Computer Science for the Young Mind

TJHSST Computer Systems Lab Senior
Research Project Paper
2008-2009

Paul Im

10/30/08

Abstract

Technology has made tremendous leaps and bounds throughout the ages. More and more of today's work force has taken to the field of computer programming. But sadly enough, there has been limited effort to teach computer science at the elementary school level, which this project addresses. The purpose of this project is to implement computer programming to educate elementary school students in math and science.

1. Introduction

Since the beginning of civilization, mankind has made momentous leaps and bounds in technological advancement. From the Industrial Revolution of the 1880s to the dawn of the Digital age, because of how far we've come, we now live in a world once deemed unimaginable. At the center of this new world sits an adapting education system, where more and more people are taking up jobs in the field of computer science. Unfortunately, very little progress has been made at the lowest level of education in terms of technology: children.

Just how young is too young to start programming? (Gates, 2008) The purpose of this project is to answer this question by advancing an already successful computer science program at Cardinal Forest Elementary School via Scratch, a program developed by MIT. Along with fellow students Jessica Gorman and Crystal Noel, I helped provide teaching. Though not all the students at Cardinal Forest Elementary participated, enough of them did so to sustain the program.

2. Background

2.1 Computers, Children, and Education

Traditional computer science programs utilize traditional programming languages, such as Java, Python, or C++, all of which are geared toward high school and college students. The first attempt to teach computer programming to younger children was with Logo, which involved telling a turtle how to move around to make various

pictures. Since then, other preliminary programming languages, including Squeak, Alice, and Scratch, have been implemented with varied levels of success.

Unfortunately, most computers are used to reinforce allegedly outdated teaching methods, most commonly as a medium for transferring information. This method has proven to be very ineffective. As several studies have shown, students learn better when they immerse themselves in lessons rather than simply listen lectures (Gates, 2007).

The goal of this project is to continue the development of an ongoing computer science curriculum at Cardinal Forest Elementary School. First started in 2007 by Gregory Gates, the curriculum, inspired by "computer clubhouses" at the Massachusetts Institute of Technology (MIT), are weekly sessions that run from 30 to 45 minutes at a time. Not all students participate, but those that do are integral parts of the project.

2.2 Scratch

Scratch was developed and released in 2007 by MIT; it gets its name from its dynamic editing style, allowing users to edit programs as they're still running, similarly to how DJs mix music during performances. Instead of typing commands on various lines of code, the programming language editor uses simple, colored boxes and images to provide coding, as if assembling structures with building blocks (Fildes, 2007). Reminiscent of art programs such as Kid Pix, the colorful and intuitive interface makes it easy for users to tell that Scratch was specifically geared toward a younger audience, and users need only to drag boxes to designated programming fields and connect them together as they see fit. Users are even given the ability to create and edit custom sprites. As of 2008, the latest release is version 1.3.

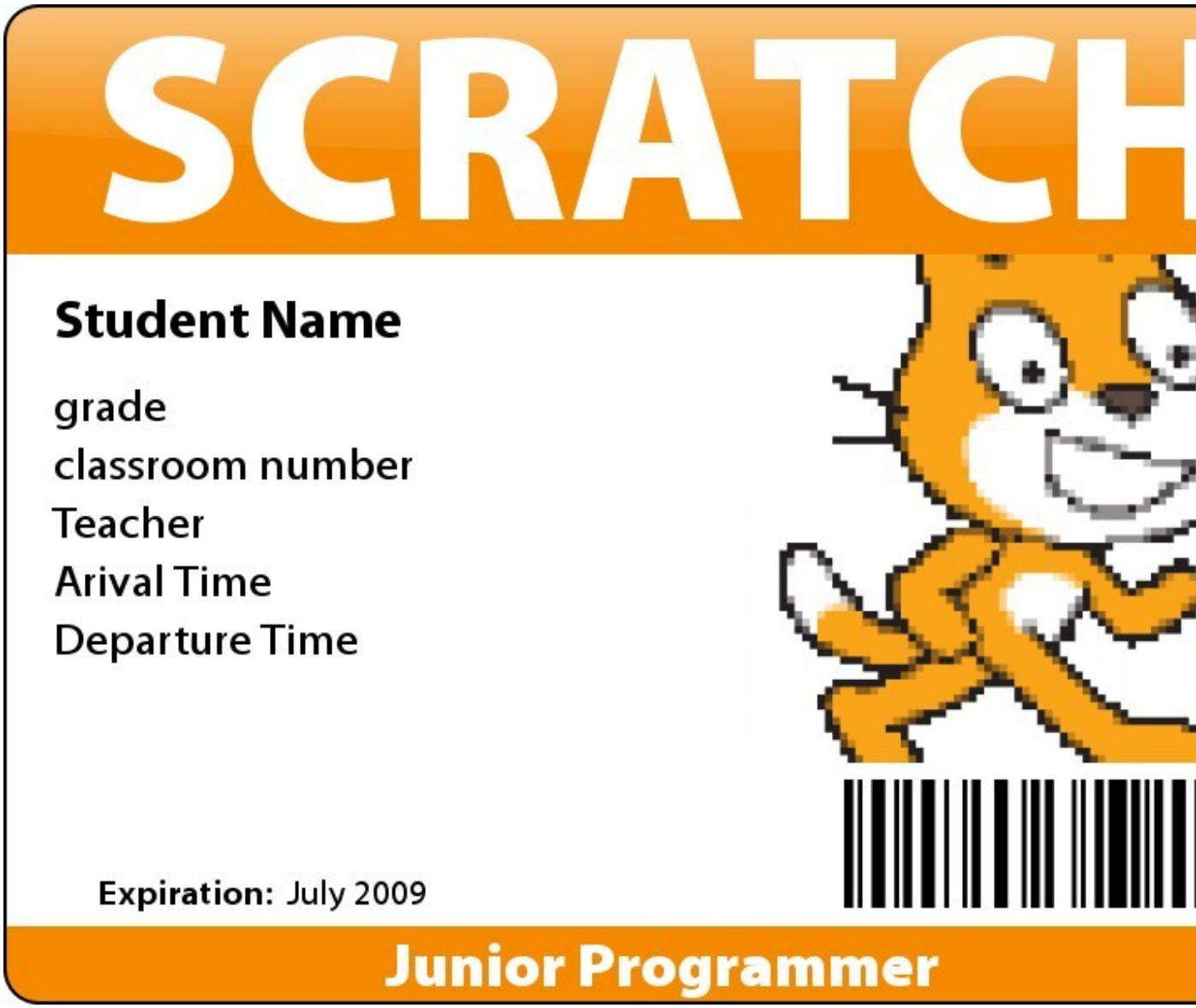

Figure 1: Student Badge for Scratch Classes

3. Procedures and Methods

3.1 Timeline

In September, I heard of the possibility of implementing a computer science program for elementary school kids and took up the opportunity after careful consideration. I contacted the principals of elementary and middle schools nearby

Thomas Jefferson High School inquiring about such a possibility. After weeks of hearing nothing from said principals, I received a call from Mr. Frederic Allard, the same teacher contacted by Gregory Gates last year.

We went to work shortly after I confirmed my willingness to participate. Also working on the project were Crystal Noel and Jessica Gorman, who had previously been chosen as successors to Gates' program, but were nonetheless more than happy to include me.

3.2 Class Structure

The program had to be expanded on in a series of steps. First, my teammates and I had to work out how we would divide students who had previously taken the Cardinal Forest Elementary School Scratch course from those who had not.

Unlike my teammates, I had no way of efficient transportation to Cardinal Forest Elementary. Instead, I used a teleconference system to broadcast live from TJ. The three of us all wrote different programs to help teach the children and studied various aspects of teaching in the process.

Teleconferencing allowed me to directly contact the students and faculty at the elementary school students without leaving Thomas Jefferson High School. In instances when I could do no such thing, I created program demos for Crystal, Jessica, and Mr. Allard to use in class for each lesson.

4. References

Feldman, Michael B., and Bruce D. Bachus. Concurrent Programming CAN be Introduced into the Lower-Level Undergraduate Curriculum. Ms. 16 Sept. 2008 <http://www.seas.gwu.edu/~adagroup/concurrency/bachus/>.

Fildes, Jonathan. "Free Tool Offers 'Easy' Coding." BBC 14 May 2007. 6 Oct. 2008 <http://news.bbc.co.uk/1/hi/technology/6647011.stm?ls>.

Gates, Gregory. TJHSST Computer Systems Lab Senoir Research Project Paper Elementary Education in a Technology Age. Ts.

Hazzan, Orit, Judith Gal-Ezer, and Lenore Blum. "A Model for High School Computer Education: The Four Key Elements that Make It!" ACM SIGCSE Bulletin 40.1 (2008): 281-285.

Hoffman, Mark E., Timothy Dansdill, and David S. Herscovici. "Bridging Writing To Learn and Writing in the Discipline in Computer Science Education." ACM SIGCSE Bulletin 38.1 (2006): 117-121.