idAnimate: A General-Purpose Animation Sketching Tool for Multi-Touch Devices

J. Quevedo-Fernández and J.B.O.S Martens
Department of Industrial Design
Eindhoven University of Technology
Eindhoven, The Netherlands
{j.quevedo.fernandez, j.b.o.s.martens}@tue.nl

***Abstract*— Creating animations is a complex activity that often requires an expert, especially if results need to be obtained under time pressure. As animations are potentially relevant in many different contexts, it is interesting to allow more people to use them for communicating ideas about time-varying phenomena. Multi-touch devices create opportunities to redesign existing applications and user interfaces, and new classes of animation authoring tools that use gestural interaction have therefore started to appear. Most of them focus on specific applications, such as cartoon and puppet animation. In this paper, we present *idAnimate*, a low-fidelity general-purpose animation authoring system for sketching animations on multi-touch devices. With *idAnimate,* the user can intuitively manipulate objects with his fingers, while the system records the trajectories and transformations and uses them to build animations. Next to describing *idAnimate,* we also present a comparison study with an alternative state-of-the-art animation-sketching tool, called *K-Sketch*. The experimental results lead us to conclude that *idAnimate* is easier to understand and learn, and that it is significantly faster for certain types of scenarios, at the cost of a reduced precision and detail in the outcome.**

Keywords-Animation; Multi-Touch; Creativity-Support-Tools; User-Study; Design

I. INTRODUCTION

If a picture is worth a thousand words, then a movie is worth a thousand pictures. Compared to static images, animations and motion pictures more explicitly illustrate the course of actions and reactions that constitute an inherently dynamic situation. This can lead to improvements in communication and understanding [[3], especially when describing complex concepts that involve a chain of events [5]. As animations are of interest in a variety of contexts, from education to business, and can support both information transfer and artistic expression, it is worthwhile to explore how they can become more widespread.

Despite the potential interest, it turns out that non-expert animators hardly ever use animation-authoring tools. A number of animation tools exist today, the most popular one being Adobe Flash [4]. However, as argued more explicitly in the next section, for many contexts and users, such professional animation tools lack two important characteristics; (1) they are not easy to learn and use, and hence only adopted by expert animators who can afford the initial investment, and (2) they are insufficiently sketch-like, requiring an effort and commitment to detail that is quite substantial and often not needed.

In an attempt to bridge this gap between existing tools and user requirements, a number of animation tools and techniques that feel more natural and intuitive in the hands of non-expert animators have been developed. Recent developments in input devices, in particular the increasing popularity of multi-touch interfaces, have opened up new opportunities to develop easier to learn and faster to use animation techniques, but there is as yet no consensus on what is the most effective approach.

In order to explore and test how animation tools could profit from the benefits that multi-touch interfaces have to offer, we have developed a specific animation tool, named *idAnimate,* for them. To better understand the consequences of the design trade-offs adopted in *idAnimate*, we have conducted a small experimental study to compare *idAnimate* with an existing state-of-the-art animation-sketching tool, called *K-Sketch.* This latter application was developed for tablet devices and hence does not incorporate multi-touch.

This paper first describes the related work and the *idAnimate* tool. The description of the tool includes the design goals and the main characteristics of the user interface and animation technique. The paper continues presenting the comparison study, which includes a brief description of *K-Sketch.* The paper subsequently presents and discusses the experimental results, and draws lessons for the future.

II. RELATED WORK

In 1969 Ronald M. Baecker presented Genesys [17], the first computerized animation system that recorded dynamic changes in the position, orientation and shape of visual objects as they were applied by the animator (usually with the help of a stylus). Since then, many systems have followed, introducing a variety of additional animation techniques. Using the flipbook metaphor, GIF animations grew popular during the 90's, partly because they could be easily distributed digitally. Adobe Flash and Director popularized the key-frame technique, in which the animator creates two frames and the system interpolates in between. Another important technique that is commonly used in video games and film production is procedural animation, where computer algorithms define the motion and position of visual objects over time [2]. However, while obviously very useful in the hands of skilled animators, these tools do not explicitly address novice (or infrequent) users.

Some newer systems have focused on developing techniques that feel more natural and intuitive to non-experienced users. Two specific examples are: a) articulating 3D figure animations based on 2D figure sketches [9] and b) sketching the motion of a character [14]. While non-expert animators can use such tools, they offer interaction mechanisms that are tuned to a specific application, i.e., articulating the motion of a character, and are therefore not necessarily suitable for general-purpose animation.

More interesting in view of general-purpose animating is the motion-by-example technique described by Moscovich. This technique, in which the user can drag an object around the screen while the system records the location trajectory as a function of time, has been widely adopted in animation tools that target novice animators, such as *K-Sketch* [16], *Sketch-n-Stretch* [8], or *Sketchify* [19]. *Sketchify* (which was co-developed by the second author) is primarily intended for quickly creating prototypes with real-time input from a diversity of sensors, and therefore its interface is simply too sophisticated and complex for rapidly sketching animations. The tool *idAnimate* that we present in this paper shares similarities with *K-Sketch* and *Sketch-n-Stretch*, as it extends the motion-by-example technique, which tracks an object's position, to also include orientation and size. The extension to multi-touch devices (specifically, the iPad) can potentially speed up an animation process, as the user can transform multiple properties at a time instead of consecutively. An additional advantage of multi-touch devices is that multiple users can potentially cooperate on a single animation, although the full potential of this idea is only likely to emerge on multi-touch surfaces that are larger in size than the iPad that we used in the current case study.

Ceylant and Capin [6] developed a multi-touch interface for animating 3D Meshes that produced a sense of movement in objects by deforming and reshaping visual objects through the fingers, which is similar to the multi-touch technique by Takayama and Irigarashi [11] for manipulating the shapes of 2D characters. While both techniques share some features with our own prototype, mostly due to the common use of a multi-touch interface, they are mainly intended for character animation, and are hence fairly specialized. To the best of our knowledge, the most similar existing tools to *idAnimate* are *Toontastic [1]* and *Photopuppet HD* [15], which use multi-touch gestures for creating animations. The latter tool is mostly intended for cartoon animation in a professional studio, as it offers many sophisticated and dedicated features, while *Toontastic* has a very specific application domain, i.e., storytelling for children.

III. IDANIMATE

A. *Overview*

The primary goal of *idAnimate* is to allow users to rapidly author simple but expressive animations. The concept of *idAnimate* was inspired by a series of studies that were carried out within the design teams of diverse companies. These studies aimed more generally at collecting requirements for tools and methods that can support design communication [7, 10]. The need to author dynamic visualizations in a sketch-like way emerged quite naturally in this context study.

As the aim was to create a tool that could support creativity, discovery and innovation, we drew inspiration from the design guidelines proposed by Shneiderman [20] and Resnicks [21], to assist us in mapping the general objective of sketch-like animations into more specific design goals that can potentially also be verified in subsequent observational studies.

B. *Design Goals*

1) *Low Threshold*

In order to make the process of creating animations easily accessible to non-experienced animators, the threshold for using the tool should be extremely low. This means that novice users, including those using the tool for the first time, should be able to create simple animations with no or minimal instruction.

2) *Speed*

The tool should allow users to rapidly create animations. Ideally, the cost of creating an animation should be similar to that of creating static sketches, so that they can just as easily be discarded and replaced by alternatives.

3) *Support Exploration*

The tool should allow users to try out different alternatives and examples, and should make it easy to roll back to past situations.

4) *Flexibility and wide walls*

The tool should not limit its users to a predefined set of patterns or prescribed scenarios. The tool should ideally allow the majority of users to describe most ideas that they can think of, within the reasonable boundaries of what animations can describe.

5) *Simplicity*

Besides having a low threshold, the tool should feel simple and intuitive to use after a prolonged period of time. Instead of providing an extensive set of highly configurable features, the tool should provide only those that are strictly necessary to achieve the desired flexible outcome.

C. *Tool design*

The aforementioned design goals led us to select tablet devices as the platform for the prototype, mainly due to their form factor, portability, and the affordances that such multi-touch devices provide.

Multi-touch interaction paradigms such as resizing and rotating make it possible to extend the existing motion-by-example method [18] to the more general transformation-by-example method that is at the heart of *idAnimate*. This latter technique essentially allows users to freely manipulate objects using a multi-touch surface while recording the changes in position, size and orientation over time.

Commonly accepted multi-touch gestures are generally perceived as highly intuitive thanks to the direct manipulation of objects through the tip of the fingers, which effectively maps the finger movements to the geometric parameters that are being controlled by the system.

 ISBN: 978-1-61208-275-2

Since our technique complies with common practice, it should be perceived as easy to learn and easy to use. As stated before, next to being engaging and inviting, multi-touch gestures are potentially faster, as three different transformations can be applied simultaneously.

The projects produced with the tool are called Animated Sketches. An Animated Sketch consists of a group of visual elements (objects) and their associated transformations (position, orientation, size) over time. It is possible to import visual elements from different sources into the application, including attachments in emails, images resulting from internet searches (Bing, Google or Flickr images), photographs taken with the built-in camera (if the device has one), or images already available in the users library. The tool also includes a simple sketching application that can be used to create new visual objects, or to modify existing ones.

Figure 1 shows a screenshot of the animation editor. Mark 1 shows the list of objects that are contained in the current animation. By modifying the order of objects in the list, they move in front or behind each other. Mark 2 shows the controls for adding a new object, editing the images associated with an object, duplicating, inspecting, deleting or clearing the animations of the selected object. Mark 3 shows the animation controls including the timeline and the play, pause, record and rewind buttons. Mark 4 shows the canvas where the objects are displayed and where the user interacts to define the animations. Mark 5 shows the inspector, which allows users: 1) to change between the different images that an object can display, 2) to manage the geometrical properties of an object that can be transformed, and 3) to specify when an object is visible or invisible.

D. *The animation technique*

Animated Sketches are composed of one or more visual elements called Animated Objects. An Animated Object is a region of varying position, size and orientation that can at any instance display one of a set of possible images, where most often the image reflects the state of the object concerned. This way an object can be moved, rotated and scaled while its visual appearance (image) changes.

1) Transforming through gestures

Multi-touch gestures are the key to intuitively defining the transformations on Animated Objects. In order to animate an object, the user has to select it from the list of available objects in a scene. Using two fingers, a user can translate, rotate and scale simultaneously the selected object in a single action. While recording, all of the transformations performed on the device are registered and time-stamped relatively to the animation start time. When the user initiates the record action, a three second countdown is displayed to give him time to prepare the position of the fingers.

In order to provide control over small objects, despite the limited size of the tablet display (9.7 inches in diagonal), the user does not necessarily need to touch the actual object, but instead he can interact anywhere on the device. This way the user can also avoid blocking his view on the selected object. However, the consequence of this design choice is that it is currently not possible to animate two or more objects simultaneously, which is something that will probably need to be reconsidered when migrating to large-sized multi-touch displays that are operated by several users at a time.

2) Synchronization and Concatenation

Complex animations are accomplished by adding new object transformations to existing animations, effectively concatenating transformations. In order to coordinate the animations of different objects, we use the record-while-playback technique. The user can replay a previously created animation, and meanwhile new transformations can be applied on other objects. These additional transformations are recorded, time-stamped and synchronized with the existing animation.

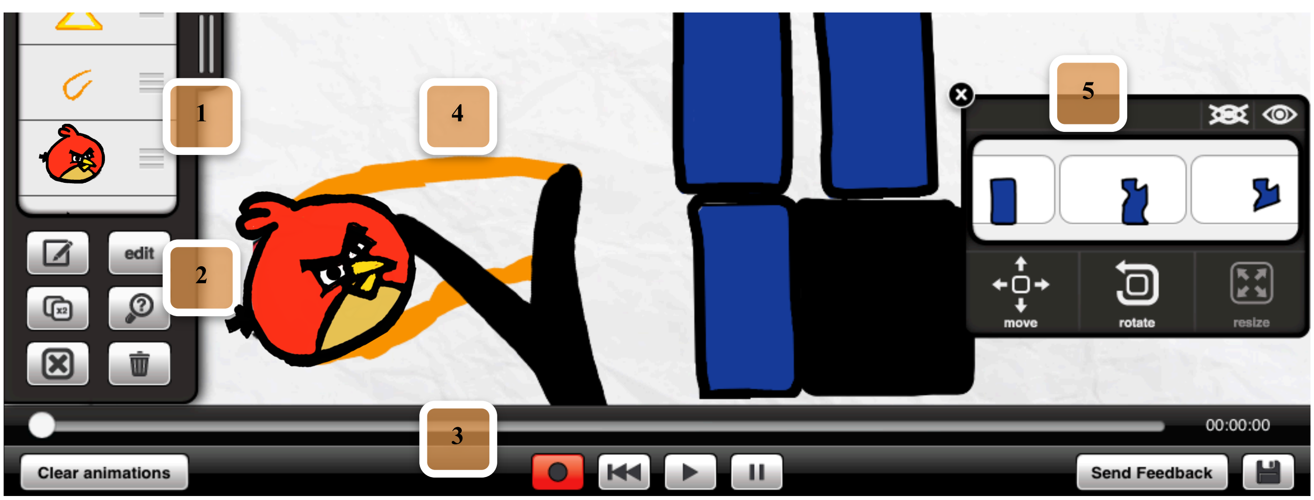

Figure 1. *idAnimate* being used to animate a game concept (Angry Birds). Mark 1 shows the list of objects, mark 2 shows the possible actions on an object, Mark 3 the record and playback controls, mark 4 the canvas and Mark 5 the object inspector.

 ISBN: 978-1-61208-275-2

If a user animates an object that was already animated previously, all existing object transformations, from the moment that he starts to interact, are overwritten. Therefore the user can easily and intuitively refine or redo part of the animation of an object without compromising earlier parts.

3) States

The user can dynamically change the image associated with a visual object. In order to change states, the animation can be replayed, and during the playback, the user can select the desired state from a menu, and consequently the image to be displayed. The changes in the state are recorded and time-stamped with the rest of the animation. An alternative way is to change the states not while recording, but while the animation is stopped, which allows for more time and precision. Specifically, users can navigate to a particular point on the timeline and select the desired state at that time.

Switching states allows to change the visual appearance of a region, for instance changing the color of a light from red to green, the text displayed on a screen, or the different expressions of a character. Flipbook animations can easily be constructed in this way.

E. Implementation

A software prototype of *idAnimate* has been implemented for *iPad* multi-touch devices, using the Objective-C programming language and the Cocoa Touch Framework.

IV. COMPARISON STUDY

In order to increase our understanding of the trade-offs incorporated in the aforementioned animation system, we have conducted a comparison study between *idAnimate* and *K-Sketch*, a state-of-the-art general-purpose animation-sketching tool that relies on a similar technique (motion-by-example), but that does not require or use multi-touch.

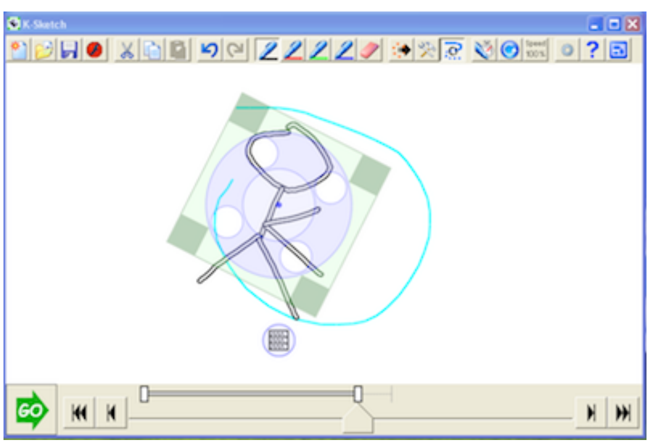

Figure 2.The *K-Sketch* animation tool, showing a selected figure together with its manipulator. The different tools are at the top, while the animation and playback controls are at the bottom.

Figure 3. The *idAnimate* animation tool, with the object selector displayed on the left, the object inspector on the right, the canvas in the center and the animation and playback controls at the bottom.

A. MATERIALS AND METHODS

B. The Tools

1) idAnimate

Please refer to the *idAnimate* section of this paper for more details, as well as figures 1 and 3 for the interface.

2) K-Sketch

K-Sketch (see Figure 2) is a general-purpose animation-authoring tool for Windows computers that is intended to be used with a keyboard and mouse or stylus. *K-Sketch* starts with an empty canvas where objects can be drawn. Objects are selected by drawing a loop around them while holding down the Alt key. When an object is selected, a manipulator (see figure 2) is displayed on top of it.

The manipulator allows users to apply distinct transformations on the object, such as translation, scaling, rotation or skewing, or to modify the orientation of the object relative to the movement direction. The manipulator can be repositioned on the object, effectively changing the center of rotation. To record, the user has to hold down the Alt key while transforming one of the above properties, and the system records the changes over time. *K-Sketch* also supports flipbook animation. At any moment in time the user can erase or draw something new, or decide to show or hide an object.

To record simultaneous motions of two or more objects, the user has to rewind and record new motions while the existing ones are playing. The user can add various transformations to a single object, such as a translation and rotation (like a spinning ball). In order to accomplish this, he has to first record the translation, rewind and subsequently record the rotation. By default the system overwrites the old motion with the new one. Alternatively, the user can select the “Fix Last Motion”. This option shows a menu with the different alternatives for combining the existing motion with the new one. For example, if we first translate and then rotate an object, *K-Sketch* will show three different ways to combine them: (1) overwrite, (2) translate

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-275-2

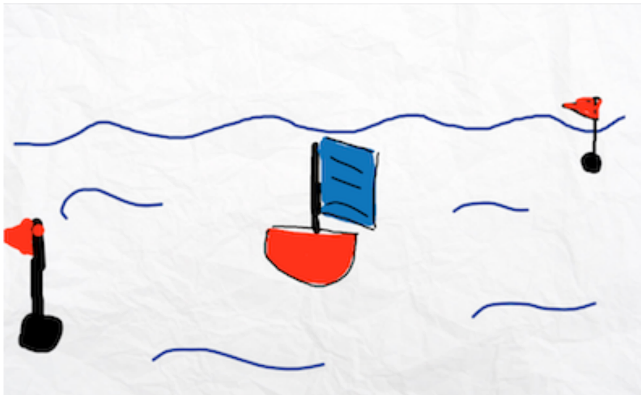

Figure 4. Scene for the sailing boat animation task.

and rotate around the center of the object, or (3) translate and rotate around the initial center. The user can preview all three options and make a choice. Last but not least, *K-sketch* supports recursion or animating of subparts (called Move Relative). To accomplish this, the user needs to select a subpart of an object that is already animated, and animate it with its independent handler. Unlike other features, this latter one is not available in *idAnimate.*

C. Study Approach

Two representative animation tasks were selected (see more details ahead) to allow for an explicit comparison. Each participant was asked to perform both tasks with both tools, randomizing the order across participants of which tool to use first. Before using each tool, participants were given a brief tutorial. The tutorials followed a script that was based on a slightly extended version of the car-exploding example [12] in the demonstration videos of K-Sketch. During the tutorial, participants were allowed to ask clarifying questions to the experimenter.

After the brief tutorial, the participants were shown a static image of the scene to create, and were explained verbally the script and details of the animation. The choice to use a static image instead of an actual animation was motivated by our objective to direct participants towards a specific outcome, while allowing them freedom to decide what exactly to create and which features to use.

The participants were given a tool with a blank canvas, and were asked to sketch the visual objects and animate them. They were encouraged to not spend time on creating good-looking and detailed graphics, but to create sketchy looking ones instead. Once a participant had finished completing both tasks with one of the tools, he or she would be given the brief tutorial on the second tool, and would be asked to repeat the same tasks but with this second tool.

D. Surveys and Interview

After completing the tasks with one of the tools the participants were asked to fill out a slightly modified Usability Satisfaction Questionnaire [13] (see more details in the results section). Upon having completed both tasks with both tools, participants were asked to complete a final survey. In this third survey, they were asked to directly compare the two animation techniques by specifying their preference (for various aspects of the techniques) on a scale from -3 (*idAnimate*) to +3 (*K-Sketch*), including 0 as the neutral or no preference. The aspects on which to compare included intuitiveness and playfulness (see the Results section for more details).

A five to fifteen minute semi-structured interview concluded the experiment. Subjects were asked to characterize the differences between the two tools, starting from the attributes used in the comparison survey.

Up to now, the quality of the animations that were produced has not yet been explicitly graded by either the participants themselves or an external reviewer. We are still considering whether or not this is worthwhile doing. Instead, within the comparison survey and post-session interview, participants were asked to argue why they would choose one or the other tool to create better quality animations, or to create them in less time.

E. Data collection

The participants were video-taped during the execution of the tasks, their responses to the questionnaires were collected on paper, and the semi-structured interviews were audio-recorded. As the focus of the study was on comparing both animation techniques, the participants were encouraged to answer mostly in terms of the animation techniques themselves and less in terms of the user interfaces of the tools (which were obviously different).

F. Subjects

Twelve subjects participated in the study, 4 female and 8 male. The age of the subjects ranged from 21 to 34 years. Six participants were PhD students, one an assistant technician in an immunology laboratory, while the rest were bachelor, master or recently graduated students. Table I provides detailed background information on the participants. None of the participants were expert animators, nor did they create animations on a regular basis. Two of the participants with a background in industrial design had minimal experience with Adobe Flash, while one had used Adobe After Effects (video special effects software) once. One of the Computer Scientists had created stop-motion animations in the past, while another had used Flash occasionally.

TABLE I. PARTICIPANTS

Background	Current Position	Participant count
Computer Science	PhD Candidate (3), Postmaster Student (1), Bachelors Student (1)	5
Bio molecular Sciences	Laboratory technician (1)	1
Pharmacy	PhD Candidate (1)	1
Industrial Design	PhD Candidate (1), Bachelors Student (1), Masters Student (1), Recent Masters Graduate (1)	4
Mechanical Engineering	PhD Candidate	1

G. *The tasks*

Unlike in other fields of research, there is no agreement or standardization in the domain of animations on what constitutes a representative task for comparing animation techniques or systems. For the purpose of the study we looked at common tasks as identified in the preliminary studies conducted by Davis et al. for K-Sketch (specifically, Translate, Scale, Rotate, Set Timing, and Move Relative), and selected two representative tasks.

1) *Task 1 – Angry Birds Game*

The scene is composed of three objects, an angry bird, and two rocks or blocks. The bird has to fly towards the first block and hit it. As a result of the collision, the bird flies away modifying its original trajectory, and the first block falls down. While falling down, the falling block hits the second block, which in turn falls down, like a falling domino. See figure 1 for more details.

2) *Task 2 – Sailing boat*

There is a boat on the sea that has to travel from point A to point B. Point A is in the bottom-left corner, close to the observer. Point B is in the top-right corner of the screen, closer to the horizon. The boat has to travel from point A to point B. Because point B is further away, the boat has to become smaller as it progresses along the path. Since it is a boat floating on water, the boat has to rock slightly as it moves. Finally, the flag on the boat has to wave. See figure 4 for more details.

H. *Devices*

K-Sketch runs on Windows and *idAnimate* on an iPad. To compare both tools in the most similar way possible, we used a Fujitsu-Siemens Stylistic ST-6012 Tablet PC equipped with a stylus for *K-Sketch*. The Tablet PC had buttons around the display, one of which was mapped to the Alt key, since *K-Sketch* requires frequent use of this key. For *idAnimate,* an iPad 2 was used.

V. RESULTS

A. *Task Analysis*

1) *Data Analysis*

The time spent by each participant in each of the tasks was divided into the time used for drawing graphics and the time actually used for animating, both measured in seconds. For the purpose of the study only the time used for animating was considered. As the distributions of the observed times were clearly skewed, we used the logarithm of the observed times in the statistical analyses, as they more closely followed a normal distribution, which is required by traditional statistical procedures such as the t-test. Although the original intention was to include measurements of error rates, we found this too difficult to operationalize, as it was too hard to decide on what constituted an error and what did not. It was not straightforward to determine if a user was correcting an error, refining an animation, or just playing around. Asking them explicitly would obviously have disturbed their process.

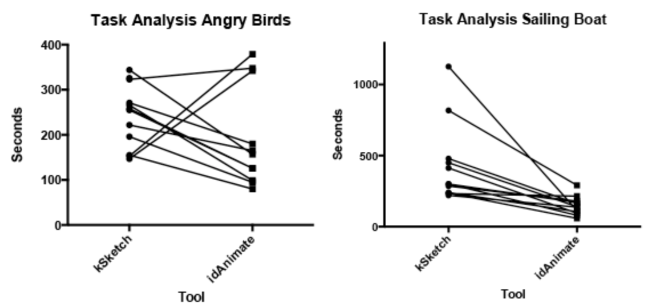

Figure 5. Observed times for the Angry Birds and Sailing Boat tasks

2) *Task 1 – Angry Birds*

All of the participants were able to complete the Angry Bird task with both tools. Table II and Figure 5 show that the animation times were very similar on average. Some subjects performed faster with *idAnimate* than with *K-Sketch*, while for other subjects it was just the other way around. The participants obviously did not perform as a homogeneous group.

TABLE II. TIMING STATISTICS FOR THE ANGRY BIRDS TASK.

Tool	Min/Max	Std. dev.	Mean	Std. Error of the Mean
K-Sketch	147/344	69.335	243.0	20.015
idAnimate	80/410	123.327	208.916	35.601

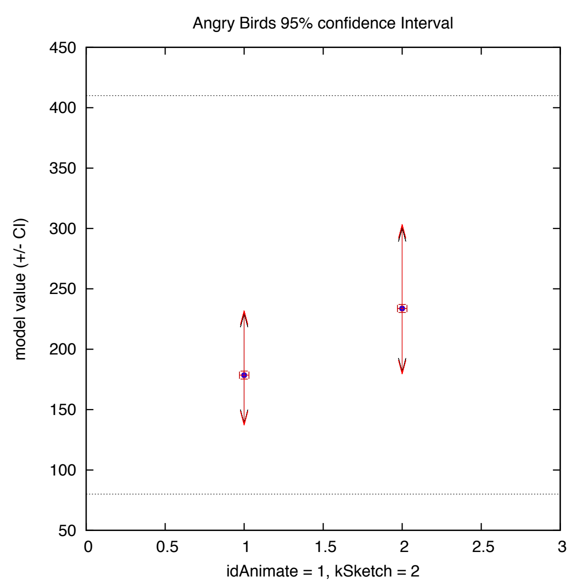

Figure 6. Overlapping 95% confidence intervals for the time to complete the Angry Bird task for both tools.

TABLE III. T-TESTS FOR THE RESULTS OF THE COMPARISON SURVEY

Aspect	Mean Difference	Significance	95% confidence interval (min/max)
User Satisfaction	-1.25	0.009 (Yes)	-2.112/-.387
Pleasant Experience	-1.33	0.020 (Yes)	-2.067/-.599
Easiness	-1.750	0.000 (Yes)	-2.522/-.9778
Intuitiveness	-1.916	0.000 (Yes)	-2.704/-1.128
Playfulness	-2.000	0.000 (Yes)	-2.605/-1.394
Animate Rapidly	-1.666	0.009 (Yes)	-2.826/-0.506
Animate Precisely	0.667	0.255 (No)	-.555/1.888

Assuming unequal variance and performing a T-test, we obtained: T(21) = 1.4108 (p=0.171326). There is hence insufficient evidence to reject the null hypothesis that both tools require the same time to complete the task. Figure 6 shows the estimated 95% confidence intervals for the untransformed average times. Note that the intervals are not symmetric since they were determined on the logarithm of the time, and then reverted to actual seconds. We observe that both intervals overlap, which agrees with the fact that the difference in average performance is not significant.

3) *Task 2 – Boat Animation*

All participants were able to complete the boat animation with both tools, except for one participant who after trying a number of times to merge two motions with *K-Sketch*, needed help from the experimenter to do so. Table IV shows the descriptive statistics, while the individual times are plotted in Figure 5. Figure 9 displays the 95% confidence intervals for the average times, while Figure 8 shows the 95% confidence interval for the time difference**.** We observe that the average times are considerably different in this case, being smaller for the multi-touch tool. Assuming unequal variance we performed a T-test on the transformed times with the following result: T(21) = 5.1248 (p=3.89654e-05), with |T| > 2.0748 (p=0.05), indicating a significant difference. Because of the size of the effect, and the agreement between the subjects, the power of the test was estimated at β = 0.999, despite the small number of participants. The multi-touch tool is significantly faster for this task. After reverting the time back to seconds, we find a 95% confidence interval for the time difference between 116.5 until 404.9 seconds (see Figure 8).

TABLE IV. TIMING STATISTICS FOR THE BOAT TASK.

Tool	Min/Max	Std. dev.	Mean	Std.Error of the Mean
K-Sketch	220/11260	227.703	425.250	80.166
idAnimate	59/291	63.981	145.833	18.469

TABLE V. WILCOXON SIGNED RANK TEST RESULTS FOR THE SUS QUESTIONNAIRE.

Dimension	Significance	Positive / Negative / Ties.	Standardized test statistic
Overall Easiness Satisfaction	0.046 (Yes)	1/9/2	-1.996
Intuitiveness	0.043 (Yes)	2/8/2	-2.019
Learnability	0.070 (No)	2/8/2	-1.813
Playfulness	0.007 (Yes)	0/9/3	-2.687
Comfortable	0.088 (No)	2/5/5	-1.709
Simplicity	0.117 (No)	1/7/4	-1.567
Pleasantness of the interface	0.026 (Yes)	1/9/2	-2.228
Recover from Mistakes	0.454 (No)	6/3//3	.749
Efficiency	0.796 (No)	3/4/4	.258

B. *Analysis of Subjective Ratings*

1) *Standard Usability Questionnaire*

Figure 10 shows a summary of the results of the modified Usability Satisfaction Questionnaire, including the mean, the standard error of the mean, and the questions asked.

We performed non-parametric repeated measures Wilcoxon Signed Rank Tests on the observed ratings in order to identify significant differences (see Table V). We observe that users have a preference for the multi-touch technique in terms of Intuitiveness, Overall Easiness, Satisfaction, Playfulness, and Pleasantness of the interface. There are some trends mentioned in Table V that cannot be decisively concluded based on our limited experiment, more specifically, *idAnimate* tends to be preferred in terms of Comfort and Learnability, while *K-sketch* tends to be preferred in terms of Efficiency and Recovery from Mistakes. The observed differences for the subjective attributes that are not included in the table were not significant.

2) *Comparison Questionnaire*

The comparison questionnaire asked users to indicate for a number of aspects which tool they preferred. If they selected -3 that would mean they had a strong preference for *idAnimate* for such an aspect, while +3 would imply a similar preference for K-Sketch. The 0 was included as neutral or no particular preference for any of the tools. As shown in Figure 7, *idAnimate* was preferred over *K-Sketch* on all attributes except one, i.e., Precision. All participants but one would use the multi-touch technique to animate something rapidly, but when judging precision there was no clear agreement. Eight out of the twelve users would select *K-Sketch* for animating something with precision, two would select *idAnimate*, and two did not have a preference. See Figure 7 and Table III for additional details.

VI. DISCUSSION

A. Speed versus Accuracy

The quantitative data allows to conclude that the multi-touch technique (*idAnimate)* is significantly faster then K-Sketch in case of the Boat animation, but not in case of the Angry Birds animation. This is due to the fact that the latter task mainly required users to synchronize the animations of different objects, while the required animations for each object individually were rather basic In most cases, users only moved the bird, and barely changed it's orientation as it flew through the air. Most subjects used only a single rotation of the boxes to make them fall down. Therefore, this task did not reveal differences in performance between both animation techniques. The observed differences in animation times can be attributed largely to the differences in user interfaces, and to how easily objects could be selected.

The sailing boat task, on the other hand, discriminated very well been the two tools. In this task at least three parameters of the boat had to be animated, including the location, the orientation and the size. In this task participants performed significantly faster with the multi-touch animation tool. This was due to the fact that it was possible to animate these three parameters simultaneously, as opposed to having to change these parameters individually and in succession. This task posed different challenges for both techniques.

With the multi-touch tool users had to act out a complex movement with one hand, and they would not always get it right the first time. In fact they would normally perform an initial attempt just to see how it looked, and then refine it on a second or third attempt. Using a single hand to move, scale and rotate an object simultaneously in a controlled way is quite difficult, and very often the accuracy suffered, which had an evident influence on the quality of the output. This was described in the following way by one of the participants during the interview:

"*It's definitely a downside of the iPad app, you have to do everything in one go… …three things at a time can be quite challenging."*

"*With idAnimate you can do it more fluently, but you have to practice your movements… … to act out something with your fingers is something new and you have to get used to it*".

"*idAnimate felt more easy to drag things, especially to make the boat. The multi-touch was easier even though it didn't create the exact result that I wanted because it was less precise*"

Despite this drawback, the technique proved to be sufficiently accurate for "quick and dirty" animations.

Participants were slower with *K-Sketch* but in general more accurate. There are two factors that influence this tradeoff. First, the stylus is simply more accurate than the finger, as the finger gets in the way, is thicker, and essentially provides less fine control, both for sketching and for animating. But also the animation technique itself had an effect. With K-Sketch the user animates each property

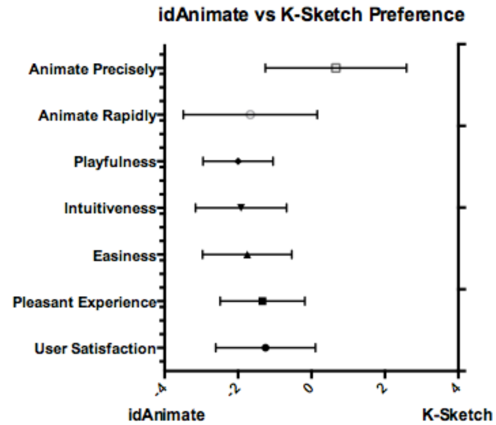

Figure 7. Results from the comparison questionnaire between *idAnimate* (on the negative side) and *K-sketch* (on the positive side) (mean and standard error of the mean).

successively, building the animations step by step. Consequently, the user has more precise control over each of the independent gestures.

"*The quality was certainly higher with the second application (K-Sketch), because of the pen especially, and because of the complexity of the movements*"

"*With K-Sketch you have all the gestures, like the resize, the scale or the rotation separately, and you can control them also independently, and that also gives you a little bit more control*"

"*Your expressive power with K-Sketch is finer. You can combine the movements better, you can make more movements at a time, and be more precise, but especially because you can do it step-by-step*".

But this can also backfire. When animating a particular property, it was not possible to observe the animation of the previous property, implying that users had to act blindly. To assist in such a task, *K-Sketch* displays the trajectory of the previous motion path that gets filled up as the time goes by, showing the time-varying position of the object. This was often found hard to interpret. Additionally, when a user had completed the animation of a particular property, he had to

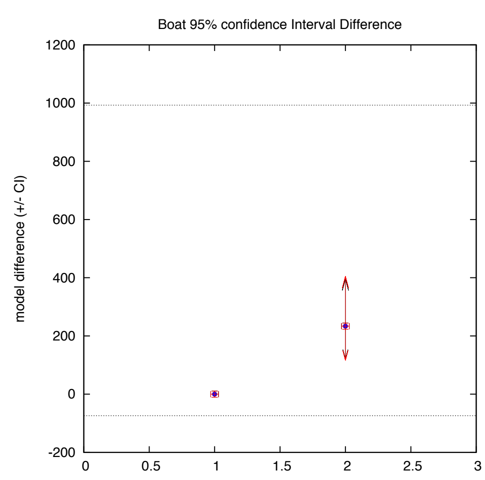

Figure 8. Sailing boat task 95% confidence intervals difference.

merge it with the previous one by selecting an option from a number of alternatives, for example with different centers of rotation. Users often felt unsure of which option matched their expectations best. As more and more motion paths and transformations were applied, the process became more and more difficult to predict. This was found overwhelming to some novices, especially those without a design-related background. Furthermore, in some cases the expected choice was not even there, and participants had to undo and try again because they could not figure out what went wrong.

"*The frustrating part of K-Sketch is that when you want to combine animations, the more you make the more scenarios you get, and that is a bit confusing*"

K-Sketch also offers the possibility to animate a subpart of an animation. This feature was found really useful, and most participants used it to animate the flag on the boat. With *idAnimate* most participants used the flipbook technique instead of animating the flag as an independent object. However, sub-animating with *K-Sketch* was sometimes also found confusing, due to the fact that the larger animation moves continuously as you animate the subpart, and you do not follow it with the stylus, but instead remain in the same place as where you begun animation as if the object was not actually moving.

"*With the pen there is a moment that you are animating and the object continues moving and this is a little bit chaotic*."

B. *Intuitiveness and Playfulness*

Certainly, animation-sketching tools are all about making animations quickly, and about minimizing the trade-off between quality and speed. The threshold for embarking on the task and the simplicity to achieve it, how funny and engaging the overall experience is, are also important. This can be particularly influential in certain contexts, such as using animations to facilitate workshops or brainstorms, and for certain users such as children, seniors, or non-animators in general.

Participants found the multi-touch technique to be significantly more intuitive and easy to use. This was because the underlying conceptual model was easy to understand and very direct, a result of the mapping between the multi-touch gestures and the properties of the objects. Such mapping resembles object manipulation in the physical world. By manipulating objects with the hands directly, users are able to very easily prepare a plan of actions to achieve a desired result.

"*With the multi-touch tool you make the entire movement, it's easier, as you do not have to think, you just let yourself go*".

Furthermore, the multi-touch tool was significantly more playful and more fun. In fact, it was so playful, that three out of the twelve participants created additional content in the animations, such as flying birds around the boat, or making the floating buoy move as if it was actually floating.

"*I animated the floating buoy with idAnimate because it was attractive, it was like a game, it facilitated the act.*"

"*idAnimate felt easier, more fun, more like a game. You can start to do silly things with your own fingers, while the other one feels more professional*".

We believe that this perception of playfulness was due to the combination of how tactile and physical the multi-touch interface feels, together with the simplicity of the technique, its immediate feedback and the attractive graphical design of the tool.

In the case of *K-Sketch, u*sers had to decompose their actions into a series of steps that the system supported in order to carry out their goal. They had to think ahead which motion path they were going to apply, which transformation would follow, how they would be combined, etc. This required more cognitive effort, and relied on a good understanding of the underlying conceptual model. This model was difficult to understand and operate by some users.

"*The interaction style of K-Sketch was more complex,*

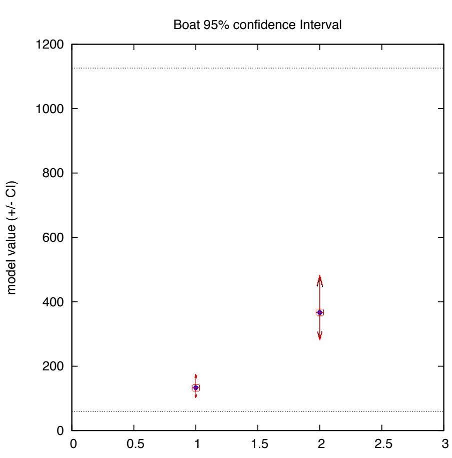

Figure 9. Non-overlapping 95% confidence intervals for the time to complete the Sailing boat task with both tools

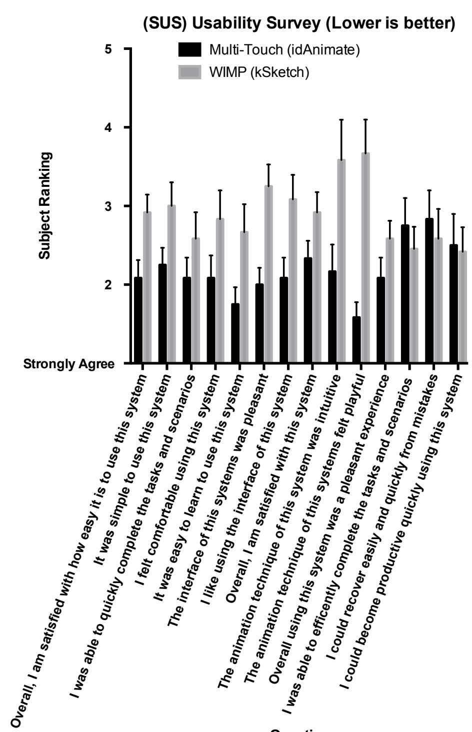

Figure 10. SUS results, mean and SEM.

less direct or so… …I had to think more often with it.”

However, for experienced users this may actually be an advantage. If you are able to plan the actions based on this decoupling of transformations, the technique provides more control.

C. *Different contexts*

In summary, we argue that the preference for either tool depends on the context. *idAnimate* is most suited in case animations must be created and discarded quickly, if the details are not important, and has the potential to be extended to a situation where multiple people work simultaneously on a single animation. *K-Sketch* is better suited for individual activities, where the level of detail and control required is more substantial (although still significantly below that required of professional animation tools). Both techniques can support novice users with minimal instruction, although *idAnimate* appears to have a lower threshold since its underlying conceptual model is easier to understand.

“*I would pick the second one (K-Sketch) if I had to make an application for a corporate setting or something like that… …For a brainstorm the iPad application is more convenient. You just draw it and move it.”*

VII. Conclusion and Future Work

In this paper, we have presented a general-purpose animation-sketching system for multi-touch devices, called *idAnimate.* We have presented a comparison study aimed at increasing our understanding on what trade-offs and differences this multi-touch animating system poses when compared to system that does not rely on multi-touch. The experiment supports the conclusion that the multi-touch animation-sketching system (*idAnimate)* is perceived as significantly more intuitive, playful and easier to learn. Additionally, we have observed that *idAnimate* may be significantly faster in cases where the user has to apply multiple simultaneous transformations on objects, and that in comparison to K-*Sketch* it generates animations with lower precision.

These results and observations are being used to guide the next iteration of the tool. From the study we have learned that it is really difficult to isolate specific aspects of the tool in order to benchmark the technique, which has lead us into using for the follow up study a longitudinal approach, focused more on qualitative aspects of the experience.

VIII. Acknowledgements

The work presented was made possible through a grant from the Dutch Government under the IOP-IPCR Program.

References

[1] A. Russell. 2010. ToonTastic: a global storytelling network for kids, by kids. In Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction (TEI '10). ACM, New York, NY, USA, 271-274

[2] A. Watt and M. Watt. 1992. Advanced Animation and Rendering Techniques, Theory and Practice. Addison-Wesley Professional.

[3] A. Wong, N. Marcus, P. Ayres, L. Smith, G.A. Cooper, F. Paas, and J. Sweller. Instructional animations can be superior to statics when learning human motor skills, Computers in Human Behavior, Volume 25, Issue 2, March 2009, Pages 339-347, ISSN 0747-5632, 10.1016/j.chb.2008.12.012

[4] Adobe Flash Technologies . Retrieved March 2013. http://www.adobe.com/products/flash.html

[5] B. Tversky. J. Bauer Morrison, and M. Betrancourt. 2002. Animation: can it facilitate?. Int. Journal of. Human-Computer. Studies. 57, 4 (October 2002), 247-262.

[6] D. Ceylan and T. Capin. A Multi-Touch Interface for 3D Mesh Animation. Bilkent University.

[7] D. Ozcelik, J. Quevedo-Fernandez, J. Thalen, and J. Terken. 2011. On the development of electronic design tools and associated guidelines for supporting the early stages of the design process. In Proc. of the Second Conference on Creativity and Innovation in Design (DESIRE '11), ACM, New York, NY, USA, 115 – 126.

[8] E. Sohn and Y.C. Choy. Sketch-n-Stretch: sketching animations using cutouts. IEEE Computer Graphics and Applications, *IEEE* , vol.32, no.3, pp.59,69, May-June 2012.

[9] J. Davis, M. Agrawala, E. Chuang, Z. Popovic, and D. Salesin. 2003. A sketching interface for articulated figure animation. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '03). 320 – 328.

[10] J. Quevedo-Fernández and J.B.O.S. Martens. A User-Centered-Design Perspective on Systems to Support Co-located Design Collaboration. Proceedings of the 15th International Conference on Human-Computer Interaction. *In Press*. Springer LNCS. 2013. In Press.

[11] K. Takayama and T. Igarashi. 2007. 2d Animation Authoring System with FTIR Multi-touch Table. Interactive Tokyo.

[12] K-Sketch Car exploding video tutorial. Retrieved March 2013. *K-Sketch* car exploding tutorial - http://youtu.be/UYLR5Nf4mWs

[13] J.R. Lewis. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. International Journal of Human-Computer Interaction. 1995, Volume 7, 57-58.

[14] M. Thorne, D. Burke, and M. Van de Panne. Motion doodles: an interface for sketching character motion. In Proc. of ACM SIGGRAPH 2004. ACM, New York, NY, USA. Article 24.

[15] PhotoPuppet HD iOS Application. Retrieved March 2013. http://itunes.apple.com/us/app/photopuppet-hd/id421738553?mt=8

[16] R.C. Davis, B. Colwell, and J.A. Landay. 2008. K-sketch: a 'kinetic' sketch pad for novice animators. Proc. Of Human factors in Computing Systems (CHI '08). ACM, New York. 413-422.

[17] R.M. Baecker. 1969. Picture-driven animation. In Proceedings of the May 14-16, 1969, spring joint computer conference (AFIPS '69 (Spring)). ACM, New York, NY, USA, 273-288. 1969J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon.

[18] T. Moscovich. 2007. Principles and Applications of Multi-touch Interaction. PhD Dissertation. Department of Computer Science at Brown University.

[19] Z. Obrenovic and J.B.O.S. Martens. 2011. Sketching interactive systems with Sketchify. ACM Trans. Comput.-Hum. Interact. 18, 1, Article 4, 2011.

[20] B. Shneiderman. Creativity support tools: accelerating discovery and innovation. Commun. ACM 50 (2007), 22-32.

[21] M. Resnick, B. Myers, K. Nakakoji, B. Shneiderman, R. Pausch, T. Selker, and M. Eisenberg. Design Principles for Tools to Support Creative Thinking. Institute for Software Research. Paper 816. (2005)