Benefits of Computer Education

Jessica Gorman
Computer Systems 2008-2009
2nd Quarter Draft

June 8, 2009

Abstract

Computer science has become a more integral part of everyday life as technology advances. By teaching children computer science at a young age, they are able to better understand their technology-infused world. Because of its importance, a computer science program that uses the programming language Scratch was developed to educate students at Cardinal Forest Elementary School. This project focused its research on whether the computer science education benefitted students in ways other than simply gaining knowledge of computer programming.

1 Introduction

Does an education in computer science actually benefit young children? Perhaps they would be better off spending their time learning more about their core subjects. In order for computer science classes to be worth a young student's time, they should provide the child with something more than just a beginner's knowledge of computer programming. The goal of this research project was to investigate exactly how young students benefit from a computer science education and whether a certain type of child benefits more. For example, some children are very shy and prefer to work by themselves, while others are more talkative and willing to ask their classmates for help. Does a computer science program help the shy children learn the value of getting help from their peers, or perhaps the talkative children benefit more because they learn the importance of listening to the teacher's directions?

To answer these questions, I have worked with first and second graders at Cardinal Forest Elementary School who participate in a computer science education program. My partner, Crystal Noel, and I developed a curriculum for the two classes, one of first graders and one of second graders, and taught a thirty minute lesson each week for these students. Throughout the duration of the program, I observed changes in the children's learning behaviors and I developed a survey to be completed by the students enrolled in the program.

2 Background

2.1 Technology and Children

Because technology plays a dominant role in children's lives, it is important for them to gain an understanding about computers. Even at a young age, most children have been exposed to a vast amount of technology- technologies such as television, video games and computers. Therefore, it makes sense to also give children a computer science education, so they can begin to integrate their education into their everyday lives and better understand the technology around them. Learning about technology can also benfit young minds in other ways. For example, as students become more fluent in technology and create their own 'computer culture,' the students eventually realize that in order to create more advanced projects, they need to work together and ask for help. [2]

The focus of the research project is to determine the benefits of a computer education for children; recent studies have signified a strong possibility that there are other benefits that come from a computer education other than simply becoming more fluent in technology. Computer education takes advantage of the 'spiral of learning' in which students are encouraged to become creative thinkers as they imagine, create, play, share, reflect and imagine again. [4] By developing a computer science education program for the students at Cardinal Forest Elementary School, the students should benefit in multiple areas of their core curriculum.

2.2 History of the Cardinal Forest Scratch Program

Greg Gates, a student who graduated from TJ last year, created the Cardinal Forest Scratch program last year with the help of the school-based technol-

Figure 1: The first grade students working on their Scratch programs at Cardinal Forest Elementary School.

ogy specialist at Cardinal Forest Elementary School, Mr. Fred Allard. Gates set out to determine whether teaching elementary school students computer science was possible and he concluded that children became more technically savvy as the year progressed and they also became more interested in computer science, and technology in general.[3] The students participated in the program by working on Scratch computer programs developed by Gates and Allard. (See Figure 1.) Noel and I have helped continue the Cardinal Forest Scratch program, and we each chose different research topics to study during the program.

2.3 The Scratch Programming Language

In order to teach the students at Cardinal Forest Elementary School computer science, a kid friendly programming language needed to be used. Gates and Allard chose to use Scratch, a visually based programming language developed by MIT that allows children to drag and drop sections of code to build their programs.[5] (See Figure 2.) Mitchel Resnik, one of the develop-

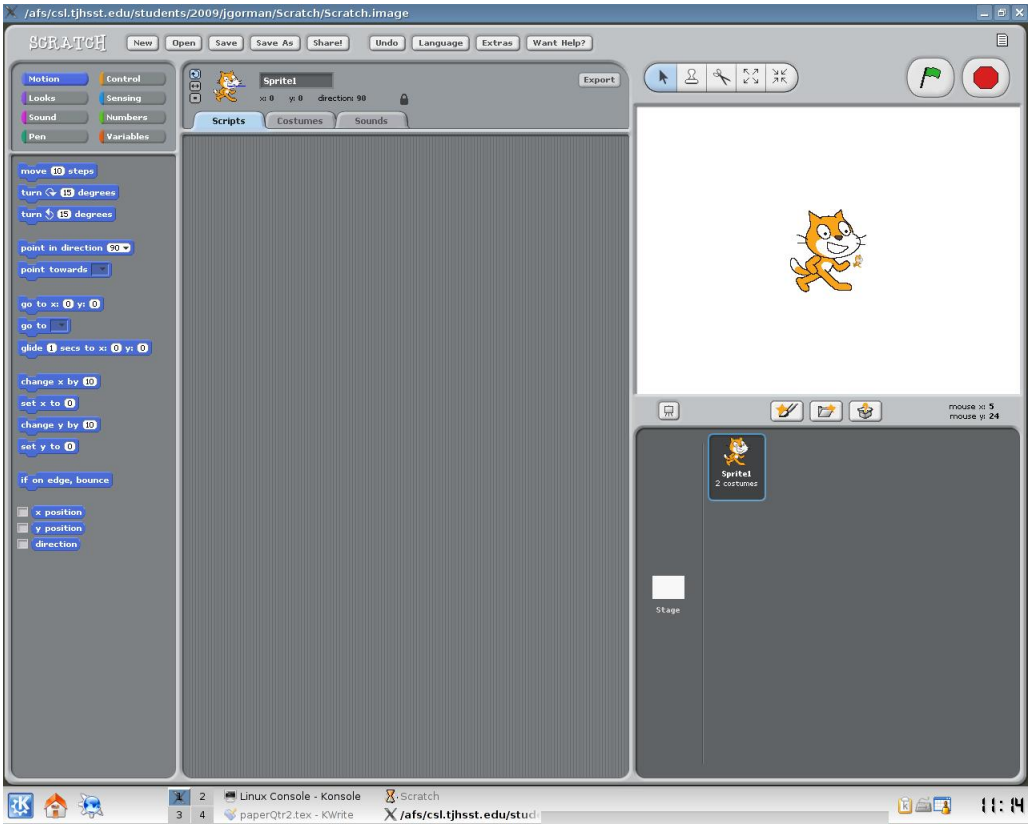


Figure 2: A screenshot of Scratch- notice the blocks of script on the left and the stage and sprite on the right.

ers of Scratch, has done extensive research on computer education and has concluded that children learn most effectively through a "spiral" of imagining, creating, playing, sharing, reflecting and imagining again. The Scratch program language was created as a way to help students incorporate spiral learning. The students can imagine a program and then easily build it with the Scratch programming language. The methods and visuals in Scratch are particularly useful for creating games and multimedia presentations, so children are able to have more fun with computer programming. Students are able to share their projects online through the Scratch website, fostering even more imagination and creativity as students can then see what others have done. [4]

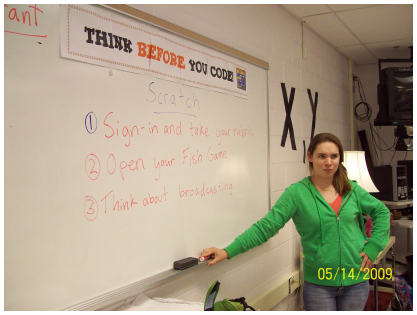


Figure 3: Crystal Noel writes the lesson plan on the board and waits to begin the Scratch session.

3 Developmental Sections

3.1 Lessons

By using the Scratch program, the students of Cardinal Forest Elementary School were taught basic computer science skills and problem solving techniques. Each week, curriculum was developed and then taught during the Thursday class sessions. (See Figure 3.) Using the knowledge gained through the curriculum, the children became more familiar with computer science and were able to manipulate the methods available in Scratch. Following is an overview of the projects that students completed.

3.1.1 Shapes Project

In order to give students a basic understanding of how the Scratch programming language works, a project was developed to teach them basic computer science techniques such as creating sprites (the characters used in Scratch), uploading a background, and moving their sprites in a coordinate plane. Because first and second graders have not yet learned negative numbers, much less coordinate planes, our team spent several weeks using number lines to teach the students about movement coordinate planes. Once the children had finally grasped the concept of ordered pairs, we began working on the Shapes Project. The students first uploaded the coordinate plane background and manipulated their sprite so it would move around all four quadrants, forming a square. The students learned the difference between the'go to' method,

which immediately transports the sprite to the x-y location, and the 'glide' method, which causes the sprite to move slowly giving the illusion that it is gliding. Once the students created the square, they learned how to manipulate the pen methods, which allowed the sprite to draw its square path as it moved. A rubric was created and given so the students could check off the steps they had accomplished. (see Figure 4.)

3.1.2 Winter Wonderland Project

Crystal Noel and I decided that the next logical step was to teach the students how to further manipulate a sprite by having it interact with other sprites and the background. The Winter Wonderland Project challenged the students to upload a sprite and edit it, giving it a second costume. The students also learned how to switch between backgrounds and were introduced to the concept of broadcasting, which allows sprites to communicate with each other. To introduce broadcasting to 1st and 2nd graders, Noel and I knew we needed a very simple definiton with real life examples. We designed an activity in which two children worked together to create a smiley face. One child had to draw the head as the other child was faced away from the board. Once the first child was done drawing the head he had to 'broadcast' that he was finished, so that the second student knew to turn around and finish the face by adding eyes and a mouth. The students then understood that their sprites would broadcast to the background to let it know they were done with their actions. This project also incorporated older lessons, such as 'glide,' and so it tested ability to implement what they've already learned and apply it to something new. (See Figure 5.)

3.1.3 Fish Game Project

In the Fish Game project, the students incorporated previous lessons, such as broadcasting, and used their skills to create an entertaining game. This project showed the students that computer science can be used to create interesting and fun programs that have a purpose. The Fish Game provided an opportunity for the students to learn how to move a sprite with the keyboard, how to keep score using variables and the concept of random. (See Figure 6.) The Fish Game was essentially a practice session so the students could understand how to create a game. The skills taught in the Fish Game project would allow the students to begin the next phase of learning, in which

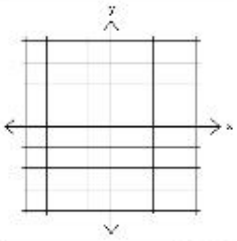
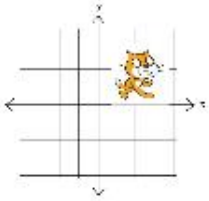
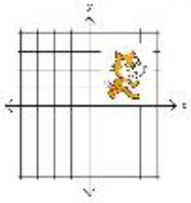
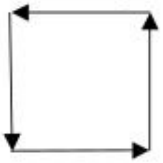
Task	Check?
Stage: 	
Does Kitty **start** at x: 100 y: 100? 	
Does Kitty **end** at x:100 y:100? 	
Does Kitty use **4 steps** to move in a square? 	
Bonus: Does Kitty **draw** the square as he moves? 	
Bonus: Does the square that kitty has drawn **disappear** when you start the program over again?	

Figure 4: Rubric for the Shapes Project

Part 1: Winter Wonderland Project Rubric

Step	Task	Check?
1	Stage: 	
2	Does your snowman have 2 costumes? 	
3	Does your snowman **start** at x:-100 y:-100? 	
4	Is the snowman wearing the costume that you drew for him?	
5	Does your snowman **glide** around the ice rink? 	
6	Does your snowman stay on the ice rink? 	
7	**Bonus:** Does your snowman glide around the ice rink forever?	

Figure 5: Rubric for the Winter Wonderland Project

the students would develop their own games and incorporate the skills they have learned.

3.1.4 Individual Game Designs

The last project for the year gave the students a chance to desgin and create their own game. The students spent the first several weeks working on problem solving activities where they created a rubric to break a problem up into steps. For example, they were shown a completed game and given blank rubrics. On the rubric, they drew pictures of the different steps that had been incorporated into the game. After completing the planning activities, the students, as a class, brainstromed some game ideas. The games were then narrowed down to four choices: an animated story, a Mario game, a Pac Man game, and a Dodgeball game. (See Figure 7.) Each student picked one of the four games and then each game group met together and discussed what they wanted their game to look like. Each student created their own individual project, but they were able to ask thier group members for help with scripting and planning. At first, the students were confused and constatnly asked for help from the teachers when they didn't have a tightly structured schedule. However, the students became more comfortable with their freedom and were able to problem solve more efficiently as the weeks went on. Many of the students were able to finish the first parts of their games and they will be encouraged to continue working on the games at home over the summer.

3.2 Measuring the Benefits of Computer Education

By using Scratch, the students of Cardinal Forest Elementary School were taught basic computer science skills and problem solving techniques. Each week, curriculum was developed to be taught during the Thursday class sessions. Using the knowledge gained through the curriculum, the children became more familiar with computer science and were eventually able to create their own programs. The Scratch program also influenced the student's progress in other academic aspects. In order to measure the amount of benefit the Scratch program had on the children, a survey was created and distributed to the students for completion. This survey measured the students learning style, learning ability, and personality.

Fish Game Rubric

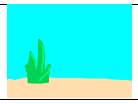
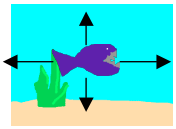
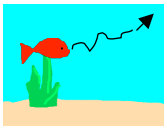
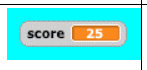
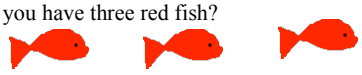
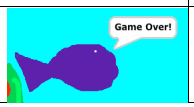
Step	Task	Check?
1	**Stage**: 	
2	Big Fish **Sprite**: 	
3	When you **press** the arrow keys, does the Big Fish move that way? 	
4	Red Fish **Sprite:** 	
5	Does your red fish move **randomly**? 	
6	Does your red fish **hide** when the Big Fish eats it?	
7	Does your red fish **show** in a different spot after it is eaten?	
8	Does your Big Fish change **costumes** when it eats the red fish? 	
9	Does your game keep **score**? 	
10	Do you have three red fish? 	
11	Does the Big Fish **say** "Game Over"? 	

Figure 6: Rubric for the Fish Game Project

Figure 7: One first grader shows off his Mario game.

A similar survey was then distributed at the end of the year to the students, along with a short assessment that asked questions on material encompassing the entire year. The survey measured the students' interest in Scratch and Computer Science, as well as their social growth. Students were asked if they enjoyed Scratch, if they worked on Scratch projects at home and if they had made new friends at Scratch. The assessment asked students to illustrate a x-line and a y-line and label the positive and negative sides. It also asked students to write a set of random numbers and to describe broadcasting.

4 Results

All types of children participated in the Scratch program this year. However, did a particular type of child benefit more from the program and what were those benefits? It was likely that shy students would benefit the most from Scratch because they needed to learn to work together and ask each other for help. These reserved students would need to go outside of their comfort zone

if they wanted to finish their programs and learn the techniques needed to do so. As the computer class taught these students the value of teamwork, they could apply this knowledge to their other classes and it would be evident that they were more open to working with others. Teamwork is a lifelong skill and it is never too early to learn it; if the Scratch program is able to help students benefit in the long run by giving them skills such as teamwork, it will be easier to convince other schools to implement computer courses in their curriculum. It was hypothesized that the more outgoing students would also benefit from the computer science program because they would learn that, instead of talking when the teacher's talking, they need to listen if they want to learn how to complete their program. As the students began to understand the importance of listening to directions, they should be able to apply this knowledge to their other classes and they would learn more effectively. Furthermore, as computers become more integral to everyday life and younger children are required to manipulate technology, it is more important for children to learn more about technology. Using computer science, children can learn more about what makes their electronic devices work, and can also learn how to problem solve and manipulate coding to make their program do what they want it to.

4.1 Survey and Assessments

In order to measure the benefits of the Scratch program, a survey was created and distributed to the teachers for of Session A students. Surevys were also filled out by the first grade students from Session B. Similar surveys were distributed at the end of the year, however the questions on this survey honed in on questions that I decided were more important over the course of the year. The original surveys showed a general trend that the students in the Scratch program were skilled in math and science, as well as problem solving. Therefore, the second survey focused less on an increase in math and science skills because it would be difficult to compare it to the open-ended wording of the first survey.

Instead, the second survey recorded the students' interest in computer science by asking whether they like Scratch and worked on it at home. About twenty-five percent of the students claimed to work on Scratch at home. This statistic, while slightly small, is still encouraging because it shows that these students are interested enough in computer science that they are willing to spend their time working on Scratch, instead of watching television or

playing with their PlayStations. Hopefully, this number will increase over the next several years as the program progresses. The second survey also asked students if they had made any new friends at Scratch. About 75 percent of the students replied that they had made new friends. Clearly, because the Scratch program encourages the students to work as a team, they were able to build new friendships. Learning to work as a team and make new friends is a monumental benefit for young students, as they will have to use teamwork for much of the rest of their lives.

The assessment that was distributed measured how well the students have retained their knowledge over the year. The first two questions, which asked students to draw and label a coordinate plane, were very difficult for students. Only ten percent of the first and second grader were able to correctly draw an x and y line and label the positive and negative fields. However, the other 90 percent essentially understood what the coordinate plane system was, they were simply unsure of which was x and y and which parts were positive and negative. These students originally had never seen a coordinate plane before, and they didn't know what negative numbers were. So, the introduction to the coordinate system was very difficult for them. Because the students at least understood what the coordinate plane should look like, I consider it to be a success and hopefully when the studetns reach Algebra, they will feel more confident as they have already been introduced to coordinate planes. The assessment also asked the students to create a random set of numbers. 98 percent of the students were able to correctly form a random set, which will help them understand math and science concepts better in the future. The students were also asked to define broadcasting, a Scratch term that describes how the sprites in Scratch can communicate with each other. While few students were able to correctly define the term to the extent that Crystal and I wanted, about half of the students were able could give a partial definition or an example of how broadcasting was used in class. Therefore, many of the students can clearly learn more complex computer science concepts and understand what they are used for. This is encouraging for the future of the program as it shows that some students will be able to retain their computer science skills as build on them next year without having to restart.

4.2 Observations

Throughout the year, I observed changes in students over the year. While there are countless examples I could use, I will focus on three particular

students and describe the benefits provided to them by Scratch.

4.2.1 Student A

Student A was one of our brightest second grader students; he was always able to answer questions and was willing to help his fellow classmates with their scripts. During our Scratch class time, Student A was very well-behaved, however, Crystal and I learned from our mentor, Mr. Allard, that this sudent usually had anger problems and would fight with other students when he was in his core classes. However, because he enjoyed Scratch and computer science so much, Student A understood that he needed to behave appropriately if he wanted to continue participating in the program. It was encouraging to see how this student was able to learn skills through the Scratch program that he couldn't learn in his core classes. Hopefully, Student A will learn to incorporate his teamwork and listening skills into his core classes, so that he can become a better student.

4.2.2 Student B

Student B is a more representative example of a group of students that benefitted from the Scratch program. This student had average computer science and math skills compared to the other students in the class. Student B was extremely outgoing and loved to use Scratch time to get to know the other students better. However, this socializing often caused the student to not pay attention when Crystal and I were discussing the scripts for the program. Therefore, this student constantly needed help catching up and was a distraction to other students. As the year progressed, the student grew more and more quiet when instructions were given, so that they would know what to do. Not only did the student learn the importance of listening to directions, but the quieter environment made it easier for other students to hear the directions.

There was a group of students who followed the pattern described above, especially in the second grade class. Because of the constant chatter at the beginning of the year, it was very difficult to make progress in the lesson plans because directions needed to be repeat four or five times. However, in February and March, the class made a noticeable inprovement in their ability to listen and follow directions; the amount of scripting completed in one class period increased significantly as the class needed directions repeated

less frequently. Therefore, many of the more talkative students learned to be respectful and quiet so that they could learn better.

4.2.3 Student C

Student C is also a representative example of several quieter students in the Scratch program. Student C was extremely intelligent, but was reluctant to answer questions, or offer assistancce to his struggling neighbors. Crystal and I began to ask the student to help his neghbors with their scripts. At first, he would help the other students only if Crystal or I asked him to. However, as the year progressed, the student began to open up more and would assist his neighbors without a reminder. Student C learned to share his knowledge with those around him, even if it was outside of his comfort zone.

There were several students among the first and second graders who followed the changes in behavior described above. These students are now much more confident of their abilities and are more willing to work with others. The Scratch program helped them develop better social skills, as well as gave them a chance to learn about computer science.

5 Conclusion

The Scratch program at Cardinal Forest Elementary School provided students much more than an opportunity to learn some basic computer science skills. The students also worked on social skills, such as teamwork and listening to directions. These social skills will assist the students in becoming more productive learners, which will translate into their other classes, as well as all future classes. While it was difficult measure the benefits of computer science on these young students, it was shown through observations and surveys that, in general the more extroverted students learned to quiet down while the quieter students learned to open up more to their classmates. Furthermore, many students made new friends through their teamwork in Scratch, discovered an interest in computer programming and and gained a better understanding of problem solving methods, and math and computer science concepts. Hopefully, Cardinal Forest Elementary School will continue to implement the Scratch program for their students, as it provides more benefits than simply giving the students an introduction into the world of computer science.

References

[1] Yasmin Kafai and Kylie Peppler, "Creative Coding: Programming for Personal Expression.

[2] Yasmin Kafai and Kylie Peppler, "Seeds of a Computer Culture: An Archival Analysis of Programming Artifacts from a Community Technology Center.

[3] Greg Gates, "Elementary Education in a Technology Age, 2008.

[4] Mitchel Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) In Kindergarten.

[5] Andres Monroy-Hernandez and Mitchel Resnik,"Empowering Kids to Create and Share Programmable Media.

[6] Andres Monroy-Hernandez,"Scratch: a platform for sharing user-generated prgrammable media.