Robotics As A Component Of A General Education Course

Susan P. Imberman, Sarah Zelikovitz

College of Staten Island, City University of New York
imberman@mail.csi.cuny.edu
zelikovitz@mail.csi.cuny.edu

Abstract

We describe a robotics assignment for a General Education course. This assignment is part of an Introduction to Computer Technology course designed for the General Education requirements at The College of Staten Island. The goal for this assignment was to acquaint students with basic robotics while satisfying the course goals of algorithmic thinking, limitations of computers, problem solving, and debugging

INTRODUCTION

Non-major computer science is not a new concept. Most colleges and universities have some type of computer literacy course. These courses in the past have mostly concentrated on the use of computer applications such as word processing, spreadsheets, power point presentations, etc. Some of these courses are discipline specific. [Gabbert, P.] Others are more general.

Most authors agree that a computer science course that goes beyond literacy should include topics such as networking, basic operating system terminology, algorithmic method of problem solving, the internet, web development and advanced searching techniques. [Boyd, D. Gurwitz, C., Stegink, G.]. Most importantly, these courses should appeal to a wide audience and be engaging and fun.

The declining enrollment in the number of computer science majors has prompted many computer science faculty to look beyond basic literacy, creating a curriculum that can appeal to the liberal arts student. It is hoped that computer science courses that fit into the General Education requirements of a college's curriculum will not only allow students to experience the breadth of topics in computer science, but also encourage them to take more courses in the subject, possibly even majoring in the discipline. At the College of Staten Island we recently created such a course.

The General Education Computer Science Course

Our goals for our general education computer science course were manifold. Much discussion and thought went into the course content, keeping in mind the student population for which the course was intended. "What does a student need to be technologically educated?" was a major focus in designing the course. Essentially this is the same question posed by most disciplines having courses in the General Education core curricula. The course's major goal is to give a student a "well-rounded" technical education.

General Education at the college of Staten Island consists of two parts. Students of all majors must take two English courses, one History/U.S. government course, and a one credit physical education course. Aside from those specific courses, students must obtain credits in the following categories: Mathematics; Social Science; Scientific Analysis; Textual, Aesthetic and Linguistic Analysis; Pluralism and Diversity, and Contemporary World. The CSI college catalog enumerates all courses that belong in each of these categories. The list of courses that fulfill the "Scientific Analysis" requirement include courses in the natural sciences and courses in technology.

In the Fall 2007 semester the Computer Science Department at the College of Staten Island offered a new course, *Introduction to Computer Technology*, that fulfills the "Scientific Analysis" requirement for Liberal Arts Majors. This course is intended to introduce non-science students to fundamental concepts in computers and technology.

The course was presented to the CSI General Education Committee and other College governance bodies as an enhancement to the technology component of the "Scientific Analysis" category of General Education. It teaches students important analytical skills, and allows students to learn how and when to apply their knowledge of technology concepts productively. In particular, this course covers the concepts of: representing information, computer operations, networking, algorithmic thinking, limitations of computers, problem solving, debugging, privacy, security. The goal of the course is to have students understand the theory and science behind technology concepts. These concepts are taught with weekly or bi-weekly complementary lab exercises that allow students to apply newly learned methodological techniques by using current web-based technology, software and applications.

In particular, the labs in the course included: searching with, and comparison of, different search engines, diagnosis of hardware problems and BIOS settings analysis, use of software to gather network statistics and information, HTML and JavaScript programming, analysis of path algorithms using local maps and java applets, using techniques to evaluate security of computers and websites, evaluation and experimentation with speech recognition software, and robotics.

Since the course is in the "Scientific Analysis" category of General Education, all lab exercises needed to follow the scientific method. Many of the labs have a discovery phase, where students read articles, follow links, or try applets to learn about the problem that they are exploring. Many of the labs, too, include the creation of a hypothesis and the testing of that hypothesis. For example, the speech recognition lab explores word error rates with different passages, and different users. Students hypothesize which passage and which user would have the lowest error rate based upon their knowledge of speech recognition programs, and then test their results.

Students are required to write a report after the completion of each lab. The lab report components are: introduction and objective, description, observations, results, and references. This lab report forces students to think about the objective of each lab, and to review what they have done, why they have done it, and what they have learned from it. These reports help students focus on the important aspects of the labs as they are completing the labs.

Robotics has become an integral part of the computer science curriculum at The College of Staten Island (CSI). We have had much success integrating robotics into our CSI class and our Artificial Intelligence class. [Imberman, Imberman and Klibaner] Students at both the advanced and introductory levels found the robotics component engaging, interesting, and fun. Hence, it was only natural to include a robotics assignment into the new General Education course.

The NXT

We chose the new LEGO® Mindstorm NXT as the robotic platform for this course. The educational Mindstorm kit comes with all that is needed to construct a basic robot. For our robotic assignment, we needed a robot whose architecture was a minimum of two motors, the NXT brick, and a light sensor. Keeping in mind the need to engage the student, we decided to go beyond our basic needs and build a more complex robot. The Mindstorm kit comes with instructions on how to build such a robot. Our robot followed the LEGO® design. Figure 1 is a picture of a complete robot. As one can see from the figure, it is no wonder that students tended to anthropomorphize this robot. Its appearance is one that you would traditionally expect a robot to be, i.e. humanistic. It is definitely a much "cooler" and engaging architecture than two motors, and one sensor. Another reason for the enhanced architecture is our intention to use the platform in other classes. Thus, this architecture gives us much flexibility in this regard.

Another factor influencing our decision to use the NXT was the graphical programming language, written for this platform. Unlike text based programming languages, the NXT graphical language, based on Lab View, allows students to program complex robot behaviors with a few graphical blocks. This is done by modifying parameters associated with each block. The language allows for complicated programming constructs such as loops and decisions in a way that is easy for the nonprogrammer to grasp. Figure 2 shows an NXT graphical program that will move a robot in a box-like path.

Other materials needed for this project were white foam board, black masking tape, and green art paper. Materials were basically inexpensive. Each Mindstorm kit costs ~$250, and the art materials cost about $75. We used 15

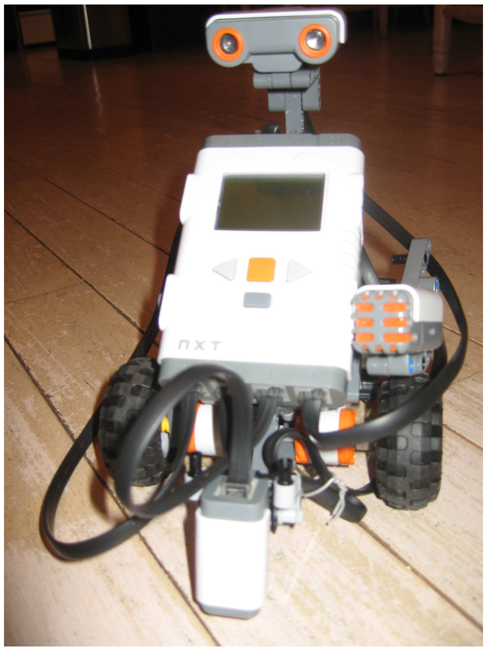

Figure 1 – NXT Robot

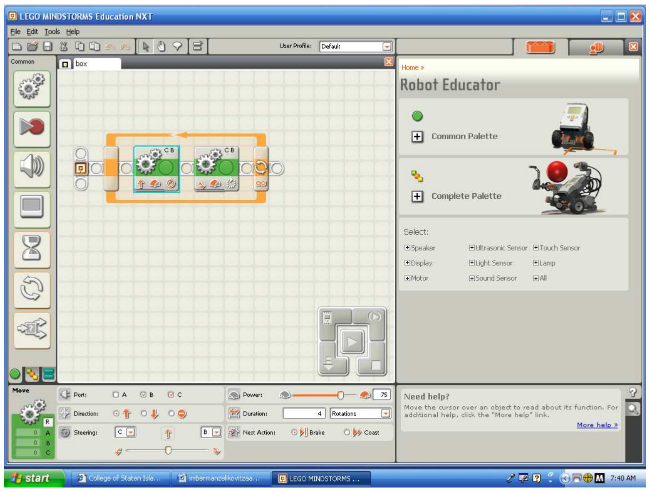

Figure 2 – Screen shot of NXT program

NXT kits, with two or three students sharing a robot. Therefore our total expense was less than $4,000.

The NXT Gen Ed Project

In developing this project we had several desired goals. First the lab was designed to address the course goals of algorithmic thinking, limitations of computers, problem solving, and debugging. Second, we wanted to create a hands-on exercise that would be challenging but not frustrating. Third, we also wanted to connect what the students did in the laboratory to robotics in the real world. Since the course was part of the Scientific Analysis group in the college's general education curriculum, projects had to follow the scientific method. Thus, fourth, our project had to allow for some experimentation. But the real challenge was accomplishing all of this during a two hour lab period!

At the lab session prior to the NXT project, a half hour was spent introducing the components of the robot, describing the hardware, input/output ports, and allowing the students to play with the robots. Students tried the sensors, and learned how to download a short program onto the NXT. By the time students entered lab to create their projects, the NXT robot was a bit familiar, and students understood some of the hardware components and some of the major steps of using the software.

Each project for this class included an introductory section which provided background information relevant to the project. The robotic lab's introduction gave a basic definition of the field of artificial intelligence and its relationship to robotics. The concept of what constitutes a robot is discussed with mention of famous robots of fact and fiction. The current state of robotic applications is explored with an emphasis on the Mars rovers. We emphasized this because the final task in the project was to program the NXT to "find life" on Mars.

The project starts with a one move block program. When a block is selected, as is the first block is in Figure 2, the screen displays several parameters. Students are encouraged to experiment with each of these, noting their observations as to the robot's behavior. Students then add more move blocks to get more interesting robot movements.

The concept of a loop is easy to implement using a loop block. The *Going Loopy* task instructs the students on the benefits of being able to repeat tasks in a computer program. The students place two **move** blocks inside a **loop** block and are instructed to experiment and adjust the **move** block parameters so the robot moves in a box pattern. This is the program displayed in Figure 2. It is easy to see graphically that the blocks inside the orange loop are repeated. We only use continuous loops in this lab and don't explore other loop concepts such as variable controlled loops. Robot programs are stopped by pressing the gray back button on the NXT.

The next task explores the light sensor. The concept of sensor calibration is described. Students are told to use the NXT's help files to learn how to calibrate the light sensor. When the NXT is connected to the computer, and a **light sensor** block is selected, the readings from the sensor are visible under the hour glass in the lower left of the parameter panel, Figure 3. Students were told to see what light reading they obtained by placing, black, white, and green paper under the sensor.

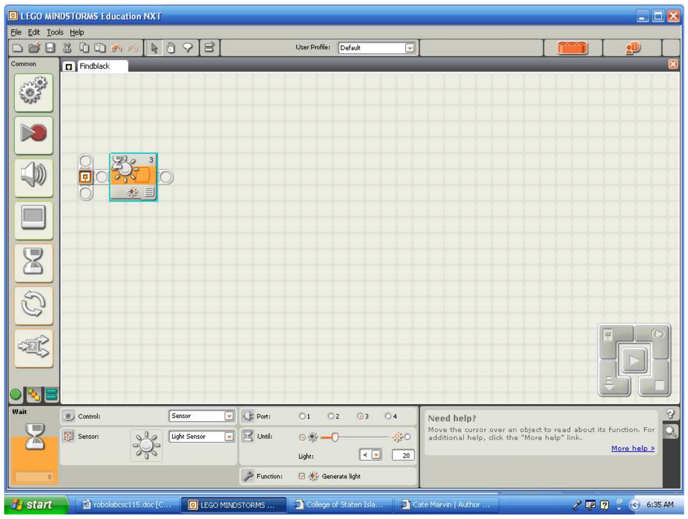

Figure 3 – Light sensor block

The ***until*** parameter sets a threshold value that the robot waits for, before continuing onto the next block in the program. Students are directed to adjust this parameter to detect black and to play a sound or phrase using the **sound** block, placing it onto the sequence beam after the sensor block. The task is expanded to require students to move the robot along a white foam board road with black stripes. Each time a black stripe is encountered the robot is to make some noise indicating it found black. Figure 4 depicts this striped road.

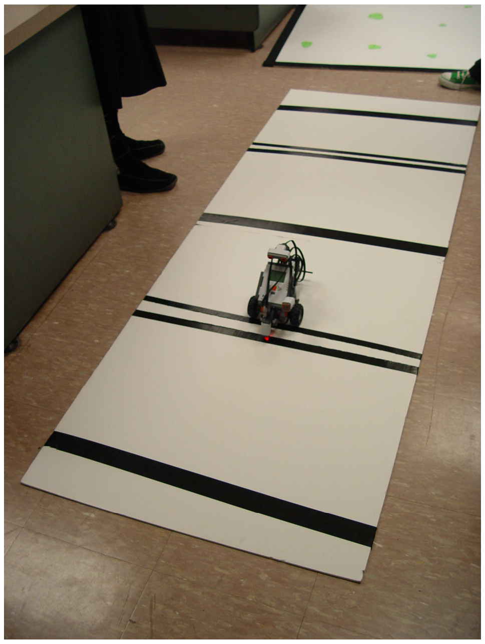

Figure 4 Black striped road

From here we create Mars rovers. It doesn't take much to modify the black stripe seeking program to one that explores Mars. The premise is that the student's Mars rover has landed on a plateau. The rover is to explore the plateau without falling off. The plateau is represented by a 48 inch by 76 inch white foam board surface, edged in black masking tape, demarcating the surrounding abyss. A video of a successful and unsuccessful robot can be viewed at: http://www.youtube.com/watch?v=Lo3oxF7du_Y

As extra credit, we describe how computer programs allow for decisions or branching, using the **decision** block. We demonstrate a program that can discriminate between black, green, and white. Using this, the Mars exploration program of the previous exercise is given purpose. Students are told to explore the plateau for Martian life (little green "aliens" glued to the white foam board background). Once found, the robot must send a signal to

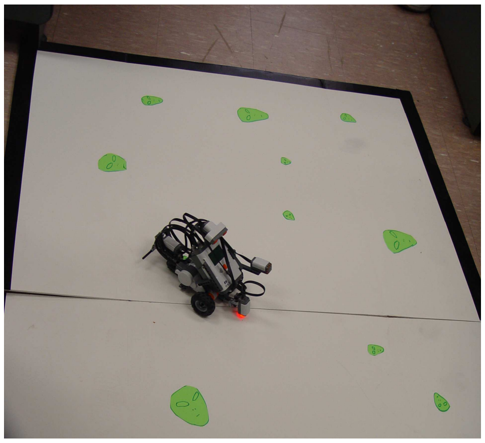

Figure 5 Our Martian Landscape

NASA via its communication device (the **sound** block). Figure 5 depicts our Martian landscape.

Lab directions are written in a step by step format, with no assumptions made with respect to the student's prior knowledge. Instructions are annotated with many screen shots and photographs. The complete project description can be found at: http://www.cs.csi.cuny.edu/~imberman/gened/Robotics.pdf

SUMMARY AND DISCUSSION

This paper describes a robotics project written for a course designed to satisfy a "Scientific Analysis" requirement in the General Education curriculum at The College of Staten Island. The course was offered for the first time in the Fall 2007 semester. We ran four sections of this course, each with an enrollment of 20 students.

Student reaction to the robots and to the NXT lab was overwhelmingly positive. The level of student participation and excitement in the two hour lab period was high. Students also found the lab to be quite challenging. At the end of the semester students were asked about the level of difficulty of each of the labs, and it seems that students found the robotics lab most difficult of all labs. Students spent the full two hour lab period on the project, and most did not have time to attempt the extra credit portion of the lab.

Faculty members who taught the course were asked why students found the lab difficult. The general feeling was that it was challenging for students to learn new software, new programming concepts, and new hardware all at once.

In the future, we plan to expand this project into a two week project that allows students more time to play with the robots, introduces more conditional components, and allows for more creativity.

References

Boyd, D.W., Lamprey, R. H. 2000 Computing as a General Education Requirement, *ACM Southeast Regional Conference,* Clemson, South Carolina

Gabbert, P, Discipline Focused Non-Major Computer Science Courses, 2004 ***Journal of Computing Sciences in Colleges**,* Volume 19 Issue 3

Gurwitz, C., The Internet as a Motivating Theme In a Math/Computer Core Course for Nonmajors, 1998 *ACM SIGCSE Bulletin , Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education SIGCSE '98*, Volume 30 Issue 1

Imberman, S., 2004 A Laboratory Exercise Using LEGO Handy Board Robots to Demonstrate Neural Networks in an Artificial Intelligence Class, *AAAI 2004 Spring Symposium Series Report*, Stanford, CA. March 22-24,.

Imberman, S., Klibaner, R., A Robotics Lab for CS1, 21st Annual Consortium for Computing Sciences in Colleges Eastern Conference, October 2005

Stegink, G., Pater, J., Vroon, D., 1999 Computer Science and General Education: Java Graphics and the Web, *ACM SIGCSE Bulletin , The proceedings of the thirtieth SIGCSE technical symposium on Computer science education SIGCSE '99*, Volume 31 Issue 1