Tools for Middle School Students to Create Vignettes

Naomi Caldwell
nhcaldwe@uiuc.edu

Matthew Miller
mjmille2@uiuc.edu

Jeffrey Naisbitt
naisbitt@uiuc.edu

ABSTRACT

Educators need a way to encourage youth in the school systems to discuss problems and issues relevant to their lives. Vignettes are short stories that are designed to model, in a less complex manner, real life issues. The purpose of a vignette is to encourage discussion. Currently, teachers and students have to write up vignettes and send them off to professionals to animate them on the computer. This process can be time consuming and costly.

We have developed a tool which will allow the user to quickly and easily create vignettes through an interface which should be fairly intuitive to teenagers, who are accustomed to chat applications such as AOL's Instant Messenger. The system we created allows users to create and run their vignettes. The system design allows easy deployment in schools at no cost.

User tests indicate the interface could be improved, but overall, it is rather intuitive and easy to use. The chat interface we use for dialog input is particularly beneficial. We conclude that, with a few modifications, the system would be beneficial for use in middle school classrooms.

INTRODUCTION

Educators need a non-threatening mechanism to encourage youth in the school systems, particularly those in junior high, to discuss problems and issues relevant to their lives so that the situations can be dealt with and the teenagers can be provided with the necessary support. Vignettes are short stories without endings that are designed to model, in a less complex manner, real-life issues such as racism, culture differences, prejudice, and isolation. The purpose of a vignette is to encourage discussion. A good vignette is fairly simple, sets up a situation in which there is no "right" answer, and is flexible enough that children from different backgrounds can relate to the story.

We have created an application which allows teachers to set up vignettes similar to what the youth would face in person. Then, through an interface similar to instant messenger, the youth can finish out the vignette as they would deal with it. This is important because currently teachers have to write up vignettes and send them off to professionals to set them up on the computer, and this process can take months, as well as being costly. Our application allows the educators to completely create these vignettes on their own quickly and easily. It also allows the youth to make up their own vignettes as a means of expressing themselves, improving writing skills, or just for fun. In addition, our application utilizes an instant messaging type of interface, which teenagers are accustomed to using.

CONTRIBUTIONS

Our work has four major contributions. First, it allows users to create vignettes without requiring animation design expertise. This allows users to create their vignettes without any collaboration, if desired. This is important for a couple of purposes. The time needed to create vignettes is drastically reduced because a person can operate on their own schedule rather than working with a web designer. Also, we feel this is a better environment for users to comfortably produce vignettes since they may feel awkward sharing a personal story with a designer. Once the vignettes are produced, the creator can remain completely anonymous. As a result of the faster turnaround time and increased privacy, more vignettes can be created, providing a larger set for students to analyze and discuss.

Our second contribution is that the interface is designed to be intuitive to the middle school age group (ages 12-15). The bulk of the work on a vignette will probably consist of inputting dialog to the scenes. Our interface uses a chat interface to allow the user to input the dialog. As discussed later, chat and text messaging are one of the most used computer applications among teenagers. Therefore, the chat interface should feel less like work to the student and provide an environment to which they are accustomed.

The third contribution of our work is that it encourages students to use technology. Funding for computers in K-12 education has increased with the rise of the Internet, but this is of little use if students do not have productive applications to work with. From this perspective, our tool does not improve skills or explicitly teach like many educational tools (e.g., learning typing or math), but encourages students to create a final product. The vignette produced is useful because educators and students can analyze it. Encouraging production via computers is a skill which will benefit students later in life.

Finally, our system is designed in a way that is readily deployable in public schools. Specifically, this refers to two criteria. First, the system is relatively easy to set up. It is platform independent, so no specific operating system is required. The setup could easily be placed into a small script which basically just sets the appropriate environment vari-

ables. Second, the entire run-time environment for our tool can be obtained at no cost. This allows schools to add our tool to as many systems as desired with no site license issues. Also, the tools are readily adaptable to a web interface so the tools could be used over the Internet if the appropriate access controls and file system were set up on a server.

RELATED WORK

There are three general research directions which are most relevant to our project. First, we describe research with interfaces designed for children. Next, we explore projects which investigate applications that enable the early design of user interfaces. This area is of interest because users are typically given the ability to specify temporal ordering among various states or scenes. Finally, we look at research that has been done in instant messaging usage and interfaces. We provide brief overviews of current and past research in these areas and describe their relevance to our work.

Designing for Children

Interestingly, in searching for literature related to our project, we found that multiple projects focused on interface design and environments for young children (less than ten years old), but virtually none for teenagers. As mentioned our project is targeted at the teenage age group by designing an instant messaging interface to allow dialog creation. We note that such a task, while perhaps relevant for some younger children, is designed primarily for teenagers who would be more familiar with the interface. Therefore, we consider work in this section relevant to our work because it is focused on younger children, rather than adults.

One of the earliest large-scale projects designed specifically for kids was KidsRoom [3] developed at MIT. The primary aim of the project was to create an immersive environment for children to interact with physical objects rather than virtual space. The room was designed to be group-oriented, rather than focused on one or two individuals. They claim that as group size increases, the children tend to become more comfortable and have more fun in the environment. Another challenge was creating an environment where all the narrative, help, and directions is done within the context of the story rather than explicitly. Robustness is another issue since the system should gracefully handle children who do not follow instructions (i.e., there should not be one rigid script which users must follow to enjoy the story). KidsRoom was specifically targeted at children ages six to ten.

A large part of the project was based on movement recognition in a noisy environment with multiple sources. This was handled by using a loose interpretation of the intended motion (e.g., nearly any arm movement is interpreted as rowing) and allowing the story to progress even if the children repeatedly ignore instructions. The system would attempt to instruct users to follow the intended course by having characters issue suggestions in a colloquial manner and hints from multiple omniscient narrators. KidsRoom demonstrated the story theme keeps children using the room rather than trying to test its limits. Particularly beneficial is allowing children to perceive that their actions are influencing the story. Participants should understand a clear cause and effect relationship, which is especially difficult in a group setting. To avoid making the room seem mechanical in narration, two narrators were used. This is beneficial when an instruction must be issued more than once because it avoids sounding mechanical and repetitive.

The KidsRoom project is relevant to ours because it helps define how younger children may behave different than adults when using an interface. For example, children are more likely to test the limits of the software or intentionally not follow directions. Therefore, the design should be robust to this behavior. It also motivates the benefits which children experience in group-oriented activities. Obviously, the goal of designing an immersive environment in which children are a part of the story is different than our goal of allowing children to create their own story.

An extensive, on-going project called KidPad [4, 2, 8] is a collaboration between HCI researchers, educators and students to support an interface for children to create stories. The project is focused on ages six to eleven. The initial goal of the project was to encourage visual and verbal literacy among kids via open-ended drawing and telling stories. One of the initial observations was that students loved having a zooming feature to tell stories. This allowed them to provide more detail in certain parts of the story. They also enjoyed having a text box associated with objects to give additional information when selected. *Local tools* were designed to allow the children to do common activities like draw and erase. These tools were placed directly on the screen, rather than having a separate palette as is typically seen in drawing programs. The zooming allows a non-linear, complex story whereby different objects of a larger picture can essentially hold a story of their own. The locations of scenes are linked to larger drawings.

Another issue in KidPad (actually a larger project called KidStory [2]) is the use of collaboration. The goal is to encourage, but not force, collaboration by providing a shared interface. This is done via Single Display Groupware (SDG) technology. The idea was to allow children to experience a benefit by collaboration that could not be achieved as an individual (e.g., two users could combine crayon colors to create new colors which would not be available to a single user alone). Many of the issues involved in this addition are classical mutual exclusion problems (e.g., what happens if two users try to select a different area to be zoomed at the same time).

The KidPad project is similar to ours because an interface is being designed to allow storytelling for children. While KidPad is focused around non-linear, spatial linkage, our design is centered around a linear, temporal linkage between

different parts of the story. Also, since our design is for older children, such "fun" novelties like zooming may become annoying to users not as fascinated with such navigation. However, there are many ways that research from KidPad can be applied to our tool. The primary selection tool for characters and backgrounds is similar to local tools because selection is done graphically within the primary window. Also, collaboration fits rather naturally into our project. Instant messaging is inherently a group-based activity. The idea of having multiple characters lends itself well to various students playing different roles to create a story. In the future, we would like to allow the instant messaging interface to work over a network so various characters could enter their dialog from remote locations.

The use of vignettes [9] in education is a driving force of this project. These are small stories which convey a situation to students and allow them to respond to the problem and resolution from the story. Vignettes are useful to education researchers for learning about students' reactions. Our work is complimentary to the research because we intend to allow students to easily create their own vignettes. This will facilitate more stories being available for student response.

Tools for Early Interface Design

Research described in this area is focused on allowing users to create early interfaces quickly, easily, and efficiently. The aspect of this research that is most relevant to our project is the specification of the linkage between objects. In our project, we want to allow students to specify linkage between scenes for a temporal ordering. Our project is similar to early prototyping because we want the user to create something simple to describe a story. In the future, more complex actions are desired, such as animation and making the scenes more life-like. Similarly, tools in this section started with a simple goal and added useful features, such as automatic translation of a sketched design to Visual Basic.

The SILK project [10, 11] was designed under the observation that early design tools are good at specifying the layout, but bad at showing the behavior of interfaces. Therefore, the user needs a way to storyboard their interface to show state transitions. Users can then carry out simple actions on the sketched interface which would not be as realistic if paper and pencil were used. Arrows are used to link a certain object (e.g., button) with a different state. The sketching interface is used to give the flexibility and efficiency of the pencil and paper approach. SILK was also extended to output Visual Basic or Lisp code for the sketched interface. This allows the user interface to then be demonstrated with a real computer interface.

The major difference between this work and ours is that it is entirely centered around sketching, whereas our interface is drag-and-drop with typing. This difference exists with our work and all the tools mentioned in this section (i.e., they are all designed for sketching). Eventually, it would be beneficial to incorporate some simple sketching into our project to allow users to dynamically create new characters and backgrounds. SILK takes a fine-grained approach to transition specification by allowing the starting point to be specified as an object within a scene. Our approach is more coarse-grained with transitions only specified between scenes. This granularity, combined with the drag-and-drop nature of our interface, makes sequential ordering by thumbnails or textual descriptions of the scenes more natural. Also, in SILK, transitions are as important as the static layout. In our project, transitions are secondary to the scene creation and dialog.

The next relevant work is Anecdote [6] which allows designers to do early sketches of multimedia prototypes. One of the key innovations of this project, is that existing multimedia content can be imported into the sketches. This content is interchangeable with textual annotations or sketches if the content is not available. The primary advantage of this approach is it better allows the designer to consider multimedia attributes, such as bit depth and sampling rate. For linkage, they have a separate view (Link View) which shows all scenes from the storyboard and allows connections via arrows.

This work is similar to ours because we allow media to be imported into our design. However, our content is static images. Also, we want to have one primary view, rather than switching between different modes as Anecdote does.

The DEMAIS tool [1] is also designed for multimedia designers to communicate behavior within the interface. One major contribution of this work is an elaborate quantitative analysis of DEMAIS with paper and pencil and a commercial product. This tool also takes a fine-grained approach to specifying temporal behavior because within a scene, various events may be designed to occur at certain times in response to events in the main window. The sketch is translated into a prototype which can be tested as an interface. The experimentation was done by allowing a client to work with an expert designer to achieve a task. Each of the various tools is then rated and ranked according to various criteria.

One weakness mentioned with DEMAIS, that is relevant to our work, is the users found the ability to only work on one storyboard at a time a disadvantage. We too allow only one storyboard to be worked on at a time. However, we feel this is not a major issue in our project since there are not complex relationships between separate scenes. We anticipate the focus to be creating dialog for the current scene rather than concern for its relationship with other scenes. Our tool is more similar to the interface of a slide show editor, such as Powerpoint. While the relationship between the various slides is important and a temporal ordering is specified, the interface is centered around the content of one slide. One complementary aspect of DEMAIS is the extensive testing framework which was helpful in defining criteria by which users can rate our interface and determining how to quantify

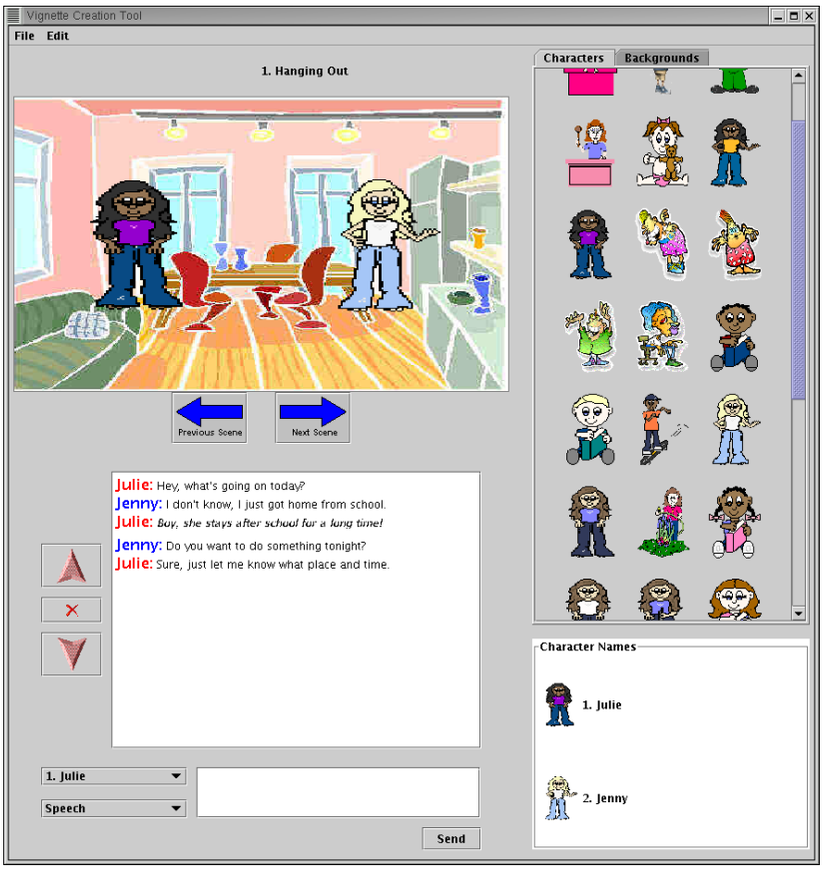

Figure 1: Vignette Creator

our design.

Instant Messaging Design and Use

Research in this section is focused on instant messaging. One major issue is how this can be used, particularly in a work environment. Another issue is how better interfaces can be designed for real-time messaging. This research is relevant to our work for a couple of reasons. First, it motivates our work by showing why teenagers prefer this type of messaging and are comfortable with it. Secondly, it proposes solutions to common instant messaging problems which we could integrate to increase the usability of our tool. However, most of the problems with instant messaging are caused by users' inability to know what the other user is doing or when they are going to type. Most of this research would be more relevant if our tool used a network connection for communication, rather than a shared local machine. The ability to communicate over a network is an area of future work.

In [5], the authors study reasons teenagers use text messaging so prevalently compared to other means of communication, such as using the telephone. Text messaging is slightly different than instant messaging because short messages are sent (usually using a cell phone) which are then answered asynchronously. However, many of the advantages and disadvantages teenagers perceive with text messaging are relevant to instant messaging. The basic motivation teenagers have for text messaging is it is quick, cheap and easy to use. The study collects the habits of a group of teenagers by having them log information every time they use this technology. One observation is girls are more likely than boys to use text messaging. One common use teenagers find is adjusting arrangements and coordinating activity among friends. Less common is chat and gossip, with communicating with their family being the least common use.

Some advantages of text messaging include the interface being familiar and being able to reduce the conversation overhead of communication (e.g., greetings and similar generally accepted procedures). Another consideration is that teenagers are usually tight on money and messaging is a cheaper means to communicate than calling a person. Some problems that were found with text messaging include the language evolving so fast that acronyms and phrases are unknown or ambiguous. Because voice and facial expressions are not involved, sometimes it is more difficult to determine a sender's intent from the content. Also, it is relatively easy to send a message to the wrong person. This work is complementary

to ours because it enumerates reasons teenagers would find an interface such as ours intuitive and usable.

The focus of [7] is to describe problems that arose when attempting to introduce instant messaging to the workplace. The user group is adults, which tend to be less familiar with the instant messaging interface than teenagers. The interface was designed primarily for group messaging. Therefore, feature such as allowing users to view the last few hours of conversation while they were offline are important. They introduced an initial version of their interface and, when over 90% of the users stopped using it, modified it according to user suggestions and reintroduced the new version. Some of the modifications included standard bug fixes, more privacy options, and more team-based training. The new version showed better, but not great, adoption rates. One of the major reasons workers were hesitant to adopt the interface was the tool was seen as more social and not as helpful as a work program. Another limiting factor was the ability of the tool to reach and sustain a "critical mass". This means if there are not enough relevant users working with the tool, usage will decline. The results of this paper should not affect the adoption of our work because the messaging interface is not designed for long-term projects, but rather short usage times. It is expected most vignettes will be developed in one sitting. Also, there is no concept of critical mass since our goal is not to have a large number of people communicating. Instead, we use the communication interface for the task of creating dialog for a story.

Another group of research focuses on problems that arise in instant messaging and interface designs to alleviate them [12, 13]. One major problem is that of users taking turns. Since the interface does not allow the user to know when the other is typing, sometimes the conversation will get confusing. A simple fix to this problem is to have an indicator on-screen when the other user is typing. Another problem is users typically maintain several conversations or do other tasks while conversing. Therefore, the pace of the conversation may be different for the users and one may not be able to tell whether the other is offline for a long time or soon to answer. The general solution for this problem is some kind of status indicator which ages when users are idle. This work relates to our project being extended to a network setting. If users cannot see each other, they need a better understanding of what part of the story the remote user is working on. Also, if they are engaged in a dialog, there would probably need to be some control over the flow of the conversation to make sure users do not frequently overlap their dialog.

METHOD

Our tool, the Vignette Creator, allows the user to quickly and easily create vignettes through an interface which should be fairly intuitive to today's teenager, and even their teachers, who are accustomed to applications such as Instant Messenger. The layout of the system, as shown in Figure 1, is fairly simple. The view in the figure is what is shown when a scene is being created or edited. The layout essentially consists of four boxes, a menu bar, and two large arrow buttons to move forward or backward between scenes. The four boxes, the purpose of each which will be described in detail, are the picture box, the element box, the instant messaging interface, and the active character box. There is also a scene editor which displays in order the titles of all the scenes in the current vignette as shown in Figure 2. Through the scene editor, the user can title scenes, insert and delete scenes, and reorder the scenes in the vignette. The vignette player is shown in Figure 3.

Picture Box

The picture that goes with the scene is in the upper left corner of the screen with the menu bar above it. The picture box contains whatever elements have been placed in it by dragging and dropping from the element box. These elements can be moved around within the box by clicking and dragging with the mouse, and they can be removed from the scene by dragging them outside of the picture box. The main items in the picture box are the character elements chosen for the scene. Only the characters in the picture box are available in the active character box to be given lines to speak. There is also a background and whatever objects the user chooses to include.

Element Box

The elements that can be used in the scene are selected from the element box in the upper right. There are three types of elements: characters, backgrounds, and objects (objects are not currently implemented). Using the tabs at the top of the box, one can switch between the types of elements, with the box containing small pictures of each of the possible elements of the currently chosen type. The characters are the "actors" in the scene, including people of a variety of ages, genders, body types, and ethnicities. The backgrounds are the possible settings for the scene, including a school room, living room, playground, and school hallway. Each scene can have only one background. The objects are items such as schoolbooks, backpacks, and chairs which can be used as props in the scene. All of these elements can be added to the scene by using the mouse to click on the desired element and drag it over to the picture box.

Instant Messaging Interface

The "script" for the scene is written in the instant messaging interface in the lower left corner of the screen. Using drop-down menus within the box, the user can choose which character to create a line for and whether it will be thought or speech. Buttons on the left-hand side of the textbox containing the script allow the user to move between existing lines to edit, delete, or reorder them. When the vignette is played, the text in this box is run through a text-to-audio program to produce the audio for the scene.

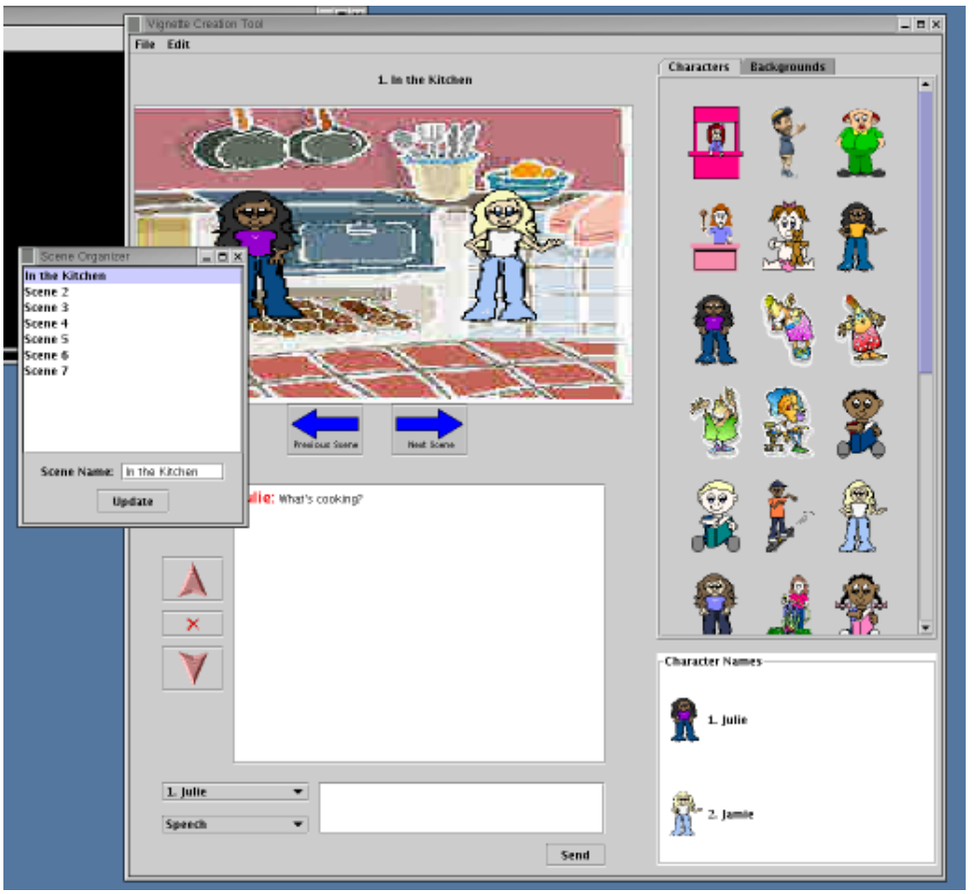

Figure 2: Creator and Scene Organizer

Figure 3: Vignette Player

Active Character Box

The active character box in the lower right corner of the screen contains small pictures of each of the characters in the scene along with their names. When a character is added to the scene (by dragging and dropping it from the element box to the picture box), a small picture of the character and it's default name appear in the active character box. The user can change the character's name by double-clicking on it ei-

ther in the active character box or the picture box. These are the only characters available for use in writing the script to accompany the scene.

Arrows and Menu Bar

The last two parts of the layout are the arrows and the menu bar. The arrows allow the user to move between scenes while remaining in the editing mode. The forward arrow switches to editing the next screen, and the backward arrow switches to editing the previous screen. The menu bar has the standard options under File – open, save, save as, close, exit. The options under Edit are delete scene, insert scene, delete background, and scene organizer. The first three are self-explanatory. The scene organizer brings up a separate window containing a list of all the titles of the scenes in the vignette in order. It allows the user to re-title, insert and delete, and reorder the existing scenes in the vignette.

Together, these four boxes, the arrows, and the menu bar comprise the interactive window shown while the user is creating or editing a scene. This is the primary view that will be used, but there is also the scene organizer which can be brought up in a separate window to display a list of all of the scenes in the vignette in order. As we had hoped and the survey results appear to reflect, this is a tool that educators can use with their junior high students to encourage discussion of issues. Having an interface that they are familiar with will help them learn how to use the tool faster. Having choices for backgrounds and characters will allow the users to create vignettes that model the real-life situations.

SPECIAL HARDWARE/SOFTWARE

As stated earlier, one of our contributions is that we have developed a platform-independent system, allowing portability and free availability. To accomplish this goal, we implemented everything using Sun's Java programming language. We used Java Swing to implement the graphical user interface. Originally, we considered various speech API's and speech engines, but due to their platform dependence and licensing issues, we chose to use FreeTTS. FreeTTS is an open source (SourceForge) text-to-speech engine, implemented entirely in Java. It implements most of the text-to-speech portions of the Java Speech API (JSAPI). Currently, it does not support the speech recognition functionality though. Using FreeTTS also provides access to a large database of information and useful utilities including more realistic voices. We feel this will provide the greatest degree of future compatibility as well. Finally, since it is also written entirely in Java, it is also platform-independent.

USER TESTS AND RESULTS

We performed some preliminary user testing using six college-age subjects. Although we would ultimately like to have middle school students test the system, we feel these subjects at least provide a general overview of the major features and weaknesses. For the tests, we had the users create and edit a vignette composed of three scenes. The users were asked to test various aspects of system functionality, including scene creation, scene reorganization, and character placement and editing. After completing the tasks, the users filled out a survey consisting of six quantitative questions, requiring the user to simply circle their response, and six free-response questions (Figure 4). We designed the survey to answer three primary concerns:

	Quantitative Question					
	1	*2*	*3*	*4*	*5*	*6*
User 1	A	A	A	SA	SA	SA
User 2	SA	SA	A	A	SA	SA
User 3	A	A	A	SA	A	SA
User 4	A	A	SA	SA	A	SA
User 5	A	D	D	SA	A	SA

Table 1: User Survey Results

1. Is the tool intuitive and easy to use?
2. Is the interface beneficial for middle school aged adolescents?
3. Is the chat interface beneficial?

The quantitative results are shown in Table 1.

The results from the user study helped us determine parts of our tool that were good and areas that could be improved. First, various aspects of the interface turned out to be somewhat counter-intuitive. Specifically, the mechanism for naming the characters and scenes was difficult. Also, many users would prefer alternatives to the drag and drop interface, such as simply double-clicking. Overall though, the users did not have substantial difficulty in performing and completing the tasks. Therefore, with some minor changes, this tool should be easy to use.

Second, most of our users felt that this interface would be beneficial for middle school students. However, as stated earlier, we would prefer to actually have the students test it themselves. Some other aspects of the interface, although familiar to college-age users, may be unfamiliar, and therefore non-intuitive, to middle school students.

Third, all the users found the chat interface to be a very effective and useful tool. It is a familiar, simple mechanism for inputting dialog. Even for people who are not familiar with "chatting" or instant messaging (such as User 6), the chat interface was fairly intuitive and easy to learn. The primary complaint was with the mechanism for switching speakers.

LESSONS

Performing user studies was very informative, and we learned quite a bit about our application in particular and user interfaces in general.

Quantitative Questions

1. The tasks would be easy for students (Ages 12-15)
2. The overall interface is intuitive
3. The interface would be useful for students (Ages 12-15)
4. The chat interface is an effective dialog input mechanism
5. The tool was easy to use in completing the tasks
6. The chat interface is useful in the vignette creation

Free Response Questions

- How could the system better designed for students (Ages 12-15)?
- How could the chat interface be improved?
- How could other parts of the tool be improved?
- What are better alternatives to the chat interface for dialog input?
- What are the strengths of the tool?
- Would you use this basic interface for the creating vignettes?

Figure 4: User Survey Questions

- User testing is extremely important. What seems intuitive to the designer and programmer may not be intuitive to the user.
- Common tasks should be as direct as possible. In our program, double-clicking on characters to change the name was the most difficult task for the users.
- Always keep in mind the complexity of performing common tasks. Do not simplify other tasks if it would require making common tasks more complex and difficult.
- Provide cues as to what actions are possible for objects. Users had to learn what properties could be modified for the characters by trial-and-error.
- Do not assume that users will want certain features. Create a prototype with the necessary functions and let user studies guide you in what features to add.
- Not everyone likes drag and drop. Some of the users wanted alternate methods of object placement. Again, it is important to determine what the users want and provide that for them.
- The chat interface still needs work. If users are typing a lengthy dialog, switching the active speaker can be tedious.

FUTURE WORK

Given the time ($\sim$ 3 months) and resource constraints (no commercial software, other classes and projects) of the project, there was only so much that could be accomplished. While we are pleased with the progress that was made, we feel the current state of the system is more of a milestone than a nearly completed project. Our work provides a solid foundation for future work on which we elaborate below.

User Testing The most obvious thing we would like to do is actually test the vignette tools on middle school students. Due to the time constraints of developing the code from scratch and the red-tape required to do this type of testing, we were only able to test the interface on our peers.

Customized Characters and Backgrounds Our current method of image selection is flexible because a user just has to add a GIF or JPEG file to the specified directory to include it. However, we would like the user to be able to create their own characters and backgrounds. A major enhancement to the tool would be to incorporate a sketching interface to allow the user to "draw" new images. At the very least, users should be given more flexibility to specify attributes of the images, such as skin and hair color.

Add Animation Currently, our tool only allows the placement of static characters. The first animation needed is lip movement for the character that is currently speaking in a scene. A straw man approach to this would be to create animated GIFs for each of the character images, though this would make it more difficult to add new character images. We would also like to give the user the ability to move the characters' arms and legs during a scene.

Make the Scenes Customizable Currently, due to perceived constraints and lack of documentation on Java's Drag-and-Drop features, we only allow the characters to be placed in one of two static positions in a scene. We would like to give the user more control over exactly where the image is placed in the scene as well as the depth. Similarly, we would like to be able to add more than two characters to a scene, which is the current constraint. Having multiple characters in a scene could bring more challenges in the interface design.

More TTS Options The TTS package being used provides many more options than we are currently using. The user should be able to select voices for characters and control the characteristics (e.g., pitch) of the voice. The TTS package we use also supports a standard format for voices, so the repository of available voices should be increased in the tool. Ideally, the user would be able to record their own voice and use it for a character's voice.

Add Networking Capabilities As mentioned, we would like to run the tools as web applets, so they could be used without installation on a local machine. This would also allow

central repositories of vignettes to be collected. Since the interface includes a chat interface, it would be natural to extend it so that users can remotely create a vignette over a network. This would include addressing the mutual exclusion and distributed system problems which arise in such an environment.

Anonymization and Filtering One common use of the tool may be that a small group of students develops some vignettes and then would like to release the vignettes publicly for others to view. Rather than having the user manually go through and change all the names and images to protect their privacy, a tool could be developed to automatically anonymize vignettes. Also, students may be prone to place explicit words and phrases in their vignettes for amusement. A filtering tool could be added to detect and change the dialog created by this misbehavior. Alternatively, the tool could be designed to automatically assign a "rating" to created vignettes to allow restrictions on who may view vignettes according to their content.

CONCLUSION

Our vignette tools provide a simple mechanism for allowing students to explore, in a non-threatening environment, problems and issues that are relevant to their lives. The vignette creator allows teachers or students to easily create a sequence of life events. This provides an opportunity for the youths to deal with situations, such as racism and prejudice, which may be difficult or uncomfortable in real life.

As the user tests show, the overall interface and system are fairly intuitive and easy to use. We have provided a solid foundation for continuing developments and improvements in the future.

Finally, our vignette tools should be very useful in educational environments for several reasons. The vignette creator uses a standard chat interface for the dialog input. This should be familiar and easy for students and teachers alike. The vignette player is very simple to use and provides an enjoyable, story-like, experience. Also, using Java and FreeTTS, our whole system can be distributed and used freely, without licensing issues and fees.

ACKNOWLEDGMENTS

We would like to thank Brian Bailey for providing us with the initial idea and suggestions throughout the semester. We are also grateful to all the users who agreed to participate in our testing.

REFERENCES

1. B. P. Bailey and J. A. Konstan. Are Informal Tools Better? Comparing DEMAIS, Pencil and Paper, and Authorware for Early Multimedia Design. In *ACM SIGCHI*, April 2003.
2. S. Benford, B. B. Bederson, K.-P. Åkesson, V. Bayon, A. Druin, P. Hansson, J. P. Hourcade, R. Ingram, H. Neale, C. O'Malley, K. T. Simsarian, D. Stanton, Y. Sundblad, and G. Taxén. Designing Storytelling Technologies to Encourage Collaboration Between Young Children. In *ACM SIGCHI*, 2000.
3. A. F. Bobick, S. S. Intille, J. W. Davis, F. Baird, C. S. Pinhanez, L. W. Campbell, Y. A. Ivanov, A. Schütte, and A. Wilson. The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment. *Prescence: Teleoperators and Virtual Envirnoments*, 8(4):367–391, August 1999.
4. A. Druin, J. Steward, D. Proft, B. Bederson, and J. Hollan. KidPad: A Design Collaboration Between Children, Technologists, and Educators. In *ACM SIGCHI*, 1997.
5. R. E. Grinter and M. A. Elridge. y do tngrs luv 2 txt msg? In *ECSCW 01*, pages 219–238, September 2001.
6. K. Harada, E. Tanaka, R. Ogawa, and Y. Hara. Anecdote: A Multimedia Storyboarding System with Seamless Authoring Support. In *ACM International Multimedia Conference*, pages 341–351, November 1996.
7. J. D. Herbsleb, D. G. Boyer, M. Handel, and T. A. Finholt. Introducing Instant Messaging and Chat in the Workplace. In *ACM SIGCHI*, April 2002.
8. J. P. Hourcade, B. B. Bederson, A. Druid, and G. Taxén. KidPad: a Collaborative Storytelling Tool for Children. In *Extended Abstracts of ACM SIGCHI*, 2002.
9. Intercultural & Cross Cultural Teaching Portal. http://icctp.ed.uiuc.edu/Vignettes/Index.html.
10. J. A. Landay and B. A. Myers. Sketching Storyboards to Illustrate Interface Behaviors. In *ACM SIGCHI Conference Companion*, April 1996.
11. J. A. Landay and B. A. Myers. Sketching Interfaces: Towards More Human Design. *IEEE Computer*, pages 56–64, March 2001.
12. A. Voida, W. C. Newstetter, and E. D. Mynatt. When Conventions Collide: The Tensions of Instant Messaging. In *ACM SIGCHI*, April 2002.
13. D. Vronay, M. Smith, and S. Drucker. Alternative Interfaces for Chat. In *ACM UIST*, 1999.