function.
5. The tools for manipulating a selected item were very helpful. (*Note, this participant is referring to how she likes having all functions in one manipulator.*)
6. Copy paste. Betweening.
7. N

[The TAB Lite, G] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Nope.
2. Switching between modes/tools to complete movements / zoom in–out somewhat awkward. Sizing tool only goes in one direction / at one angle. Somewhat awkward when trying to resize things in different directions / at different angles.
6. No.
7. N

H.8.9 Post-task Questionnaire: Task H

K-Sketch

[K-Sketch, H] 4) Is there anything that made this task particularly ***easy***?

1. Not too much moved at once.
2. Fix prev. operation function.
3. Angling & moving the character & its wings at first.
7. N

[K-Sketch, H] 5) Is there anything that made this task particularly ***hard***?

1. Ø
2. Inversion movement and rotation all happening on the same image/character.
3. I either made a mistake with the button or there was a bug at a certain point, but it was hard for me to go back & try to fix the mistake. (*Note, the wings in this participant's animation separated from the character's body in a strange way.*)
7. N

[K-Sketch, H] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Moving handles & rotating handles. [The orient to path] area on the controller.
2. Fix prev. operation function.
3. I liked the ease of use of the focus feature. (*Note, this is a reference to the scale operation.*)
5. The ability to move/rotate the "handle"; the "steering" functionality.
7. N

[K-Sketch, H] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. When the menu for handles was inaccessible (behind the timeline).
2. Nah.

3. Trying to figure out if I had made a mistake in selecting the right choice in the "fix last operation" or if I had made a mistake with the original animation.
7. N

The TAB Lite

[The TAB Lite, H] 4) Is there anything that made this task particularly ***easy***?

1. Mapping it out first. Not too much was going on at once! The timing was very evenly spaced.
5. Repetitive movements.
7. N

[The TAB Lite, H] 5) Is there anything that made this task particularly ***hard***?

1. Not really.
7. N

[The TAB Lite, H] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Duplicating slides, then manipulating them was handy (rather than copying/pasting/etc.).
7. N

[The TAB Lite, H] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. That it can't read my mind about what I'm trying to do when I copy & paste—which makes the in-between go a little crazy.
7. Manipulation points that appear after selecting an object—they are not flexible (I can't move them to where I want them).

H.8.10 Post-task Questionnaire: Task I

K-Sketch

[K-Sketch, I] 4) Is there anything that made this task particularly ***easy***?

1. Being able to repeat motions (repetitive) rather than do each object individually.
2. Duplication / cut-n-paste of motion functions.
3. Being able to paste both objects & their motions and then being able to manipulate the sets of objects & motions saved time & effort.
7. N

[K-Sketch, I] 5) Is there anything that made this task particularly ***hard***?

1. Nope.
2. To make timing of revolution and wave patterns coincide/coordinate would be more difficult if I wanted to be more precise.
3. Not really.
7. Having to time things in this one was a challenge.

[K-Sketch, I] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. Copying motions.
2. Duplication / cut-n-paste of motion functions made it easy to make a relatively impressive wave effect.
3. I liked being able to select & move objects & motions.
5. The ability to copy/paste animation paths.
6. Tick marks!

[K-Sketch, I] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Nope.
2. Nope.
3. No.
7. N

The TAB Lite

[The TAB Lite, I] 4) Is there anything that made this task particularly ***easy***?

1. How evenly spaced it was & how drawable the shapes were.

[The TAB Lite, I] 5) Is there anything that made this task particularly ***hard***?

1. Not really—once I got started, it made sense.
5. Manipulating identical elements became confusing at times.

[The TAB Lite, I] 6) Is there anything you particularly ***liked*** about this tool while completing this task?

1. How it filled in the revolving planet kind of curved, rather than a perfectly straight line. (*Note: this didn't actually happen.*)
5. The ability to override in-between frames after altering keyframes.

[The TAB Lite, I] 7) Is there anything you particularly ***disliked*** about this tool while completing this task?

1. Nope.
5. Not having a keyframe marked / not having a timeline (in seconds) next to the frames.

H.8.11 Post-study Questionnaire

4) What factors influenced your answer to the preceding question (How likely are you to use K-Sketch again)?

1. I had a blast with it!
2. Simple to learn and navigate. Relatively easy to use and make decent animated works.
3. Mostly the fact that K-Sketch doesn't have individual frames. A lot of the difference in features seems to come from this.

4. K-Sketch felt more intuitive. It made animation for a "first-timer" more accessible quicker. It provided several more short cuts that TAB Lite didn't offer. Time manipulation (event markers) make K-Sketch easier to use in half the time TAB Lite did.
5. The tool seemed overly simple at times and imprecise for complex animations/movements.
6. Comparison to TAB Lite.
7. Learning curve is not steep, and the program delivers.

6) What factors influenced your answer to the preceding question (How likely are you to use the TAB Lite again)?

1. After using K-Sketch, I don't really think I have the patience for TAB Lite's quirks—every one of the tasks was easier w/K-Sketch. However, I had fun with TAB Lite too… and some of the animations *looked* better in TAB Lite— & it had more colors.
2. More sophisticated, but more complicated. Pretty easy to screw up at any given step. Necessity to be more methodical in creating. Too many buttons for a simple farm girl like me…entirely too much math, especially once you start changing/replacing frames.
3. Having to both worry about frames & the overarching animation seemed tedious. (*Note, this is a reference to the fact that changing one key frame can have far reaching effects on other frames.*)
4. (*See this participant's response to (4) above.*)
5. I liked that the keyframes felt familiar (from using Flash before) and even though it felt more complex thank K-Sketch, I liked how precise it was.
6. Comparison to K-Sketch.
7. Is a pretty overwhelming tool. If I hadn't had someone here to teach me how to use it, I have a feeling it would have taken way too much time to learn it otherwise. I would have abandoned it. PRICE. (*Note, this last word indicates that this participant felt that the TAB Lite was too expensive for her to purchase it.*)

7) Are there any comments you'd like to make about K-Sketch or your experience of using it during this study?

1. Think I made them throughout the study! But I really enjoyed it! Of the two tools, this one was my favorite.
2. (*See this participant's response to (5) above.*)
3. I thought that there was a certain smoothness that related to being able to see the animation all at once & then going to certain points w/in the animation to make changes of additions. This often helped me keep my focus on the entire animation rather than worrying about bits & pieces of it.
4. I think if K-Sketch includes some layering tools (back, mid, foreground elements) a geometric shape tool, and a color palette, it will be a great introductory platform for novice animators.
5. It was great being able to manipulate objects in so many ways—rotating, steering, shrinking, flipping—and animating them was easy, but the whole animating process felt sloppy and imprecise.
6. The drawing controls were intuitive and easy to use. I felt less clumsy manipulating drawings. I felt much less competent trying to coordinate more than one object simultaneously. The timing bars at the bottom did not feel intuitive.
7. Looking back, I think it's a tool that was certainly created with a user-centered perspective. The differences between the complicated animations were virtually unrecognizable. I could not tell you if one tool's animations were "cleaner" or more closely resembled the originals than the other's. Furthermore, my experience using K-Sketch was better—more pleasant and *way* more efficient. This results in an overall better user experience. I had to meticulously map out plans on TAB *and* do math. I'm not math-phobic, but having to do it was time-consuming. With K-Sketch, I didn't have to draw plans or do calculations, which leads me to believe it is a more "intuitive" tool.

8) Are there any comments you'd like to make about the TAB Lite or your experience of using it during this study?

1. I think I made them throughout the study!
2. (*See this participant's response to (6) above.*)
3. Adversely, I think having to switch back & forth between frames & the animation was a two stage process; having to go between those stages repeatedly was tedious.
4. TAB Lite is clunky, in that for the most part events have to be placed in a linear fashion. The "juggling man" took a notable amount of time. I'd hate to undergo the "fighter jet" sequence → one might as well learn actual animation techniques. (*Note, "actual" animation refers to traditional frame-by-frame animation.*)
5. This was definitely a more complicated tool and required more planning to get multiple elements moving on different timelines. However, it also felt like it would be easier to make complex animations or precise changes.
6. At first I thought it would be much harder than using K-Sketch, but I quickly acclimated. At first, I thought the film reel graphic a little silly until I saw how it simplified getting the timing right. It seemed like it was more patterned on an animator's mental model than on a novice mental model, but I soon began thinking and feeling more like an animator anyhow.
7. It was interesting using TAB *after* K-Sketch, because there were things that I really wanted TAB to do (that K-Sketch *did* do), that it didn't. TAB assumes one conclusion, whereas K-Sketch provides me with options. (*Note, this participant is saying that K-Sketch's method for in betweening is more flexible than the TAB Lite's.*) That is a very important feature that contributes to an improved user experience. Additionally, I had more control over "time" in K-Sketch. One would think than an animation tool would provide optimal control over time, yet TAB didn't do that either. It kind of made me feel: "Well, you *should* know this, you *should* be able to do this" —thus making me feel sorta stupid. In contrast (and in hindsight), K-Sketch seemed to

automatically account for human (or maybe it's novice) limitations. TAB appears to have a lot more sophisticated features, but for the purposes of the animations created in this study, K-Sketch was better, friendlier.