TJHSST Computer Systems Lab Senior Research Project Paper
Elementary Education in a Technology Age
2007-2008

Gregory Gates

3/12/08

Abstract

Technology becomes more advanced and more accessible with every passing day. Education should be utilizing this technology boom in teaching current students. However, this does not seem to be the case. The goal of this project is to try and implement computer programming, through Scratch, as a tool for educating students in math and science topics. Computer science education at a younger age becomes more and more essential as computers become more advanced and more accessible with each passing day.

1 Introduction

The main question that this research project aims to answer is, "How young is too young to start teaching children how to program?" The goal is to establish a computer science program at Cardinal Forest Elementary School through the use of the MIT developed program 'Scratch.' Hopefully by watching how the students use the program I will be able to come up with an answer to the question.

This paper will detail both how Mr. Allard and myself will teach the students, and how the students go about solving the problems that they are presented with. Mr. Allard and I will develop a curriculum based on

the Virginia Standards of Learning (SOL) and Program of Studies (POS) standards for the students. The children that participate in the program are in first through sixth grade, with each grade having a similar curriculum scaled to the necessary level. Not all of the elementary school students will be participating in this project however. Mr. Allard has selected a diverse group of students for this initiative that he think would do best in and benefit from a program such as this.

2 Background

The task of educating the younger generations about programming has been attempted before. The first attempt to create a kid-friendly programming language was Logo, made by Wally Feurzeig and Seymour Papert. This programming language mainly involved telling a turtle how to move around in order to make various pictures with the turtle's "pen." Since then, multiple programming environments and languages have come about to try and engage not only youth but more specifically girls in computer science and programming such as: Squeak, Alice, and Scratch.

Despite the bountiful number of tools that modern technology gives us for teaching students, little progress has been made in teaching computer science at the elementary school level. The necessary technology is present in the schools, but it is only being used to reinforce outdated teaching methods. Currently, computers are mainly being used as a medium to transfer information, much like a television. Computers have so much more potential than that. They should be used as a universal construction material, not as a TV screen. Programs like Scratch enable kids to create whatever they want to all by themselves. Children learn better by immersing themselves in whatever they're doing, rather than just listening to a teacher telling them what to do (Papert, 1993).

The goal for this project is to establish something akin to a Computer Clubhouse at Cardinal Forest Elementary School. The original Computer Clubhouse was started by the Massachusetts Institute of Technology in Boston in 1993 to "provide more young people with the opportunity to become digitally fluent." (Resnick, 2002) At these clubhouses, kids and older youth "become designers and creators with new digital technologies. Clubhouse members use leading-edge software to create their own artwork, animations, simulations, multimedia presentation, musical compositions, websites, and

robotic constructions." (Resnick, 2002) I want to start a computer science program at Cardinal Forest where students can think for themselves and create whatever they can imagine.

3 Development Sections

3.1 Timeline

This past fall, I contacted the principals of the middle and elementary schools in the West Springfield pyramid inquiring about the possibilty of starting a computer science program at their schools. Only one school replied (Cardinal Forest) and the principal reffered me to Mr. Allard. October and Novemeber were spent sorting out which programming language to use and how we were going to use it. After experimenting with 'Squeak' and 'Alice' we decided on 'Scratch.' It was decided that the Scratch Lunch Bunch as it has come to be known would meet during Lunch on every Thursday. I personally teach the students every other Thursday.

As for the progression of topics, Mr. Allard and I hope to let the kids start programming on their own by the beginning of March. Until then, he and I will be taking turns discussing and teaching the many different fucntions of Scratch, in order to prepare the students for the time when we do let them program on their own. I know that the kids are eagerly waiting for that time as much as I am, and the goal is to move through the fundamental computer science topics as quickly as possible so that we can reach that point.

A new type of programming environment or activity may be introduced around the end of the year. Right now, the plan is to teach the kids about a program called "Cricket," which is similar to the Lego Mindstorms program. This will introduce the students to the very important topic of artificial intelligence in a fun and exciting way.

3.2 Lessons

Each class lasts from 30-45 minutes, depending on the age of the students attending. Thus far, Mr. Allard or I instruct the students on what to do for a majority of the time, but this is not our end goal. As the year moves on, we will encourage students to explore the program independently and discover new methods and techniques on their own. We hope that this will not only

Figure 1: Some girls working on their projects

teach the students how to take some initiative, but we think that they will gain a better, more solid, understanding of the program through individual exploration. Each lesson begins with the students signing in to the lab and quietly sitting down at their computers. Once everyone is seated, we begin teaching. By the end of the year, we hope to simply give students a project idea or category (i.e. celebrations or sports) and have them make a project all on their own.

On February 28th, I taught the students how to create their own sprites and backgrounds. After learning these techniques, we let the kids begin work on their own projects. Many of the kids for their first projects are making "stories" or little "movies" as opposed to any interactive type of program (i.e. a game). This was to be expected, as telling a story is likely not as difficult as making anything interactive. A couple of the more advanced techniques still need to be taught (mainly broadcasting and variables) in order to help the students' programs progress even further. However, we will hold off on teaching about these commands for a little while, as letting the kids explore Scratch on their own is of greater importance right now.

3.3 Topics

The actual number of topics that can be covered will be dictated by how quickly the students move from one topic to the next. Thus far, the students have been exposed to the coordinate axes, degrees, if-then statements, and

basic loops. Multiplication was also brought up in the loops discussion since the two subjects go hand in hand. We did this by first having the students tell their sprite to move in a square by tellin it to move, then turn 90 degrees, move, turn 90 degrees, move, turn, move, turn. We told the students that this type of coding is inefficient and that there is a way to tell the sprite to do the same thing over and over again: loops. We instructed the kids to use the "repeat" loop that is found in Scratch. A square has four sides, so the sprite would have to do the move–¿turn sequence four times in a row. Four times two commands equals eight commands, which is what the students started out with.

There are some more advanced topics unique to Scratch (most specifically broadcasting and sprite interaction) that have yet to be touched upon. Currently the students can only create "fake" or "staged" sprite interaction by telling the sprites to "wait" a certain amount of time from one action to the next. Broadcasting allows the sprites to actually interact and enables the triggering of different events. Broadcasting plays a greater role in interactive programs that it does in the story-type programs.

I will also make sure that the students know how the different types of control blocks (i.e. "forever" and "if") can be used together to make different things happen. Lessons have been given on these specific types of blocks individually, but we've spent relatively little time trying to integrate the different commands into a single script. If we're lucky, the students will learn to do this on their own during the time we give them in the next couple weeks to work on their own projects.

3.4 Resources

The Cardinal Computer Lab features 30 student computers along with a teacher workstation connected to a Smartboard. While we wish the room was bigger so that more students could be accomodated at one time, these resources are sufficient for the project. Currently, only one parent volunteer has come to the meetings, but other parents have shown interest in getting involved with the program. Mr. Allard will be in attendence at every Scratch session, and I try my best to attend every other one.

Figure 2: The older, laptop-based computer lab

3.5 Restrictions on Study

The most pressing restriction for this project is time. Seeing as how I am still a student, it is difficult for me to be at Cardinal Forest during the school day. This limits possible meeting times (when they don't conflict with either Mr. Allard's schedule or my own) to before and after school. This of course carries with it the problem of transporting kids either to or from home, seeing as how elementary schools don't have late buses. Despite these time restrictions, I have full faith that a computer club will be possible at Cardinal Forest and that the research project will be successful. The other time related restriction is the fact that I will be graduating in June and moving on to college the following school year. My hope is that another student in the computer systems lab next year will be able to pick up my project from where I left off and continue working with the students. Hopefully I will be able to check back in with Cardinal Forest during my breaks to ensure that everything is still going smoothly.

3.6 Reporting Results

To put data into a chart or graph for this project could be difficult, unless something of an assessment is offered to the kids at one time or another. Mr. Allard and I are reluctant to give such an assessment because we're afraid it

could discourage some of the kids from participating in the program. He and I will come up with a way to document the progress of the students. The data generated from this experiment will most likely be fairly subjective, based on my experiences with the elementary school students, and this would be rather difficult to simply place into a graph. The students were recently given a survey asking about their general knowledge of computer programs and computer programmers. The survey had three questions: "What computer programs do you use¿' "What *is* a computer program¿' and "What are computer programmers like¿' There was an unforseen problem in handing out this survey: many of the students (more or less from all of the grades) had trouble with the vocabulary. They had never heard of any software being called a "computer program" before, and this caused the survey to take up much more time than it was meant to. I do not have the results from this survey yet.

4 Results and Discussion

As mentioned before, I hope that this research project will yield good results that would encourage the implementation of a simple computer science curriculum at the elementary school level. The earlier that kids can start to program and become interested in programming, the better. The computer has the potential to start a digital revolution in learning, not only in math and science but in English, and social studies as well. By the end of this initiative's first year, I hope to show how the students have made progress in multiple school subjects as laid out by the SOL's and that technology gives us better ways to teach the students of today.

References

[1] Burd, Leo and Robbin Chapman. Beyond Access: A Comparison of Community Technology In N.p.: n.p., 2002.

[2] Papert, Seymour. The Children's Machine: Rethinking School in the Age of the Computer. New York: Basic Books, 1993.

[3] - - -. Mindstorms: Children, Computers, and Powerful Ideas. New York, Basic Book, 1980.

[4] Resnick, Mitchel. "Rethinking Learning in the Digital Age."
The Global Information Technology Report: Readiness for the Net-worked World.
By G. Kirkman. Oxford, UK: Oxford University Press, 2002. 32-37.

[5] Sheehan, Robert. Children's Perception of Computer Programming as an Aid to Designing Pr
Preston, UK: n.p., 2003.