

Electronic Placemat: A Sketching Application for Paper-and-Pencil Games

1. Introduction

Most research on computer-based sketching applications has focused on supporting design and office activities for adults. Sketch usability issues that concern children and educational games have not been addressed. Meanwhile, children's educational and games software have not taken advantage of sketching recognition to support children's activities, thus missing out on the practical and creative benefits of sketching, abilities that are naturally performed by children. I am proposing a research project called the "Electronic Placemat" to create a pen-based application to support paper-and-pencil games for children.

Using paper and pencil allows children to be more creative and exploratory and therefore allows children to be more conducive to learning than traditional educational software. Paper and pencil allow children to draw and write without constraining their imagination; children can draw what they want, when they want and how they want it to look. On the other hand, educational and games programs constrain children to one activity and one environment at a time. Also, these programs only support mouse-and-keyboard interactions, which are more tedious to use than paper and pencil. Of course, computers offer advantages in learning: error-correction, simulation, fast information retrieval, and distance collaboration. The Electronic Placemat will combine the best of both worlds.

In this yearlong research, I will create an application that has an interface that is just a blank sheet, so that it resembles paper. Interactions with the interface will be modeled after paper and pencil with the computer providing support for user actions. For example, after the grid of dots for dots-and-boxes has been drawn, the computer would recognize that dots-and-boxes is being played, and during the game the computer would indicate who is winning. The goal of this research is to gain insight on how to infer user intentions from sketches when playing a game.

In the next sections, I describe the impetus for the research and other related work. Next, I explain the properties that the application should have and the games that the application will support. Then I describe my timeline for accomplishing this research. I will conclude with future research possibilities that can be done with the Electronic Placemat.

2. Impetus for the Research

Research in pen-based computing has been around for more than 50 years. In 1962, Ivan Sutherland created SketchPad, the first program that supported drawing of images right on the computer screen. In 1966, Gabe Groner invented the first sketch recognizer called GRAIL. For decades, the focus of research has been the improvement of the accuracy and robustness of sketch recognition technologies. But recently, research on how to apply sketch recognition in practice has been steadily growing. Some of these research projects include the Electronic Cocktail Napkin (Gross and Do, 1996a), SILK (Landay and Myers, 1995), DENIM (Lin, Newman, Hong, and Landay, 2001), Teddy (Igarashi, Matsuoka, and Tanaka, 1999), and Flatland (Igarashi, Edwards, LaMarca, and Mynatt, 2000). These research projects have focused

on applying sketch recognition on design and office activities for adults, such as architecture, user interface and web design, creation of 3D models, and support for informal office activities. The Electronic Placemat departs from this trend by supporting a different informal activity, games, and widening the target audience to include children.

The impetus to create a computer-based sketching program that supports paper-and-pencil games is twofold. First, we can learn how to apply sketch recognition in a different domain from design and office activities. This is beneficial because there are different usability issues between supporting design and office activities and supporting games. Second, paper-and-pencil games can be better supported by a pen-based application in a portable platform. Current computer implementations of paper-and-pencil games do not support sketching and are not portable; therefore these implementations lose the essence of paper-and-pencil games. Unlike board games such as Scrabble, chess, and Monopoly, paper-and-pencil games are relatively simple and short; it is not uncommon to play different paper-and-pencil games on one piece of paper. Paper-and-pencil games are also played in arbitrary settings, in the car, in a restaurant, at any time. By supporting paper-and-pencil games in a portable pen-based application, the Electronic Placemat can better support paper-and-pencil games and at the same time provide insight into what kind of usability issues come up in supporting such an application.

Another benefit of creating the Electronic Placemat is its collaborative potential. Many popular children's educational computer games, such as "Where in the World is Carmen Sandiego?" and Magic School Bus games are played individually. This is largely due to the fact that, unlike video game consoles that allow for multiple players, most desktop computers only allow input from one user. The Electronic Placemat will be implemented in the Tablet PC. With a size and shape that resembles a clipboard or a pad of paper, the Tablet PC is easier to physically share between two or more people.

3. Objectives

3.1 Supported games

A diverse range of paper-and-pencil games were chosen to allow us to gain insight on various ways of interacting with paper. The games that the Electronic Placemat will support are tic-tac-toe, dots-and-boxes, missing letters, and hangman.

- *Tic-tac-toe*. The objective of this simple game is to create a row in a 3x3 grid. This game is easy to implement and will serve as a primer to the demands of supporting a paper-and-pencil game.
- *Dots-and-Boxes*. The objective of this game is to have the most boxes at the end of the game. During the game, each player takes turns connecting dots, which eventually will form boxes.
- *Missing letters*. Missing letters is a simple game where the player fills in the missing letters of a word. The player is given two letters in the beginning and end of the word to serve as clues. This game is being implemented because it involves letters and word recognition. It is also a good transition game to hangman. Interesting problems

that can be explored in this game are computer-generation of handwritten letters and automatic generation of words for the player to guess.

- *Hangman*. Hangman is similar to missing letters, but this time every time the player guesses a wrong letter, a part of the hangman is drawn. When the whole hangman has been drawn, the player loses. Challenges that will be tackled with this game include the challenges of missing letters game and generating a hangman that looks rough and hand-drawn.

3.2 User interface requirements

To address the problem of creating an interface that will naturally support paper-and-pencil games for children, the interface must fulfill the following requirements:

The user interface must be paper-like. This seems like a redundant requirement, but it is an important one. Before sketch applications such as the Electronic Cocktail Napkin, SILK, DENIM, and Damask (Lin and Landay, 2002), computing support for design in architecture, GUI, web site design, and multi-device user interfaces, respectively, forced users to create and to work on precise and definite drawings, which are detrimental when doing early-stage design where ambiguity, abstraction, and imprecision are important (Gross and Do, 1996b). Research on paper-like interfaces includes the Electronic Cocktail Napkin, DENIM, and Damask. The former has a notebook metaphor, while the latter two have a “paper and desk” metaphor (i.e., sketches are drawn on “paper” windows and these windows can be moved around on a “desk”). This project will strive to stay away from the current paradigm of graphical user interfaces (GUIs, i.e., the use of buttons, menus, text boxes, etc.) and try to replicate the type of interactions that are done with paper. By doing so, the benefits of playing with paper (using paper is more natural and exploratory) will not be lost. The questions that will be asked are: How can regular software actions, like saving, and printing, be invoked intuitively? Will this approach be easier to the user? How can the interface support the actions of the user?

The application must be portable. Paper-and-pencil games are inherently portable. These games are short and simple that they can be played on a napkin, a piece of scrap paper, or a paper placemat in a restaurant (hence the name). The Tablet PC was chosen as the platform on which to implement the Electronic Placemat because its size and shape, wireless support and long battery life make it more portable than other pen-enabled platforms, like the Macintosh Newton and desktop computers equipped with WACOM tablets. While handheld PDAs are arguably more portable than the Tablet PC, its screen size is too small and its text recognition is not as robust. Another benefit of the Tablet PC is the integration of the input and display. Gross (1994) reported that the central limitation of the Electronic Cocktail Napkin was the separation of the input and display¹, which required more hand-eye coordination from users (p. 21). With an integrated display tablet, there is no separation between input and display; the user uses a stylus to write on and interact with the computer. The Macintosh Newton and handheld PDAs have integrated displays, but the Tablet PC has the best support for handwriting recognition

¹ The Electronic Cocktail Napkin was implemented on Macintosh computers (display) equipped with WACOM pen tablets (input).

The user interface must recognize shapes and forms. The Electronic Placemat will support paper-and-pencil games by recognizing the shapes and forms in a game. One type of interaction that the Electronic Placemat might support is deducing which player has won. For example, suppose two children, Peter and Mary, are playing tic-tac-toe with EP and that Mary has formed a row; EP can draw a line across the row to indicate that Mary has won. This seems trivial, but providing this kind of support to different games poses some important questions: What kinds of sketches, diagrams, and forms, are common and different among different games? What mechanisms can we use to teach the application to recognize a particular shape?

4. Planned Approach

4.1 Phase I: Fall Quarter

The first phase of this project is to implement individual support for different games. A framework to support recognition of different game elements (e.g., lines, dots, and blank lines) will also be explored. By implementing individual games, it is my hope that I will gain insight on how to support multiple games in one unified interface.

4.2 Phase II: 1st half of Winter Quarter

The different games implemented in the first phase will be tested on children. Users will be assigned simple tasks. The users' interactions with the different games will be observed and usability issues will be noted. Users will be asked for open-ended responses to their experience. Specific questions about the different UI features will also be asked.

Also in this phase, the different implementations will be studied for common and different elements, which will help in integrating all the games into one user interface in the third phase.

4.3 Phase III: 2nd half of Winter Quarter to 1st half of Spring Quarter

In this phase, the individual games will be integrated into one user interface. The conceptual goal of this phase is to create a paper-like interface. This will be done by: (1) exploring how to determine which game is being played and (2) exploring other ways to invoke commands without buttons or menus.

4.4 Phase IV: 2nd half of Spring Quarter

In this phase, the effectiveness of the unified interface in supporting multiple games and spontaneous play will be tested. Again, tests will be conducted on children. Users' interactions with the different games will be observed and users will be asked open-ended questions about their experience. Specific questions about the different UI features will also be asked.

5. Future Benefits

I have presented here a project to create a paper-like pen-based program to support paper-and-pencil games for children. In this yearlong research, several games will be implemented and will be integrated into one paper-like interface. The issues that will be addressed are the creation of a

paper-like interface, and the recognition of shapes/forms and context. The creation of the Electronic Placemat opens up other research issues that, although will not be addressed in this project, are interesting that they are worthy of mention.

Human-like output

It will be interesting to create a way to interact with the user as if it is human. One way to do this is to respond through sketches and letters that look as if they were written by a human being. A feature that would benefit from this is a “computer vs. user” mode. For example, suppose Peter is playing hangman against the computer. Peter indicates to the computer that he wants to play hangman by drawing blank lines. The computer selects a word for Peter to solve and creates more blank lines depending on that word. It would be the computer’s job to draw the parts of the stick figure (the hangman) every time Peter makes a wrong guess. In all the cases in which the computer has to draw, the drawings will look informal and handwritten. There currently is among sketch applications a visual disconnect between human input and the output created by the computer—the former looks rough and informal, while the latter has a formal and functional look. One insight into solving this problem is EtchaPad (Meyer and Crumpton, 1997), an application that is used to create user interfaces with an informal look. Another solution is to have the Electronic Placemat learn the particular way that a user draws and writes. This can be done by storing the strokes associated with the individual letters that the child writes. Then when the Electronic Placemat needs to respond to the child, it can use the strokes that it has stored. If there is enough time, this would be the next feature to be added to my yearlong research.

Educational applications

The Electronic Placemat can also be used for education. For example, a feature can be added to learn the words that the child most often writes. This feature will be done by recognizing the words that the child’s writings and keeping a tally of the individual words. By doing this, the Electronic Placemat can use other words when the child is playing, thus increasing the vocabulary of the child. Another feature that can be added is learning what the child draws. When a child draws a picture, the Electronic Placemat will ask the child what he/she has drawn. The child will respond by writing what he/she has drawn, and the Electronic Placemat will associate the word with the drawing. Later, when the child is playing a word game, Electronic Placemat can use the child’s drawing as a clue.

Below are some examples of games that can be supported that can help educationally. If time allows, these games will be included in my research.

- *Wordbuilder*. In wordbuilder, the player is given a long word. The objective of the game is to create as many words as possible from the letters of the given word. Computing support and word recognition is beneficial in this game because the computer can check the validity of the words that the player wrote.
- *Flashcards*. In flashcards, the player writes down arithmetic problems and solves them. The computer helps the player by checking whether the answer is right or wrong.

Programming of other games using sketching

In this project, only a limited number of games will be implemented. But in the future, it would be interesting to research ways to create games by sketching. Children invent games all the time in paper and it would be beneficial to support such a creative activity. Seymour Papert (1999), inventor of the Logo programming language for kids, wrote, "I have found that when [children] get the support and have access to suitable software systems, children's enthusiasm for playing games easily gives rise to an enthusiasm for making them, and this in turn leads to more sophisticated thinking about all aspects of games." Ultimately, computers empower children to learn by themselves by facilitating the construction, exploration, and evaluation of their ideas (Papert, 1980). An insight into how this might be done is Gamut, a programming-by-demonstration tool that can be used to create games, simulations, and other interactive software (McDaniel and Myers, 1999).

Collaboration issues

"Lately, play has been rediscovered as a learning instrument. But now it threatens to turn into yet another form of child manipulation, of teaching the child "cognitive" information, while neglecting the socializing and experiential aspects of play." (Arnold, 1972, p. 5) Another interesting extension to Electronic Placemat is support for rich collaboration. One example is recognition of handwriting to differentiate between two people that are playing against each other. Another is networking support for many tablets to share one "placemat" workspace. Some research on sharing workspaces includes LiveNotes (Kam, Tarshish, Glaser, Iles, and Canny, 2002), which allow a group to share notes, and MediaBoard (Tung, 1997), a shared drawing tool.

Detailed Timeline

Documentation will be kept throughout the year to facilitate writing the paper.

November	Implement tic-tac-toe and missing letters. Start work on basic support for sketch recognition.
	Implement hangman and dots-and-boxes.
December	Finish sketch recognition.
	Winter break.
January	Create plan for user tests. Gather and schedule users for user tests.
	Start conducting user tests.
February	Finish conducting user tests and record findings.
	Start work on unified interface. Work on issues found in user tests.
March	Continue work on unified interface and issues found in user tests.
April	Finish work on the unified interface and issues found in user tests.
	Conduct user tests on unified interface.
May	Work on paper. Extra time left here just in case some of the intended tasks above are pushed back.
June	Extra time left here just in case some of the intended tasks above are pushed back.