Scratching the Surface: Kindergarten Programming TJHSST Senior Research Project Computer Systems Lab 2009-2010

Nick Grippin

April 7, 2010

Abstract

Programming is currently being taught at the high school and college levels; the earliest start a student might get is in middle school. While there are complicated concepts involved in programming, the basic problem solving skills and structure can be taught at a much earlier age. This research project involves teaching a Kindergarten course on programming through the use of Scratch, a programming language developed by MIT that focuses on visual output and creativity. The course will have a heavy math focus, tying in various concepts such as geometry and probability that are required according to the Math Standards of Learning (SOL). The students will be taught through a variety of methods, involving creating or deconstructing programs, hands-on activities and visual examples, as well as simple lectures. Assessment will be achieved through quizzes or student's answers to questions. Results showed that the students have problems with the intricate math concepts and basic computer usage, as well as attention issues. In spite of these hindrances, the Kindergarten class has learned to some extent concepts such as the coordinate plane, geometry, turning, waiting, starting on command, loops, and random numbers.

Keywords: elementary school, Scratch

1 Introduction

Currently, with a few exceptions, no elementary students are being introduced to the programming world. In middle or high school, they are suddenly plunged into a confusing world of computer jargon and algorithms. Programming can be used as a vehicle to teach any number of techniques and skills, including Math, English, Science, and other core classes. In this research project, I will be focusing on the math aspect, bringing in concepts from the Kindergarten SOL as well as more advanced topics that are necessary for Scratch programming.

1.1 Scratch Overview

Scratch was developed by MIT and released 2007. The program is designed especially for young children to teenagers, and as such is based heavily on visual aspects, sacrificing some of the more advanced coding abilities. The screen is divided into the a) stage, where the output takes place, b) script, for inserting and changing code, c) a list of current sprites, and d) the catalog of code, arranged under eight differently colored tabs (Maloney). The code, shaped into puzzle pieces, always directly influences a sprite, allowing it to perform any number of different commands or interact with other sprites. Through the use of dozens of colorful sprites, uniquely shaped code, and easy interface, the Scratch program is the perfect introductory tool to programming.

1.2 Previous Research

Previous research in this area has found that Scratch is well designed, discovering that students in higher grades can almost teach themselves through the program. The article *Programming by Choice: Urban Youth Learning Programming with Scratch* discusses how students ages 8 to 18 learned Scratch over a 18 month-long period. The researchers took programs written throughout the year, and analyzed how the students were learning, even if there were no instructors proficient in Scratch available in the classroom. The study, along with others, confirms that the well-built Scratch program is an excellent tool for teaching not only programming, but a variety of other topics as well. The students in the study were able to learn the "ins and outs" of

Scratch within a few weeks, and were creating complex games and animations by the end of the class.

1.3 Scratch on the Web

Scratch also provides an excellent opportunity to share projects through their website, "http://scratch.mit.edu/". It allows young programmers to post their Scratch creations and have other users play and vote on whether or not they enjoyed the project. A comments section allows the creator to read feedback, including both praise and constructive critism. I was able to put the Scratch website to great use, browsing through programs in order to discover new and inventive ways to approach various problems.

2 Background

In the previous two years, students have already done several research projects in this area. Jessica Gorman and Crystal Noel worked at Cardinal Forest last year, and helped students learn the mathematical techniques that the students needed to know. The main focus was on the coordinate plane, which required students to first comprehend the use of negative numbers. Gorman and Noel spent the first two months teaching these topics in order for the students to appropriately use the Scratch program. After the coordinate plane was well ingrained, they spent several months working on a basic program oriented around a winter theme in order to practically apply these skills in the Scratch program. The last part of the class was devoted to individual games of the student's choice, with the help of the Gorman and Noel to mentor the students in any other aspects of the language they needed to know.

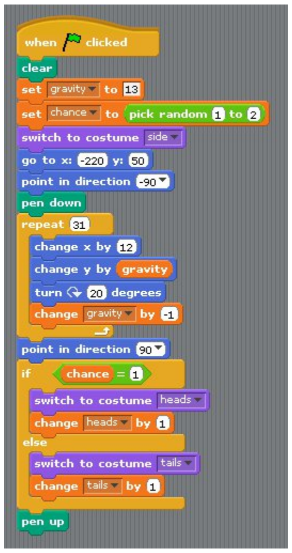
3 Development

Weekly lessons are currently being taught to a Kindergarten class at Cardinal Forest Elementary School. The expectation is that the students will have a working knowledge of programming the Scratch program by the end of the year. This is being achieved mainly through the construction of various programs that introduce concepts one at at time. For instance, one of the first programs taught to the students introduced the move and point in direction concepts, allowing the sprite to move to any position on the board. Later, the wait command was brought in, which caused the sprite to pause for however long the user wished. By bringing in commands one at a time, the students can have time to fully understand that concept before any more concepts are added. Once a firm foundation was built, I began to incorporate skills from areas other than programming. Scratch is an excellent way for young students to learn about programming skills. However, it is also a unique vehicle to introduce and teach concepts from fields other than programming. Math is the area that Scratch has the heaviest focus, and as such is perfect for implementation into the program. I have been building and focusing my lessons from both a programming and mathematical standpoint, bringing in concepts from the Kindergarten SOL. A very successful example of this was the lesson involving shapes, where the students created a program where they created sprites in four different shapes, then had to move them in certain directions. Lesson plans are structured with as many visuals and as few words as possible, with the assumption that the students are not able to read. They are taught in a variety of ways, from hands-on activities to straight step-by-step instructions. The latter seem to be better as far as time constraints, but hands-on activities and visual demonstrations always result in better comprehension and memorization.

3.1 Coin Flip

Later in the year, in order to study the problem-solving abilities of the Kindergarten class, I created a program to simulate the flipping of a coin. Keeping with the math focus, I used it as a springboard to teach the concept of a random number. In class discussions, the students were able to make the connection between random numbers and the flipping of a coin. Later on, they also made the connection between the piece of code that generates the random numbers in the coin flipping program. I had the students then try to create the program on their own, using a file with the pieces scattered around the scripts area.

All the students understood the need to have the 'When Green Flag clicked' piece first, but after that step their problem-solving abilities fell short. Most of the students simply put the pieces of code together in the scattered order, and when it didn't work as expected, they would not rearrange the code to try a different approach. I did not expect for the students to be able



```
when clicked
clear
set gravity to 13
set chance to pick random 1 to 2
switch to costume side
go to x: -220 y: 50
point in direction -90
pen down
repeat 31
    change x by 12
    change y by gravity
    turn 20 degrees
    change gravity by -1
point in direction 90
if chance = 1
    switch to costume heads
    change heads by 1
else
    switch to costume tails
    change tails by 1
pen up
```

Figure 1: The completed code for the coin flip program

to complete the program and get it running perfectly on their own, but I did expect to see some of the code in a logical order. Instead of looking at the code to see what it would do , the students would simply put the pieces together on a whim. While it does show that the students know how to use the Scratch program and are interested in learning how to create code, the students have not yet developed the necessary problem-solving skills.

4 Discussion

There are several problems that Scratch brings into the classroom whenever it is used with the Kindergarten level. The first, and largest problem so far is simply general computer use. Concepts like right clicking on the mouse, opening, saving, and navigating through files and folders are all foreign to the students, and must be taught or walked through step by step. The majority of the other problems come from the more advanced math topics that are not introduced in Kindergarten. These topics, such as the coordinate plane and degrees, also must be taught to some degree, even if it is only a basic understanding. Apart from the Scratch-related problems, Kindergarten students have a notoriously short attention span, which means that lessons must be

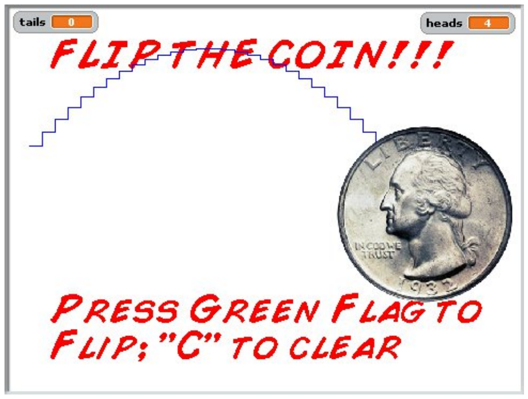


Figure 2: The completed code for the coin flip program

short, visually appealing, and must grab the attention of the student almost immediately. Even with these issues, it is still rewarding for both the class and the instructor when the students are able to memorize and apply concepts from either math and programming. A five-question quiz that I gave the students to see how well the understood various pieces of code ended with an average score of three. There were a few zeros that I did not understand from the students, until I asked how many quizzes they had taken so far. The answer was also zero. Given the fact that this quiz was their first, and was noticeably difficult for them, an average score of three was more than acceptable. Throughout the year, I have attempting teaching math both through the Scratch program and in straight demonstrations. While hands-on activities always seem to work best, the students always understand a math concept better after seeing it demonstrated in Scratch, thus confirming my theory that Scratch is an excellent vehicle for teaching in fields other than programming, even for young students.

5 Expected Results

As far as the student's ability to program in Scratch on their own, starting at the Kindergarten level is too early. The necessary problem-solving skills have not yet developed to the point where the students can think through the different approaches in order to decide which option would work best. Scratch can be used at this level as a visual example to teach the Kindergarten Math

SOL topics. Students have enjoyed the lessons that use Scratch for visual examples, and benefit from those lessons more than from lectures. There are several reasons to continue using Scratch with Kindergarten, primarily the greater understanding students have of topics taught with Scratch used as a visual example and as a way of increasing interest. Students also enjoy the creative freedom and simple way that Scratch approaches coding, which gives rewards that meet their efforts. It provides an excellent tool to shape lessons around, or even focus on the student's problem-solving abilities.

The project will be completed at the end of the year with the students having a basic understanding of the eight different sections of code, as well as being able to successfully create programs involving loops, motion, and user input. Students will also be expected to demonstrate a higher knowledge of general computer use, such as saving and navigating through files without help. A much more important goal is for every student to demonstrate full knowledge of the covered concepts in the Kindergarten Standards of Learning, as well as concepts from Scratch. The Scratch programming class will hopefully give them the tools and encourage them to pursue programming throughout the rest of their education.

References

[1] M. Resnick, "Scratch:Programming for All", *Communications of the ACM*, 2009

[2] K. Peppler, "Collaboration, Computation, and Creativity: Media Arts Practices in Urban Youth Culture", *Proceedings of the Conference on Computer Supported Collaborative Learning*, 2007.

[3] J. Maloney, "Programming by Choice: Urban Youth Learning Programming with Scratch", *MIT Media Laboratory*, 2008

[4] C. Lewis, "How Programming Environment Shapes Perception, Learning and Goals: Logo vs. Scratch", *Univerisity of California*, 2010