Introducing Computer Science Through Animation and Virtual Worlds

Susan H. Rodger[*]
Department of Computer Science
Duke University
Durham, NC 27705
rodger@cs.duke.edu

Abstract

We describe a course for non-majors that teaches computer science concepts and programming by creating simple animations and building 2D and 3D virtual worlds. Students work with scripting languages, an interactive programming environment, a programmable modeling environment, and finish with a simple programming language. Students work in pairs on computers during class. Each student creates a web portfolio to display their work.

1 Introduction

All college students have used the World Wide Web, but few know what computer science is. We have developed the non-majors course CPS 49S [5] Animation and Virtual Worlds at Duke University to teach students computer science concepts and programming through simple animation and 2D and 3D virtual worlds. The first language students learn is HTML so that they can create a web portfolio of their work in the course. This is followed by a scripting language to build simple animations, a programmable modeling environment to create 2D worlds of decentralized systems, and an interactive programming environment to build 3D worlds in which they control the appearance and movement of objects. Finally, they finish with a simple programming language to move robots in a 2D world. During the course, computer science concepts are introduced with relation to the animation and virtual worlds.

Being a non-majors course, the course targets students with no experience and no interest in majoring in computer science. This type of student may be more hesitant to learn programming. By presenting programming through objects in a visual and animated way, students find programming fun. To limit frustration, the course is taught in a room with computers. During class students work on these computers in pairs, receiving assistance from the instructor and teaching assistants.

In Section 2 we describe related courses. In Section 3 we give an overview of CPS 49S and in Section 4 a description of topics in the course. In Section 5 we evaluate the course and in Section 6 we list future changes.

2 Related work

We list two other courses that create virtual worlds, both requiring students to have experience in computer science, art, or some other discipline. At Carnegie Mellon there is a course entitled Building Virtual Worlds [6]. This course is an upper-level course with a mix of students from several disciplines including art, architecture, design, drama, and computer science, including undergraduate and graduate students. There is a large programming component with the focus of the course building several virtual worlds using Alice and more recently JAlice. At Bowling State Green there is a course on Virtual-Reality Art [12]. In this course the students have previous experience in either computer science or art, and the focus is on building Artistic Virtual Environments using the WorldUp Toolkit [11].

3 Overview of the Course: CPS 49S

3.1 The Students

Freshmen in the College of Arts and Sciences at Duke University are required to take a special topics seminar their first year, a chance to pursue a special topic in some depth with 15 or fewer students. CPS 49S was developed as one of these seminars in conjunction with Project Advance, an interdisciplinary program at Duke

[*] The work of this author is supported by the National Science Foundation through grants NSF HRD-9979478 and NSF DUE-9752583.

University for women interested in mathematics, computer science and statistics. After Advance students have signed up, other students could sign up for the course. In the spring of 2001, the course had twelve women and three men.

3.2 The Classroom and Format

The classroom is designed for teaching with computers. There was one computer for each pair of students. These computers are arranged in a U-shape around the sides and back of the room so that all screens can be seen by the instructor from the front of the room. Students face the front of the room for discussions and turn around facing the back or sides of the room to work on the computer.

CPS 49S uses this room in a workshop format. The instructor gives a short lecture at the beginning of class to introduce the topic, then students turn around and work on the computer. The instructor can see when people are having trouble and can get everyone to turn back around during class for a short discussion or at the end of class for a wrapup. During class students work in pairs most of the time, but a few activities used groups of size 4. All pairs and groups were assigned and changed every few weeks. Outside of class students worked on additional assignments individually.

CPS 49S was offered during the spring of 2001. We met two days a week for 14 weeks, 75 minutes each class period. The longer class period is helpful when students are engrossed in solving a problem.

3.3 The Syllabus

The syllabus for the course consisted of five main units. Students begin the course learning how to build web pages so that they could create a web portfolio. In the next unit they created simple animations using JAWAA. Next using StarLogo they created 2D simulations of decentralized worlds. In the next unit using Alice they created 3D simulations of worlds. Finally in the last unit they used Karel++ to create 2D worlds of robots.

4 Topics in the Course

In this section we describe the topics in the course and the length of time spent on each topic.

4.1 Web Pages

Each student created a web page portfolio with all the material from the course. Many of the works are textual descriptions and pictures, and some are real animations.

In the first unit of the course for one and a half weeks, students learned HTML and created several simple web pages. Initially they were given a web page that was mostly written and they had to learn how to setup the web page and fill in some of the blanks with text. Later they went through two HTML tutorials and created web pages about an animal of their choice. These web pages had certain requirements including a picture of the animal, ordered lists, unordered lists, definition lists, tables and links.

We discussed two computer science concepts during this unit. The first concept was algorithms and preciseness. They wrote algorithms to make peanut butter and jelly sandwiches and the instructor built such sandwiches in front of the class using their algorithms and showing where they were not precise. These algorithms were written up as one of their first web pages. The second concept we discussed was sorting, described below.

4.2 Sorting

We spent one day with students investigating sorting. Students were assigned groups of size four, and each group was given an envelope with over 100 big words, each word on a separate strip of paper. The first part was to sort the words in alphabetical order and arrange them so they could answer questions about the words. The second part was to write an algorithm describing the method they used to sort their words. Finally, the third part was to answer 11 questions about the words including “How many words begin with the letter ’m’?” and “What is the 100th word?” The answers to all parts were put up on a web page.

This assignment is also given in Duke’s CS 1 course. It is interesting to observe the groups developing sorting algorithms. We’ve seen students develop forms of insertion sort, bucket sort, mergesort and selection sort.

4.3 Animation with JAWAA

JAWAA [7] is a tool for creating animations easily over the web. With JAWAA one can create simple primitives such as circles, rectangles, lines and text, and move them around singly or in a group. One can enter JAWAA commands in a file, modify a simple HTML file, and easily generate an animation that runs on a web page. This unit was for one and a half weeks.

The first classwork consisted of two animations. The first animation was to experiment with JAWAA and required each pair of students to create several simple objects, move the objects, group some of the objects, and move the group. The second animation gave the groups some freedom in the topic, but had several requirements such as one of each primitive type, some object that moves as a whole and has parts that move separately, and changing the color of an object. For example, one group created a pig that wiggled its nose and then its behind.

The second classwork was a traffic simulation. Requirements included showing parts of two roads that intersect, a traffic light that cycles through red, green and yellow lights, cars that obey traffic lights, and a car crash with parts of the car flying off. Figure 1 shows a student traffic animation.

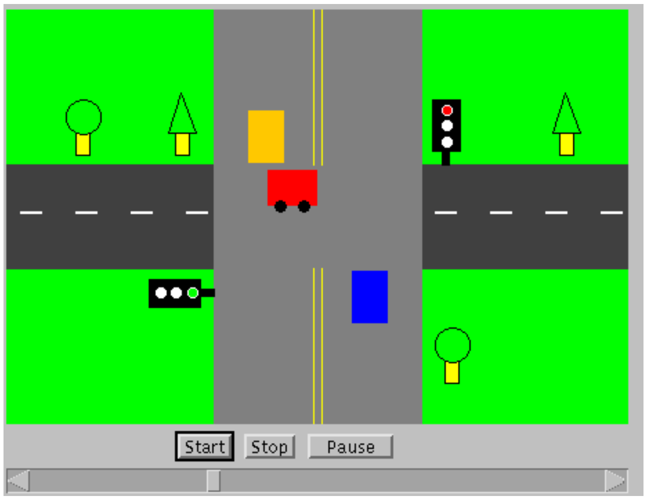

Figure 1: JAWAA Traffic Animation

The third classwork was a sorting assignment. An animation had been started for them that created 10 upright skinny rectangles of different heights. Their task was to come up with an algorithm for sorting the rectangles based on their height and complete the animation using their algorithm. We had not talked about any specific sorting algorithm. All the groups created either selection sort (starting with smallest or largest) or insertion sort. Figure 2 shows a snapshot of a student sorting animation that was selection sort by the largest.

4.4 StarLogo

We spent three and half weeks on StarLogo [8, 10] from the MIT Media Lab. StarLogo is a programmable modeling environment for exploring decentralized systems, systems that are organized without an organizer. StarLogo is an extension of Logo. In Logo one can create and control one turtle, and in StarLogo one can create thousands of turtles that move in parallel.

To create a StarLogo project, one enters control commands in one window and creates buttons and sliders in a window where the simulation will be viewed.

In the first week, students learned StarLogo through a tutorial and created a termite and wood chip world in which termites pick up and move wood chips. Each wood chip is represented by a yellow square and each termite is represented by a red square. Each termite randomly moves picking up a wood chip and carrying it until it bumps into another wood chip and then puts its wood chip down in a nearby empty space. Eventually, all the wood chips end up in a single pile.

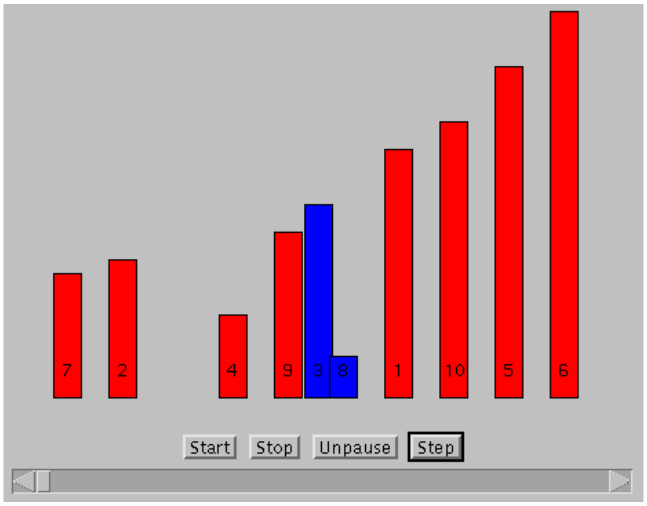

Figure 2: JAWAA Sorting Animation

In the second week, students created a traffic simulation that had similar requirements to the JAWAA traffic assignment. Figure 3 shows a student Starlogo traffic project. The student also creates buttons to control the animation but they are not shown.

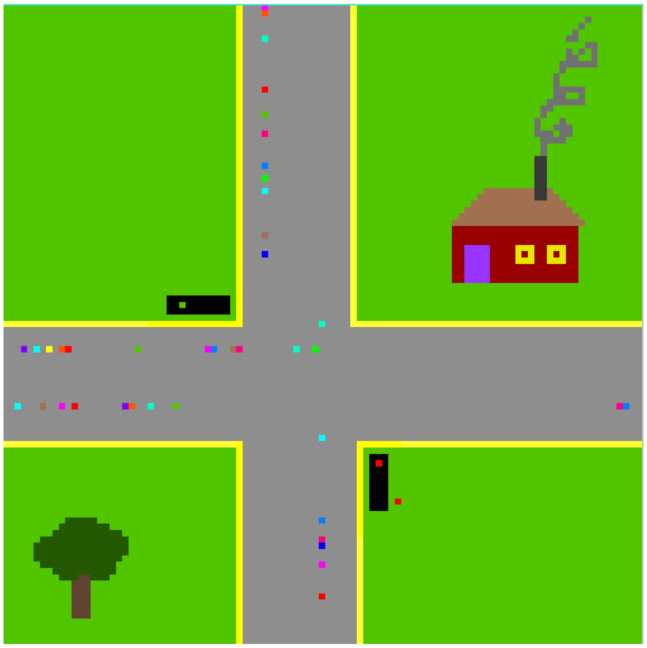

Figure 3: StarLogo Traffic Simulation

The StarLogo procedure shown below from the students' project shows how to move some of the horizontal cars which are turtles numbered starting at 6.

```
to horizontal-cars
if who > 5 and who < numcarshorizontal + 6
   [setycor -13]
if who > 5 and who < numcarshorizontal + 6
   [setxcor random -50]
if who > 5 and who < numcarshorizontal + 6
   [seth 90]
if who > 5 and who < numcarshorizontal + 6
   [loop [fd 1] ]
end
```

In the third week, students created any StarLogo project, describing it with a web page. Projects created include bunnies eating grass, fish and waves in water, an ant navigating in a maze, fireworks, the game of pong, a swim meet, and feeding fish in a bowl. The last day of the unit was spent presenting projects.

4.5 Alice

Alice [1] is a 3D interactive graphics programming environment created by the Stage 3 Research Group at Carnegie Mellon University. Using Alice novices can easily create a 3D world using 3D objects that can be selected and controlled. Many objects have multiple parts that can be controlled individually. We spent one and a half weeks on Alice, plus another day for demos.

The first day of Alice was spent going through the Alice tutorial provided with the tool. Then one week was spent on creating an Alice world with specific requirements: a flying logo, coordinated movement, interactive action, making a 3D object using Teddy and including the new object in the world. After completing the world, students put it on a web page with a description of how to interact with the world.

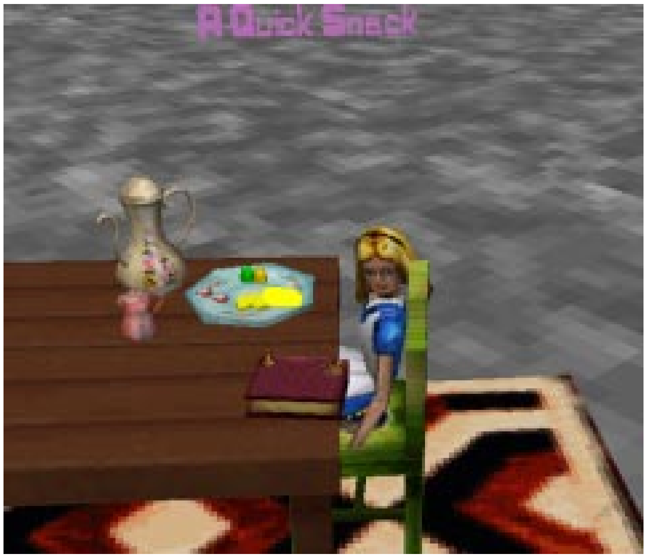

Figure 4: Alice Eating A Snack

Figure 4 shows a snapshot of a student project in which Alice is having a snack. When the world starts, Alice is not there. Click on the chair and Alice comes in and sits down. Click on the food and Alice reaches over and eats something from the plate. Click on the book and Alice opens the book and looks at it.

4.6 Being a Robot

One day was spent on learning how to be a Robot [9] in preparation for our next unit. In this non-computer exercise, students learn how complex it is for robots to complete a task. Groups of four students act out a robot and try to build a specific object out of blocks.

One person is the "head", one person is the "eyes", one person is the "left hand" and one person is the "right hand." The head is blindfolded and is the only person who knows what to build. The eyes answer questions asked by the head. The hands are blindfolded, can move the blocks and cannot speak.

The head is taken aside and shown what the team is to build, then returns to the group and is blindfolded. The head then gives instructions to the hands (also blindfolded) to build the object. The head can ask the eyes questions and the eyes can give simple responses.

4.7 Karel++

Five weeks were spent using Karel++ [3], a tool for programming robots who move graphically in 2D worlds. Users control robots in a city by specifying their direction through streets and around barriers, picking up and putting down beepers. Karel++ was a nice way to finish the course, with a substantial but gentle programming component. Many others have used some form of Karel the Robot to introduce programming in high school, in CS 0, and as an introduction in CS 1 [2, 4].

In CPS 49S, students started with a tutorial that is available on the Karel++ web site, and then class periods consisted of writing programs on the following topics: creating new types of robots, modifying inherited instructions, conditional instructions, nested conditionals, looping structures, solving a maze, and recursion.

For example, after learning about creating new robots who inherit instructions, Figure 5 shows a task in which students are suppose to create four robots, some who are private helpers, and move the beepers (shown as circles) from the starting positions shown on the left to the positions shown on the right.

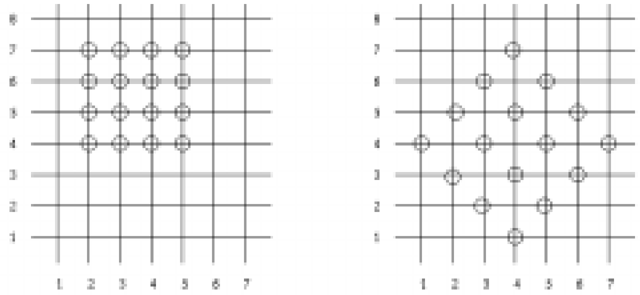

Figure 5: Moving Beepers in Karel

5 Evaluation

In this section we describe the results of three evaluations given on the dynamics and topics of the course.

Several questions asked about working with partners. The majority of students liked working with partners stating that two partners resulted in better ideas. For those that did not like working with partners, the main complaints were that the partner typing would learn more, and sometimes one partner would do more work.

We will describe the topics in the course and their favorability. For the web page unit, 100% of the class loved creating web pages and they especially liked creating a web portfolio. For the JAWAA unit, only about 50% of the class reported that they liked this unit. JAWAA is designed for programmers and is easier to generate animations as output from a program than typing commands in a file. For the StarLogo unit, 75% of the class reported that they liked this unit. For those that did not, they complained about having bugs in their program that were difficult to track down. Students demoed StarLogo projects during class and many had clearly put a lot of time into their programs. For the Alice unit, this was clearly the favorite in the class with 100% of the students reporting they liked this unit. The Alice interface was the easiest to use, its tutorial was the clearest of all the units and its programming is interactive. For the Karel++ unit, 75% of the class reported that they liked this unit. Some students thought it started off at the right pace but at some point it got too complex. The feedback on the final evaluation from the course was very positive, with students giving the course high ranks. Five of the fifteen students are now taking Duke's CS 1 course this semester.

6 Conclusions and Future Work

We plan to offer the course again with the following changes. Since this course was offered, JAWAA has been updated to include a graphical editor that can be used by novices to interactively generate an animation instead of typing JAWAA commands in a file. Using this editor would alleviate many of the frustrations students had with this tool. Also since the course was offered, StarLogo has been updated to allow one to run their StarLogo projects on the web. In addition, the unit on StarLogo needs to be broken up into smaller pieces so they learn a few concepts at a time. Alice was by far the most popular tool we used, but the less like programming with its interactive interface. We would move the Alice unit before StarLogo and include some python programming with Alice.

7 Acknowledgements

Thanks to Robert Duvall and Owen Astrachan for topic ideas for this course.

References

[1] Alice. http://www.alice.org.

[2] Becker, B. Teaching cs1 with karel the robot in java. *Thirty-second SIGCSE Technical Symposium on Computer Science Education* (2001), 50–54.

[3] Bergin, J., Stehlik, M., Roberts, J., and Pattis, R. *Karel++ A Gentle Introduction to the Art of Object Oriented Programming.* Wiley, 1997.

[4] Buck, D., and Stucki, D. Jkarelrobot: A case study in supporting levels of cognitive development in the computer science curriculum. *Thirty-second SIGCSE Technical Symposium on Computer Science Education* (2001), 16–20.

[5] CPS49S.
www.cs.duke.edu/courses/spring01/cps049s.

[6] Pausch, R. Building virtual worlds, 2001. http://www.alice.org/building.htm.

[7] Pierson, W., and Rodger, S. H. Web-based animation of data structures using jawaa. *Twenty-ninth SIGCSE Technical Symposium on Computer Science Education* (1998), 267–271.

[8] Resnick, M. *Turtles, Termites, and Traffic Jams.* MIT Press, 1994.

[9] Rodger, S. H., and Walker, E. L. Activities to attract high school girls to computer science. *Twenty-seventh SIGCSE Technical Symposium on Computer Science Education* (1996), 373–377.

[10] StarLogo. http://www.media.mit.edu/starlogo.

[11] WorldUp. WorldUp http://www.sense8.com.

[12] Zimmerman, G., and Eber, D. When worlds collide! an interdisciplinary course in virtual reality art. *Thirty-second SIGCSE Technical Symposium on Computer Science Education* (2001), 75–79.