Developing Software for Cognitively Disabled Children: Teaching Time Telling Skills

Ryan Tierney

Follow this and additional works at: https://mosaic.messiah.edu/honors

Part of the Mathematics Commons, and the Special Education and Teaching Commons

Permalink URL: https://mosaic.messiah.edu/honors/246
Developing Software for Cognitively Disabled Children: 
Teaching Time Telling Skills

Ryan Tierney
Senior Honors Project
Advised by Dr. Gene Chase
2003-2004 Academic Year
Abstract

Schreiber Pediatric Rehab Center of Lancaster County works with children with cognitive disabilities, teaching them skills and giving them physical therapy that will assist them in living independently. The Center expressed an interest in obtaining computer software to help teach time telling skills to the children. I therefore decided to develop this application for my Senior Honors Project. There are a number of recommendations when it comes to teaching time telling, and I took these into consideration when designing my software. I used Macromedia Flash MX 2004 / ActionScript as my programming tool and developed a seven lesson program to aid Schreiber in teaching the students to tell time.

The Problem

Treatment of individuals with cognitive disabilities has changed drastically over the years. In the early years of our country, they were considered “idiots” and “feebleminded” and were forced to live in almshouses or hospitals. In the 1950s, institutions became the solution for dealing with this “problem.” It wasn’t until the 1980s that people finally began to treat individuals with mental disabilities with love and respect. Special sheltered workshops and group homes were founded, and public schools began implementing special education programs (Columbia Encyclopedia). The emphasis shifted from trying to hide these individuals from the rest of the public to trying to integrate them with the rest of society. An important part of this process is teaching them skills that will help them lead independent lives. One place where these skills are taught is Schreiber Pediatric Rehab Center of Lancaster County. The Center works with young
children who have been diagnosed with cognitive disabilities, focusing on both physical and mental skills that will aid them in living normal lives. Through discussion with Angie Rice, head of Schreiber’s Occupational Therapy department, we determined that the Center has a considerable need for software that can help teach the children time telling skills. The creation of that program is my Senior Honors Project for the Messiah College Computer Science Department.

Research

The first step in the creation of a program designed to teach something to children is to find out how children learn and what the best teaching techniques are for transferring knowledge to them. Children learn best when they are learning what they are curious about. If they are learning something simply because it intrigues them, they are much more likely to retain that information than if it is forced upon them (Holt 168). The amount of information that is transferred depends on the extent to which people learn with actual understanding rather than simply committing a set of facts or rules to memory. Understanding a topic means that one can see why something works the way it does, not just that it works a certain way. Therefore, it is important to ensure that children have enough time to adequately process information. Understanding does not always happen right away (National Research Council 55-8). John Holt puts it this way:

[Children] need time to build up in their minds, without hurry, without pressure, a sense of what words look like, before they start trying to memorize particular words. In the same way, they need time for “Messing About” with numbers and numerals, before they start—if they ever should start—trying to memorize
addition facts and multiplication tables. (156)

Memory is also improved when the learning experience is pleasurable. If children are participating in an engaging, vivid learning environment, they are much more likely to retain the information they are given, because they are actually interested in what is going on. Boring lectures cannot get children excited about learning the way more "alive" experiences can (Holt ix).

There are also many teaching techniques or ideas that are detrimental to a child's learning. Issler discusses three fallacies in teaching that can hinder the learning process. The first is the idea that 'To teach is to tell.' This view holds that a teacher's primary, and possibly only, responsibility is giving information to students. This fails to take into consideration all of the different options there are for conveying information and the many other important roles a teacher plays besides 'information presenter.' The second fallacy is that 'To tell is to know.' This idea falsely assumes that a student will understand everything a teacher says. This view does not take into account the varying learning abilities of students and the fact that it takes considerable time for some students to master a subject. Simply assuming that saying something one time is adequate enough to make the whole class understand it is extremely naïve. The third fallacy is 'To know is to do.' People who practice this falsely assume that, once students understand a concept, they will automatically incorporate it into their lives and start practicing it. This is not necessarily the case, however, because children will not always think to use their new knowledge (20). These fallacies must be considered when teaching so that one does not slip into them unawares.
After learning about effective teaching techniques and how to optimize children’s learning experiences, providing a base for the project, the next step was to research successes and failures others have had in attempts to teach time telling skills. One of the most important things that all children must know before even seeing a clock is how to count. If a child doesn’t know that 46 comes after 45, there is no way he will be able to decipher what the minute hand is pointing to. Therefore, being able to count to 60 by both 1’s and 5’s is essential. Without this skill, telling time with an analog clock is impossible (Krustchinsky 40). Once the child can count, the next step is to introduce the clock face by itself. It is important that he/she understand what the clock face signifies. They have to realize that time is cyclical, and that as time passes, it also progresses around the clock face. After this, the hour hand should be introduced by itself, so that there is not too much information at once. It should be explained that when the hand is pointing directly at 9, it is 9 o’clock, but when it is a little before the 9, it is almost 9 o’clock. Once they have a firm grasp on the concept of the hour, the minute hand can finally be introduced (Krech 16-7). The first thing the children should do is understand that there are 60 minutes in an hour. They can then count out 15, 30, 45, and 60 minutes to understand that, when the minute hand is pointing to the 6, it has been 30 minutes, or half an hour. At this point, it can also be explained that the position of the hour hand between the two hours corresponds to the position of the minute hand. For example, when the minute hand is at 6, half way around the circle, the hour hand is half way between the current hour and the next hour (Andrade 38). Once all of these concepts have been mastered, the children should have a good grasp on how the clock works and how to tell what time it is.
Once the research into classroom techniques for teaching time telling skills to children was completed, it was necessary to see if trying to teach skills using a computer program is even worthwhile. I discovered that a number of researchers feel that it is indeed valuable. Roschelle states that active engagement and frequent interaction and feedback increase the amount of knowledge and skills attained in the learning process (77). Computer programs are perfectly suited for this because of the nature of a computer. The child gets to control the program, so he/she is actively engaged in the learning process. Rather than just sitting there watching something, the child gets to click buttons, type in information, and just generally control what is going on. Computers also offer children immediate feedback. If a child chooses the wrong answer, he/she is informed of the error right away. In a traditional school classroom, children often have to wait a few days before they find out if an answer they chose in an assignment was right or wrong. With computers, they can find out immediately whether or not they have an adequate grasp of the information they are supposed to be learning. Research also shows that learning environments which include dynamic images and sound are especially helpful for students with learning disabilities. This is due to the fact that children with cognitive disabilities are often times easily distracted, and so the animation and sounds coming from the computer help to focus their attention. In addition to these benefits, computers give students a way to demonstrate their knowledge without having to write, which many students with learning disabilities don’t like to or can’t do (Hasselbring 108). It becomes obvious from all of this research that computer programs can definitely be an aid to teaching and that it is certainly worthwhile to use the technology. Roschelle confirms this, saying,
Research has demonstrated that technology can lead to profound changes in what children learn. By using the computers’ capacity for simulation, dynamically linked notations, and interactivity, ordinary students can achieve extraordinary command of sophisticated concepts. (6)

Solution

After all of the preliminary research, it was necessary to choose a language in which I would program my application. Upon visiting Schreiber and receiving a tour of the facilities, I discovered that both Macintosh and Windows PCs were in use. Also, the equipment is a few years old and not normally updated. This is due mainly to the fact that there is no real tech person employed at the center. Taking this into consideration, I determined that I would need to program in a language that was not operating system dependent and could be easily installed and used by any of the staff at the center. Also, due to the fact that the users of the program will be cognitively disabled children, I decided it would have to be a language that could create nicely animated, engaging scenes while allowing for easy user interaction. Taking all of these requirements into account, Macromedia Flash MX / ActionScript seemed like the ideal solution. Macromedia Flash is a powerful yet simple animation creator that produces web applets that display directly in a web browser. Because it is browser-based, any computer running any operating system can run a Flash file, as long as a web browser is installed and the Macromedia Flash plug-in is installed. It also allows for easy user interaction through ActionScript coding.
Macromedia Flash MX 2004 is fairly simple to use, even for a novice programmer. I spent about a month learning the intricacies of the program and how to tie ActionScript in with the graphics to create an interactive product. I relied on three books to get my Flash / ActionScript training:

*Macromedia Flash MX: A Beginner’s Guide* (Underdahl)
*Macromedia Flash MX Unleashed* (Pizzi)
*Macromedia Flash MX ActionScripting: Advanced Training from the Source* (Franklin)

As can be seen in the picture below, the interface consists of a stage where all of the graphics are placed, a timeline that allows modification of each individual frame and creation of motion between them, a library that contains generic graphics and other objects that the user can create and save, and a graphics toolbar consisting of a number of common tools including draw line, draw circle, etc.

*Macromedia Flash MX 2004 Interface*
The stage is where all of the graphics are placed for each frame. It is the white square in the center of the screen, and can be set to any size the user desires. This setup is convenient, because it allows a graphic to move into the scene from off-screen, simply by starting it off to the side of the stage and then moving it on. The timeline at the top is also a convenient feature of Flash.

**Timeline**

The timeline allows multiple layers to house different graphics, and each frame in the timeline represents what will display next. The movie is run by simply moving from one frame to the next, creating the animation that is seen on the screen. The close-up of the timeline (above) shows the layers and frame features.

The library that Flash contains is another powerful and extremely useful feature. It houses generic objects that the user can create and then put instances of in his/her movie. For example, in the picture above, the library contains a graphic object that I created. I can
then put instances of this object in as many frames as I want. The library allows for reuse of things the user creates, which can save a lot of time when programming and also allows significantly smaller file sizes.

The language that Macromedia Flash uses to create interactivity in movies is ActionScript. This language is similar to the JavaScript language in many ways. Flash contains an ActionScript panel for each object on each frame of a movie. This means that the coding for each object is kept separate from all other coding, allowing for easy updates and/or fixes to code without the confusion that is often caused in other languages by having all the code in one place. For example, the following graphic shows the ActionScript panel for the hands of a clock in a sample program I created. The code tells Flash how to display the hands so that they accurately reflect the system time of the computer that the Flash movie is being run on.

**Sample ActionScript Panel**
In short, Macromedia Flash / ActionScript allows for the creation of beautiful, interactive, browser-based programs that are fairly simple to design and implement. The ActionScript element gives me the control I need for this application, while the Flash environment provides me with the perfect way to create animations that will keep the children engaged in what they are doing.

After learning how to use Macromedia Flash and ActionScript, it was time to start planning what my program would do. This required doing significant storyboarding – drawing out each lesson frame by frame, indicating what would be displayed on the screen and what would be going on behind the scenes. This was when I had to put my research to good use. I tried to follow the guidelines for teaching time by first making sure the students understand how time works – that it is a 24 hour cycle called a “day,” that the day is broken into different pieces such as morning and afternoon, and that our measurement of time is based on the sun circling the earth. After this, I included a lesson on counting from 1 to 60, as all of my sources mentioned its importance. I then followed Krech’s advice and broke the clock down into three lessons: the first introducing the clock face and explaining how the numbers go around, the second demonstrating the hour hand, and the third finally including the minute hand (16-7). The concept of seconds is briefly touched upon, but I decided that including the second hand would be too much too fast, especially for children with cognitive disabilities.

The actual programming of the software consisted of trying to create the ideas outlined in the storyboard. Although Flash is a fairly intuitive application, the programming still presented some challenges and I had to stray slightly from my original plans. The main ideas were still covered, however. I attempted to include as much
animation and color as possible, based on information in my research that showed these things aid in keeping children engaged. MIDI music files play in the background as another attempt to keep the children interested in the program. I made the buttons that the children have to click rather large to take into consideration the fact that some of the children at Schreiber have difficulty with their motor skills. Bigger buttons provide a larger surface area that can be clicked, making it much easier for these children to interact with the program. There is a significant amount of text in the program so, per the advice of Angie Rice, I added voice audio files that read all of the text. This will allow students who have difficulty reading to use the program without requiring a staff person to sit with them the whole time.

**Strengths**

My application, *It's Time to Tell Time*, has a number of aspects that make it a good solution to the problem at Schreiber. It is programmed in Macromedia Flash, so it can be used on any computer with a browser and the Flash Player 7 plug-in. The Center doesn’t need to worry about an operating system requirement. Programming in Flash also means that there is a good amount of animation and lots of bright colors to keep the children engrossed in the lesson. Music files also serve this purpose. The interactive functions take into account users with decreased motor skills by using large, easy to click on buttons. Voice files read all of the text so that users with lower reading skills will still be able to use the program independently. Finally, the lessons take a research-based approach to teaching time telling skills, covering the basics and then slowly increasing
the difficulty of the subject matter until the skill of reading a clock with both hour and minute hands is attained.

**Weaknesses**

The voice audio files are low in quality. Because of time and budget constraints, I was working with a low quality microphone. Also, to keep file sizes low, Flash automatically compresses audio files, further decreasing their quality. The files are audible, but a more professional product would include much better voice-overs. The counting lesson only teaches how to count by 1s and 10s, not 5s. It is possible to tell time with just these skills (count to the nearest 10 and then use 1s the rest of the way), but it is easier when one knows how to count by 5s. Time constraints prohibited me from including this. Finally, again due to time constraints, there are not as many practice exercises as I would have liked to include. Practice is the key to learning a new skill, and if I had more time, I would have given the students many more opportunities to practice what they were learning and get feedback on their progress.

**Conclusion**

Designing a computer application from scratch is never easy. It requires a great deal of planning and preliminary work that one might not always think of. Creating a program for cognitively disabled children, especially one that is supposed to be teaching them life skills, is even more challenging, because it requires a whole set of guidelines and rules that most applications for other users do not. With a good base of research and a thorough examination of the needs of the program, however, the design of a quality
application does not have to be impossible. This year has been a very valuable experience in the software design process, and it has shown me how much work really goes into a project like this. Although my final application does have some weaknesses, I think it is a viable solution to the need at Schreiber Pediatric Rehab Center of Lancaster County.
Works Cited


