

ACTIVITIES TO ATTRACT HIGH SCHOOL GIRLS TO COMPUTER SCIENCE

Susan H. Rodger
Department of Computer Science
Duke University
Durham, North Carolina 27708-0129
email: rodger@cs.duke.edu

Ellen L. Walker
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180-3590
email: walkere@cs.rpi.edu

ABSTRACT

We present several activities used in the two-week PipeLINK summer program for high school girls. These hands-on activities and interactive talks, presented mostly by female faculty, undergraduates, and graduate students, showed the girls the wide range of opportunities in the field of computer science

1 INTRODUCTION

Women are underrepresented in the field of computer science, particularly at the highest levels. Although women and men study computer science in high school in equal numbers, as they progress through the “pipeline” of more advanced study, the percentage of women drops at each educational level[3, 6, 7]. The primary goal of the PipeLINK program [8] is to attract and retain women in the computer science pipeline, increasing the number of females at each educational level from high school through the Ph.D.

At the high school level, the main goal of the PipeLINK program is to attract more girls to the field of computer science. A major problem is that most girls do not know what computer science is. In those high schools that have *computer* courses, these courses are either programming, keyboarding, or wordprocessing, not an overview of computer science. Many programs concerned with attracting high school students to computer science concentrate on a particular theme, e.g. computational science using high-performance computers [2], solid modeling [5], or C++ programming [4]. An alternative [10] is a one-week computer appreciation program whose activities were related to communications (email and chatting) and professional applications (desktop publishing and library). Like this program, we wanted girls to learn about the wide range of

areas in computer science, introducing them to a number of areas including programming languages, grammars, artificial intelligence, computer vision, robotics, graph theory, algorithm animation, symbolic computation, databases, history of computing, VLSI, computer and human interactions, and distributed computing. As a result, two weeks were necessary for the number of topics we wanted to cover.

This paper overviews the PipeLINK program, which emphasizes hands-on activities to introduce various facets of computer science. In Section 2 we describe the high school component of our program, in Section 3 we describe the hands-on activities, and in Section 4 we describe the non-laboratory activities. In Sections 5 and 6 we provide an evaluation and conclusion.

2 THE PIPELINK PROGRAM

The high school component of the PipeLINK¹ program provides female mentors at different levels of the pipeline, connecting women and girls through an electronic system containing email, chat groups, and bulletin boards, and providing hands-on activities for girls to learn about computer science. Sixteen high schools in the Albany, New York area were selected for participation in the PipeLINK program due to their close proximity to Rensselaer. These high schools represent a mix of public and private schools, female and co-ed schools, and rural and urban schools. Girls in grades 10-12 were encouraged to participate in all aspects of our program. The program has an academic year component and a summer program.

In the academic year 94-95, the schools were provided electronic connections to Rensselaer, one day visits by a female faculty member and student, and activity days at Rensselaer. To connect the schools, each school was provided a modem and girls were given accounts on the PipeLINK system, a computer at Rensselaer. Each high school was visited for one day by a female computer science faculty member and either a female undergraduate or graduate student. The visit was coordinated with the high school teacher. At some

¹Supported in part by the National Science Foundation's Directorate for Education and Human Resources under the Model Projects for Women and Girls through grant EHR-9450007.

schools, presentations to large groups were given and at other schools, presentations were made to several small classes. Two activity days held at Rensselaer gave the girls a chance to learn about computer science in a hands-on manner.

During the summer of 1995, the PipeLINK high school program consisted of a two-week residence at Rensselaer including free room and board, that gave twenty high school girls a broad overview of computer science through hands-on activities, talks, and interaction with many female role models. The teacher contact in each of the sixteen high schools worked with other teachers and guidance counselors to nominate one student and one alternate with sufficient background to pursue computer science, but not necessarily with a background in computing. The majority of the students selected had just completed 10th or 11th grade, and only a few of them had taken a programming course. The hands-on activities were usually 2 1/2 hours in length, and the talks ranged from 20 to 75 minutes. The female role models included ten undergraduates engaged in research for the summer, six graduate students, six professors and eight women from industry. The undergraduates were lab assistants and they coordinated additional non-computer-science activities in the evenings.

3 SUMMER HANDS-ON ACTIVITIES

Our goal in the summer program was to introduce students to a variety of computer science areas with hands-on activities that would teach them about a concept, get them started working with the concept, and then leave time for them to explore on their own. Each lab was staffed by one professor and four to five undergraduate assistants, providing a one to four ratio of helper to student. Many of the labs resulted in some type of picture, which students copied and added to their homepage (see Section 3.5). This section describes four lab activities, three of which were computer-based.

3.1 C++ PROGRAMMING

This lab was an introduction to programming using C++. Only a few students had seen programming, in either Pascal or Basic, so C++ was new to everyone. Students were taught functions, input and output, and a conditional statement through a series of three programs that they would modify.

All three programs focused on drawing a face [1]. The first program was one main function composed of output statements that drew a picture of a face. Each output statement drew a different horizontal section of the face (ears, eyes, mouth, etc.). The students were taught how to compile and run the program.

The second program drew exactly the same face, but now each section of the face was replaced by a function

and a function call in the main function. After explaining functions, students compiled and ran the program to see that the result was exactly the same as the first program. Several exercises were followed to make modifications to the program: creating lots of hair by calling the Hair function several times, swapping the calls to the Eyes and Smile functions to create a Picasso face, and adding a Nose function.

The third program was a suspect sketch program. This program had multiple functions for hair (curly, straight and parted) and multiple functions for ears (big and small). The main function was now divided into three sections. The first section was an introductory output message. The second section used a string class object, asked the user what type of hair, and then stored the users response. The third section drew the face, replacing the call to hair, with an if statement for the different types of hair. After explanations of strings, input and conditional statements, students compiled and ran this program to see what it did. Then a series of exercises led them to create their own suspect programs, making modifications for types of noses, chins, necks, mouths, eyes, or hats. Output from a student program is shown in Figure 1.

```
Pinky's Detective Agency

Describe the suspect's face and we'll sketchout a composite!

What type of hair (parted, straight, or curly)? curly
What type of ears (small, or big)? small
what type of eyes (glasses, or small)? glasses

      oooooooooooooo
      oooooooooooooooooo
      |-----|
      | \---,---/ |
      |  \---/  |
      |   u   |
      |-----|
      |-----|
      |-----|
      |-----|
      |-----|
```

Figure 1: Face Sketch program

3.2 ANIMATION PROGRAMMING

In this lab, students experimented with Xtango [9], a tool for developing algorithm animations. Simple animations can be written in Xtango using one-line commands to declare or move objects. These commands are then interpreted by an *animator* program, producing the animation. The commands can be produced as output from a program, but to keep things simple, students just typed the animation commands into a file. This eliminated worrying about programming language syntax in addition to the animator syntax.

Through a series of exercises, students wrote an animation for a traffic light with cars moving through the intersection on a green light. Initially, the students were given a short animation program that created two rectangles of different colors and the phrase *Your Name*.

The students changed *Your Name* to their name and moved their name to the top of the window surrounded by the two rectangles. In the remaining exercises, students created a traffic light, cycled through the lights in the traffic light ending in red, created a road (horizontal lines) and cars (rectangles) that drove up to the light and stopped, changed the light to green, and moved the cars through the intersection. In Figure 2, a student’s animation shows three cars stopped at the light, and the light is turning from red to green.

For the remaining lab, students created their own modifications which included changing the road to a hill, moving cars to the side of the road to let an ambulance pass, crashing cars into each other, and adding story lines. Students had fun when incorrect changes were made and cars went flying off the road.

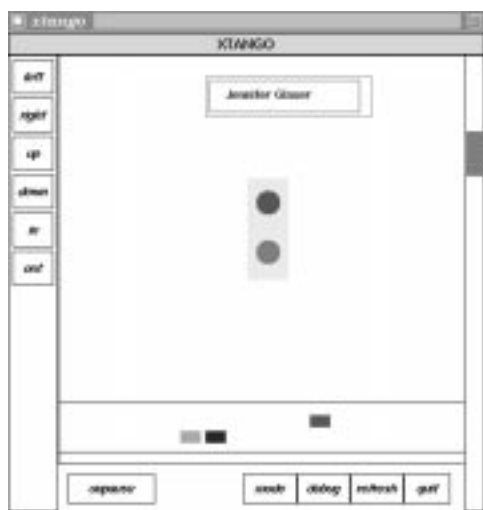


Figure 2: Traffic Light Animation

3.3 STATISTICAL REASONING GAME

In this lab, students were introduced to concepts of uncertain reasoning and expert systems, using a tool called *statrad* (**S**tatistical **R**easoning: **A**ce **D**etective) [11]. This tool allows students to learn about uncertainty in expert systems. Expert systems were introduced using the analogy of a detective solving a case.

In preparation for the lab, the students were introduced to the terminology: Belief and Plausibility as the lower and upper bounds of a certainty range. This was explained in terms of an example: “Suppose you believe that you will get a B in a class, but it is plausible that you will get an A? How surprised will you be if you get a B? an A? a C?” We also discussed how in a mystery, the detective will not always be entirely sure of the suspect, but will have a range of certainty for each one. Therefore, to decide how to act, the detective must consider both the belief and plausibility value of each suspect.

The *statrad* tool was introduced via a simple mystery (two suspects and two clues). The tool displayed a graph connecting evidence to supported hypotheses (e.g. “lefty” connected to left-handed suspects). Students clicked on the evidence as the story unfolded and could monitor bargraphs that represented the belief and plausibility of each suspect.

Once the students were familiarized with the graphical representations, they played a game similar to 20 questions, where each click on a piece of evidence yielded information as to whether that evidence had been observed. If the evidence was observed, the program updated all belief-plausibility bar-graphs. The goal of the game was to choose the right suspect with as few questions as possible, and to correctly classify their confidence in the choice. After playing for a short time with the simple murder case, students moved on to a more challenging game based on medical diagnosis expert systems.

In the final part of the lab, the students used the *statrad* tool to create their own “game”. First, they decided what the evidence and the possible suspects would be, and created an “environment file,” which they loaded into the *statrad* tool, and set up rules for their environment. Students invented a variety of games, one of which was a game to figure out who is on the other end of the telephone by voice quality (shown in Figure 3).

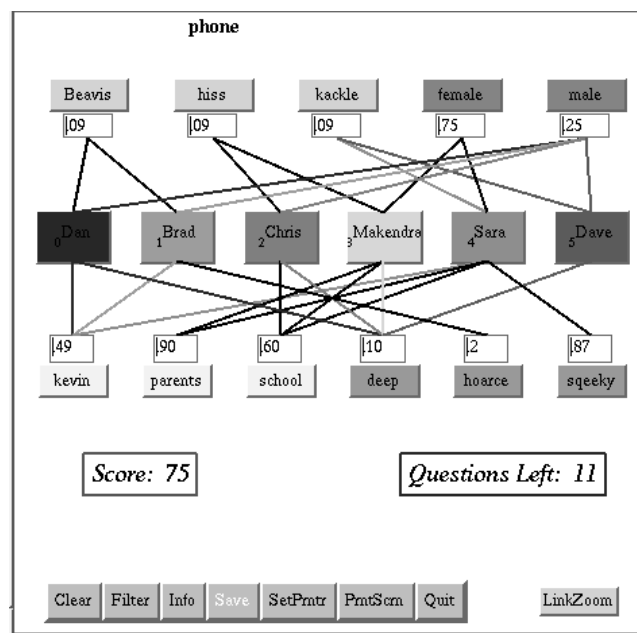


Figure 3: Graph of Suspects and Evidence for a Game

3.4 BE A ROBOT

The purpose of this noncomputer activity was to teach a little about robotics, including some of the dif-

difficulties encountered when programming a robot. The students were divided into groups of four, with each group simulating a robot building a sequence of increasingly difficult structures out of children's toy blocks.

In each group, each person had a designation: head, eyes, left arm, or right arm. The person designated "head" simulated the robot's control program. Only she knew what structure was to be built. She was blindfolded and not allowed to touch the blocks, so she had to depend on information given to her by the other participants. The person designated "eyes" simulated the robot's vision system. She watched the rest of the group and reported on what was going on, but was only allowed to answer direct requests of the head. The two people designated "left arm" and "right arm" represented the robot's actuators. They were both blindfolded, and each was allowed to use one hand to manipulate the blocks on the table. They were only allowed to manipulate the blocks according to direct commands from the head. A lab assistant was assigned to monitor each group and to tell the group when they had achieved their goal.

Each group built a sequence of four models out of their blocks. Each model used three to five simple geometric blocks from a set including a rectangle, a square, two sticks, and two cylinders. The first model was a simple stack of three blocks. The second was an "arch," which required them to balance a third block on top of two columns. The third model was a "cannon," with one block forming the gun barrel cantilevered by another block on top of it. This model could not be built with a single hand, so the efforts of the two hands had to be coordinated. Figure 4 shows the arch and the cannon. The final model was a five-block "tower," which required accuracy and balance. After a group completed a structure successfully, the members of the group changed places before trying the next structure. In approximately one hour, four of five teams completed all four structures, and the fifth team completed three of the four. At the end of the activity, we discussed teams' successful strategies and common pitfalls.

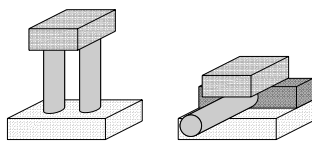


Figure 4: Arch and Cannon

3.5 ADDITIONAL HANDS-ON ACTIVITIES

We briefly describe some of the other hands-on activities used in the PipeLINK program.

In a *Making a Home* activity, each student created a home page on the World Wide Web. Beforehand, a

graduate student created a set of web pages for the summer program, including a page for the schedule (with links for all handouts), and a page listing all high school participants and their schools. During this activity (which took two lab sessions), the girls learned about the HyperText Markup Language (HTML), and developed their own home pages, complete with tables, color backgrounds, pictures of themselves, and lots of links.

An *image processing* lab using XV gave students a chance to experiment with transforming pictures. Before the lab, a digital picture was taken of each student. In addition, many students brought photographs with them that we scanned in. These pictures were used in a series of exercises that taught the students about global transformations: rescaling, rotating, flipping, and cropping the image. Next, they experimented with color manipulation, brightening and sharpening the image, blurring, edge detection, and embossing. Finally, the students were free to experiment with transforming their own images.

In the *Growing Plants and Trees* activity, students learned about L-systems, grammatical systems to describe biological developments such as the growth of plants and cellular organisms. First, students learned how to design grammars representing L-systems and then how to use a graphical tool that would interpret their grammars in a visual manner, producing an animation of a shape growing. Next, they started with a simple grammar and, in a series of exercises, made slight modifications to the grammar, resulting in changing the direction of growth, changing color, changing the size and shape of lines, and branching.

4 NON-LABORATORY ACTIVITIES

Program participation also included non-laboratory activities, including attending talks by faculty, graduate and undergraduate students from Rensselaer and other universities, touring research labs at Rensselaer, and visiting nearby companies.

The majority of our speakers were women and were instructed to make their talks as interactive as possible, and to speak on a more personal level by describing who they were and how they got to their current position. The presentations varied considerably in topic and interaction. For example, computer ethics was presented as a discussion of case studies. Symbolic computation and distributed computing were presented by running examples on a computer. The history of computing talk included showing antique computer parts. The robotics talk included video of several robots, including the Navlab van in action. The talk on translating programming languages included analogies with the complexities of translating foreign languages.

The undergraduate participants presented their sum-

mer research work in 20 minute talks spread over three sessions. Their presentations included using a computer during the talk, many slides with pictures, and asking the audience questions. In all, talks had considerable visualization and interaction with students.

The lab visits included visiting several research labs at Rensselaer and two industry sites. At one of these sites, several women in a wide range of positions (researcher, lawyer, manager, trainer, and saleswomen) talked on a personal level about their jobs and the paths they had taken to get there.

5 EVALUATION

Students completed a questionnaire as part of the application process as well as two evaluation forms during the program, one at the end of each week. Our evaluations indicated our program was successful in challenging and interesting the girls in computer science. Of the overall activities, the hands-on labs were strongly preferred over the talks.

We are tracking the girls to see if any decide to major or minor in computer science when they go to college. One girl who started the camp planning on being an engineer told us that she was still planning on being an engineer but now she wanted to minor in computer science. Another girl stated that this program really helped her see what careers were available in computers. 82% of the girls want to come back again next year. Additional comments include: "liked the variety," "liked having the older kids [undergraduates] there", "it was nice to meet women working and enjoying computer work", and "I think the program should be three weeks."

Mostly our activities ran smoothly, but the computers were down for two of the labs. In the first case, we shifted the lab to the evening and let the girls have the afternoon off, and in the second case the computers came back up within 20 minutes. We also had a distraction problem similar to [10]. Our girls wanted to spend all their time using email and Netscape, even during labs. Right before each lab began, we had to ask the girls to remove email and Netscape windows. Because of our low ratio of one helper to every four students, most students with questions received help immediately. The majority of questions dealt with syntax problems or forgetting how to use a tool.

6 CONCLUSION

We have presented a number of computer science-related activities that we have used in a summer program for high school girls. These activities would be appropriate for use in similar programs designed to attract high school girls into the field of computer science. Using a large number of women as role models aids in showing girls that these fields are accessible to them.

Acknowledgement We thank all the faculty, graduate students, undergraduate students, industry personnel, and high school teachers who participated in our program and enlightened our high school students about areas in computer science.

References

- [1] O. Astrachan, *A Computer Science Tapestry: Exploring Programming and Computer Science with C++*, McGraw-Hill, 1995.
- [2] M. Cohen, M. Foster, D. Kratzer, P. Malone, and A. Solem, Get High School Students Hooked on Science with a Challenge. *Twenty-third SIGCSE Technical Symposium on Computer Science Education*, p. 240-245, 1992.
- [3] K. Frenkel, Women and Computing, *Communications of the ACM*, Vol. 33, No. 11, p. 34-46, November 1990.
- [4] D. Finkle, C. Hooker, S. Salvidio, M. Sullivan, and C. Thomas, Teaching C++ to High School Students, *Twenty-fifth SIGCSE Technical Symposium on Computer Science Education*, p. 286-289, 1994.
- [5] E. Martin, Solid Modeling with Scheme, *Twenty-sixth SIGCSE Technical Symposium on Computer Science Education*, p. 336-339, 1995.
- [6] J. O'Rourke, Mentor Project Targets Female Undergrads, *Computing Research News*, Vol. 5, No. 4, p. 3-5, September 1993.
- [7] A. Pearl, M. E. Pollack, E. Riskin, B. Thomas, E. Wolf and A. Wu, Becoming a Computer Scientist, *Communications of the ACM*, Vol. 33, No. 11, p. 47-57, November 1990.
- [8] S. H. Rodger and E. L. Walker, PipeLINK: Connecting Women Across the Computer Science Pipeline, *Computer Science Technical Report*, Rensselaer Polytechnic Institute, (to appear).
- [9] J. Stasko, Tango: A Framework and System for Algorithm Animation, *IEEE Computer*, p.27-39, September 1990.
- [10] G. J. Teague and V. Clarke, Attracting Women to Tertiary Computing Courses: Two Programs Directed at Secondary Level, *Twenty-fourth SIGCSE Technical Symposium on Computer Science Education*, p. 208-221, 1993.
- [11] D. Vastola and E. Walker, The Design and Implementation of an Interactive Learning Tool for Statistical Reasoning with Uncertainty, *Computers & Education: An International Journal*, 1996.