Prototyping Video Games with Animation

Richard C. Davis
School of Information Systems
Singapore Management University
Singapore
rcdavis@smu.edu.sg

***Abstract*—This paper outlines a proposed design for PlaySketch, a new video game storyboarding system. PlaySketch borrows ideas from the K-Sketch animation sketching system, which allows short animations to be created in minutes or seconds. We build on K-Sketch in four ways. key frame animation capabilities, a branching timeline view, microphone and web-cam support, and hooks to connect to online game design documents.**

Keyword-component: storyboard, animation, sketching, video game, pen based user interfaces

I. Introduction

As the market for video games becomes more competitive, designers are looking for ways to produce better designs faster. Rapid prototyping is now widely regarded as an essential part of the video game design process [10]. Animation sketching is a novel prototyping method that could revolutionize game design by making it possible to create animated prototypes in minutes or seconds. Furthermore, these prototypes are flexible enough that they can be created and modified quickly during discussions with other designers or evolved into higher-fidelity prototypes. We are using animation sketching techniques to build a new storyboarding system called PlaySketch that will support the video game design process. PlaySketch will allow game designers to incorporate high-quality graphics and audio, work with complex, branching timelines, and produce better design documents to guide game development.

Many game designers begin their work with paper prototypes [10]. By acting out sequences of game play with small paper cut-outs, a group of designers can develop a rough idea for a game. Often, this results in a pitch document (also called a high concept document or a one-pager) that gives a brief overview of the game and describes any important features. Over time, this document evolves into a more detailed game design document that gives a longer synopsis of the game and goes deeper into the characters, goals, game play, and user interface. It may also describe the music and sound effects in the game. The game design document is continually revised and sent to the technical team, which uses it as a reference for development.

Storyboards are a common tool for fleshing out aspects of a design [9]. Small scale storyboards may focus on the play-by-play of the game in chunks of about 30 seconds. Larger scale storyboards will describe higher-level aspects of the game that take place over longer periods of about five minutes. By showing these storyboards to others, designers can get feedback and modify their design. Storyboard frames can also be modified during meetings or re-arranged quickly (if they are kept on separate sheets) to evaluate different possibilities.

Storyboards are a powerful tool, but they have two significant limitations. First, they are static, and they cannot effectively communicate many design details for a dynamic video game. This limitation prompts many designers to create animatics, which are storyboard frames presented with a sound track and some simple animation. The motion and sound in these animatics communicate game design ideas much more effectively than drawings and text. Alternatively, designers may produce interactive prototypes, because they are more enticing and communicate game ideas better than either storyboards or animatics [7]. However, animatics can take hours to produce, and interactive prototypes can take days, making these methods unsuitable for early-stage prototyping.

The other disadvantage of storyboards is that they are linear. Time in a storyboard always advances in a straight line, but the timeline of a game can branch depending on a player's actions. Designers need to explore these branches and work out story details together. Some designers use flowcharts for this purpose, but a flowchart cannot easily capture the visual nature of a video game, and it shares a storyboard's limitation of being a static medium.

Because storyboards are static and linear, they can often be misinterpreted by a development team. It may take several storyboard frames and a page of text to explain how art, motion, and sound combine to create a single game event. Such events can often be explained quickly and more effectively with a single animation or short interactive prototype, but producing these prototypes for every game event is too costly. Designers need a prototyping tool that combines the advantages of storyboards, animatics, and interactive prototypes to produce more effective game design documents.

Our storyboarding system, PlaySketch, will preserve the simplicity of storyboards, but it will use animation to make storyboards dynamic, and it will support branching timelines. This will give professional game designers a powerful new way to get ideas out of their heads quickly and into a group's consciousness where ideas can be refined. Furthermore, designers will be able to embed these animated storyboards in online game design documents and replace rough graphics and audio with high-quality versions as the design evolves. Animated storyboards will therefore serve as a reference, facilitating communication between designers and developers during the game development process. This paper presents a preliminary design for PlaySketch.

This work is supported by Singapore Management University and by the Singapore-MIT GAMBIT Game Lab.

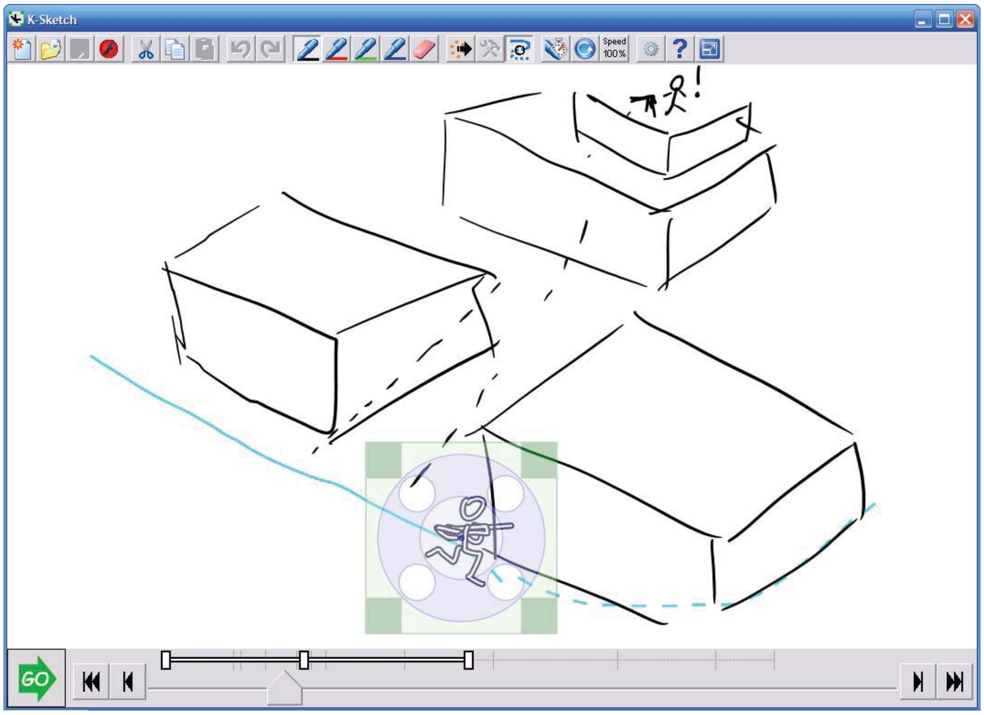


Figure 1: The K-Sketch User Interface. Users sketch and move objects in the center canvas. The slider bar at the bottom indicates the current moment in time. The symbols above the slider show the time span of motions applied to the selected object. Users create animations with a series of instantaneous movements or by demonstrating motions in real time. The blue lines show a demonstrated motion path for the character.

II. Related Work

Storyboarding has long been regarded as an important skill in the entertainment industry [12], and researchers have made other attempts to enhance storyboards. Some take advantage of traditional storyboarding behavior to speed up the production process (for example, by generating animations from storyboards [5]). Other researchers have sought to improve the process of creating storyboards for interactive systems. DEMAIS is a pen-based system with a visual language for designing multimedia content [3, 4]. StoryCanvas is a more conventional system that helps designers produce storyboards for interactive dramas with complex story lines [11].

Like DEMAIS and StoryCanvas, our PlaySketch system seeks to improve the storyboarding process for interactive systems. Like DEMAIS, PlaySketch is a pen-based system, but PlaySketch uses demonstrated animation rather than a static visual language, making it better suited to the action sequences found in video games. Our focus on action sequences also distinguishes PlaySketch from StoryCanvas, which focuses on plot lines. Like PlaySketch, however, StoryCanvas does seek to manage the non-linear structure of interactive stories.

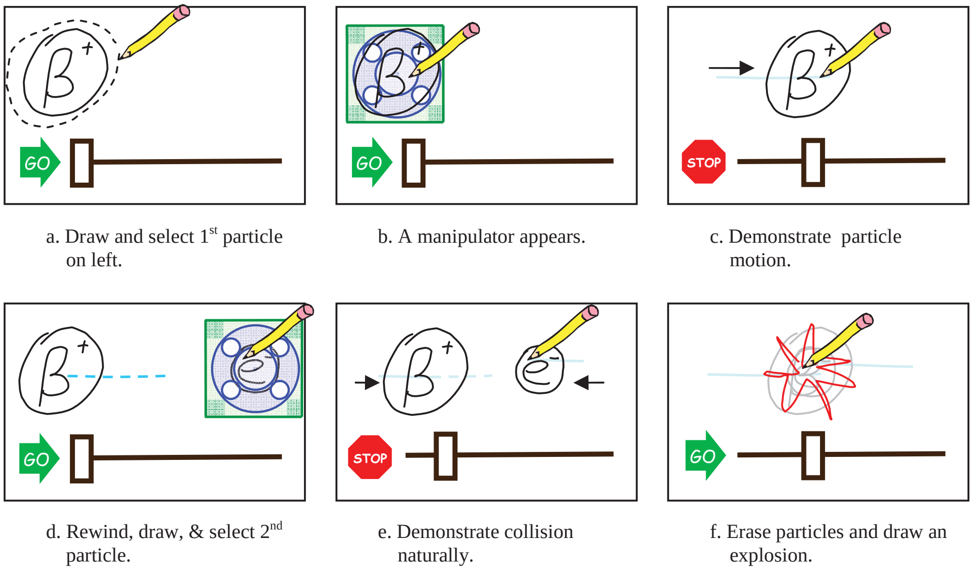
Animation sketching systems speed up the animation process by allowing designers to quickly specify rich motions with simple commands or gestures. Such systems have existed for over forty years [2], but they are receiving increased attention due to the wider availability of powerful computers with pen or multi-touch display surfaces. Some, like ASSIST [1], use physical simulation to generate motions. Others capture real time demonstrations of motion, as in K-Sketch [6] or As-rigid-as-possible shape manipulation [8]. Our system is based on K-Sketch, and we explain how this system works in the following section.

III. Animation Sketching with K-Sketch

K-Sketch is the animation sketching system that forms the foundation of PlaySketch. With K-Sketch, designers can make short, rough animations in seconds by drawing objects on a tablet computer and demonstrating their motions in real time. K-Sketch uses fluid pen input and is highly tuned to make common operations easily accessible. With as little as 30 minutes of practice, animating with K-Sketch can feel as natural as drawing.

The K-Sketch interface is shown in Figure 1. Designers begin by drawing a scene in its initial state. The animation is then created through a series of editing steps. Some edit operations move objects instantaneously at the time indicated by the time slider bar, and other operations demonstrate motions in real time (see Figure 2). This simple interface allows rough but complex animations to be prototyped very quickly.

Conventional animation tools have complex timelines that show all the transformations applied to each object over time.



a. Draw and select 1st particle on left.

b. A manipulator appears.

c. Demonstrate particle motion.

d. Rewind, draw, & select 2nd particle.

e. Demonstrate collision naturally.

f. Erase particles and draw an explosion.

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

In contrast, K-Sketch has a simplified timeline that shows only the most important events, highlighting those related to the currently selected object. In addition, a motion path appears when a designer records a motion. This motion path serves as a reminder of how an object moves. It can also be selected and modified to change the trajectory of an object or copied to move other objects in the same way. These timeline and motion path tools help designers quickly modify animations as a design evolves.

K-Sketch has been released to the public and can be downloaded online from www.k-sketch.org. It is already proving itself as a prototyping medium. In particular, one research study showed how children can use K-Sketch to prototype video games [1].

IV. PLAYSKETCH MODIFICATIONS

PlaySketch will be implemented on top of K-Sketch by adding key frame animation capabilities, a branching timeline view, microphone and web-cam support, and hooks to connect to online game design documents. Key frame animation, was requested by designers during our exploratory research. Some designers are uncomfortable using K-Sketch because demonstrating motion in real time is unfamiliar. These users are more comfortable using key frames to define some types of motion.

The most visible addition to K-Sketch will be the branching view, shown in Figure 3. The view will show one or more scenes, which contain a progression of thumbnails moving from left to right, branching or merging at various points. The thumbnails in this view represent scene fragments, which are short animations that can be edited independently (or semi-independently) from others. Selecting a fragment in the branching view will cause it to become editable in the main canvas. Branches can then be created by issuing a *Branch* command within an existing fragment. This will split the current fragment into two at the branch point (if necessary), create a copy of the fragment after the branch point, and add branch and merge connectors before and after the new fragment. Other operations, such as merges and deletions, will be done through direct manipulation of the thumbnails and connectors in this branching view.

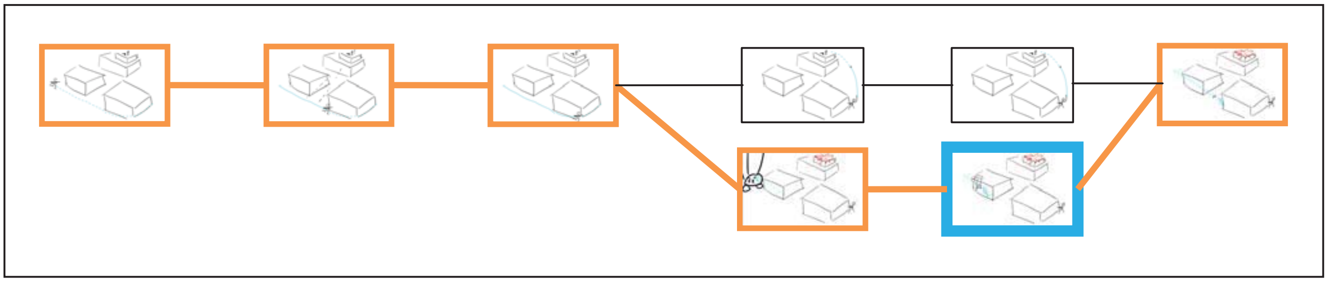
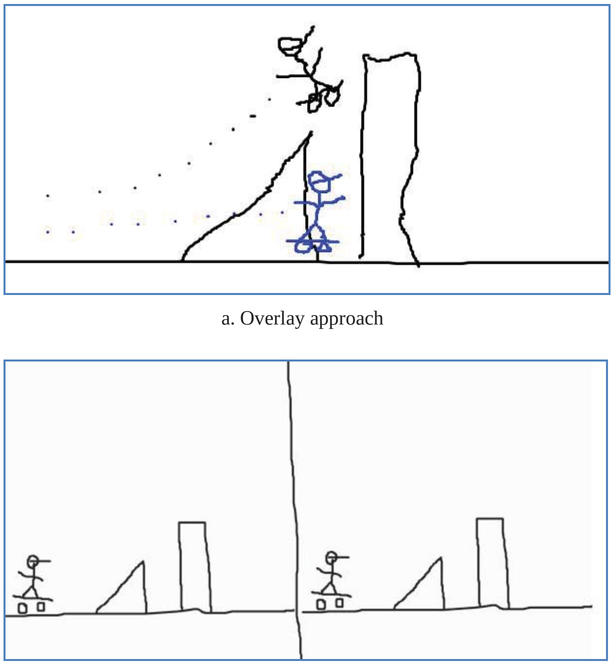


Figure 3: A rough sketch of the PlaySketch branching view. The current scene fragment is highlighted in blue. Designers can use pen strokes to quickly select paths through the network of fragments (highlighted in orange). Playing the storyboard advances through a particular path, allowing designers to evaluate that path or show it to colleagues.



a. Overlay approach

b. Split-screen appraoch

Figure 4: Two approaches to viewing timeline branches. Both were suggested by designers during our exploratory research.

Branches can serve both as a memory for design alternatives and as a repository for different story paths. It is therefore essential to provide easy ways to view different paths through the story. With a stroke of the pen, designers will be able to select a sequence of fragments for playback. During our exploratory research, some designers also requested the ability to view different sequences simultaneously for comparison. We are considering two approaches to simultaneous viewing: an overlay approach and a split-screen approach (see Figure 4).

Since audio is such an important part of animatics, PlaySketch will also allow audio tracks to be associated with scenes. Using their computer's microphone, designers will be able to record vocal sounds while demonstrating motions. They may also hum a background tune that plays throughout a scene. As a design evolves, audio designers can replace these vocal sounds with more polished audio tracks.

Some designers find that no computer interface can compete with the pleasing feel of physical tools. Because of this, PlaySketch will allow designers to take quick snapshots of pen-and-paper drawings (or other objects) using their computer's web cam. With automated tools for importing snapshots, adjusting light levels, and removing backgrounds, this process could be about as fast as drawing directly in PlaySketch.

Finally, PlaySketch will make it easy to incorporate storyboards into online game design documents. Our current plan is to make a PlaySketch plugin for the open-source XWiki platform[1]. This will make it easy to create and distribute game design documents with PlaySketch storyboards to a design team. It will also make it possible to evolve game design documents collaboratively over time, capturing refinements gradually as they are made and distributing them instantly to a team.

V. Conclusions

We have presented a preliminary design for PlaySketch, a new prototyping tool that preserves the malleability of storyboards, but uses animation to make them dynamic. PlaySketch also supports branching timelines to better support the needs of video games designers. We are basing PlaySketch on the K-Sketch animation sketching tool and adding key frame animation, branching timelines, audio support, web-cam capture, and wiki support.

References

[1] Alvarado C and Davis R. Resolving ambiguities to create a natural sketch based interface. In. *the International Joint Conference on Artificial Intelligence* 2001(2001), 1365-1371.

[2] Baecker R. Picture-driven animation. In. *the AFIPS Spring Joint Computer Conference* 1969(1969), 273-288.

[3] Bailey BP and Konstan JA. Are informal tools better?: comparing DEMAIS, pencil and paper, and authorware for early multimedia design. In. *the SIGCHI Conference on Human Factors in Computing Systems* 2003, ACM Press (2003), 313-320.

[4] Bailey BP, Konstan JA, and Carlis JV. DEMAIS: designing multimedia applications with interactive storyboards. In. *the Ninth ACM International Conf. on Multimedia* 2001, ACM Press (2001), 241-250.

[5] Chen M, Sheng FQ, and Wright DK. Sketching-out virtual humans: from 2D storyboarding to immediate 3D character animation. In. *ACM SIGCHI international conference on Advances in computer entertainment technology* 2006, ACM (2006), 349-351.

[6] Davis RC, Colwell B, and Landay JA. K-Sketch: a "kinetic" sketch pad for novice animators. In. *the SIGCHI Conference on Human Factors in Computing Systems* 2008, ACM (2008), 413-422.

[7] Hecker C and Gingold C. Advanced Prototyping. In. *Game Developers Conference* 2006(2006).

[8] Igarashi T, Moscovich T, and Hughes JF. As-rigid-as-possible shape manipulation. In. *the ACM International Conf. on Computer Graphics and Interactive Techniques* 2005, ACM Press (2005), 1134-1141.

[9] Rogers S, *Level Up!: The Guide to Great Video Game Design*. Wiley, 2010.

[10] Schell J, *The Art of Game Design: A Book of Lenses*. Morgan Kaufmann, 2008.

[11] Skorupski J. Storyboard authoring of plan-based interactive dramas. In. *Proceedings of the 4th International Conference on Foundations of Digital Games* 2009, ACM (2009), 349-351.

[12] Zhu H, Suarez F, and Lu X. Comparison of animation storyboard education in China and the United States. In. *ACM SIGGRAPH ASIA educators programme* 2008, ACM (2008), 1-5.

[1] www.xwiki.org