A Robotics Introduction to Computer Science

Debra T. Burhans

Canisius College, Computer Science Department
2001 Main Street WTC 207, Buffalo, NY 14208
burhansd@canisius.edu

Abstract

This paper describes a new undergraduate course that serves two purposes. First, it satisfies a general education requirement in mathematical sciences, and second, it serves as one of four possible first courses for computer science majors. The course has no prerequisites: the student population is drawn primarily from college freshmen. This paper focuses on the curriculum, which blends topics from basic computing, artificial intelligence, and robotics. Results of a class survey are presented and discussed. Overall, satisfaction with both the course and the use of robots was high.

Introduction

This paper describes a new undergraduate course that serves two purposes. First, it satisfies a general education requirement in mathematical sciences, and second, it serves as one of four possible first courses for computer science (CS) majors. As such it is emblematic of our attempt to offer different course options for entry into the major to both attract new students to the major and to excite those already interested in computer science. Prior to the development of these new courses all CS majors started in a standard breadth-first course, which we also still offer. The other two new "flavors" of introductory CS are Introduction to Bioinformatics and Introduction to Web Programming.

The course has no prerequisites: the student population is drawn primarily from college freshmen. This paper describes the curriculum, which blends topics from basic computing, artificial intelligence, and robotics, and how the topics to be covered have been mapped into the robotics domain. Finally, the results of a class survey are presented and discussed. Overall results indicate high satisfaction, though as will be seen a significant number of students realize they don't like programming.

The robots employed in the course are Lego Mindstorms that use the RCX platform. We have an in-house developed algorithmic language that student use for programming that allows students to work in a much simpler environment than that of a traditional programming language.

Background

Robotics courses are proliferating due to a number of factors. Availability of low-cost robot platforms, faculty enthusiasm, and the need to level the playing field and to help attract new computer science majors are just some of these. Some robotics courses are for computer science majors and serve to introduce students to AI concepts in the context of robotics. Others are pure robotics courses.

A growing number of schools are using robots in their introductory courses. The majority of these efforts can be divided into two categories. (1) Increased use of robots in CS I and II to help students grasp fundamental concepts as well as to boost enthusiasm and retention. (2) Robotics courses for non-computer science majors that serve to convey ideas from computing and AI to a general audience through robotics.

CSCI 108 at Williams College (http://www.cs.williams.edu/~andrea/cs108/) is an example of the latter type of course. It provides an introduction to the field of AI through a variety of readings and activities including a robotics laboratory. Robots are built using the Handyboard platform and are programmed with Interactive C. There are no course prerequisites. The course is not part of a CS major, rather is offered to students outside the major to fulfill a general education requirement.

In the Fall 2006 semester George Washington University offered a robot-based section of its 8-week freshman course CS 001: Computer Science Orientation (http://www.seas.gwu.edu/%7Ebhagiweb/cs1/). This course is designed to introduce CS majors to the field and includes a lab-based project. The project involves teams of students working on maze navigation with Lego Mindstorms (RCX brick). This is a short introductory experience for majors.

Lego Mindstorms have been used with the Ada programming language to teach introductory computing to both CS majors and non-majors at the US Air Force Academy (Fagin). The stated goals of this course, along with an assessment of the ease of doing so with robotics, are:

> …to introduce students to basic computing ideas, including sequential control flow, selection, iteration, input/output, arrays, graphics, procedures, and file processing. Some of these concepts easily lend themselves to robotics; others are a better suited for a more conventional paradigm. Sequential control flow is the easiest.

As this paper points out, some concepts more naturally lend themselves to the robot paradigm while others appear to be less congruent.

There are numerous other references that describe the use of robotics in the computer science curriculum, including the Technical Report from the AAAI Spring Symposium on Accessible Hands-on Artificial Intelligence and Robotics Education (Greenwald et al.)

Course

As mentioned above, all of our computer science majors begin with a breadth-first computing course (CS 0) prior to their first course in Java programming (CS 1). This introductory semester is intended to help equip students with some of the skills that are needed to succeed in CS 1. Our students overall are weak in quantitative skills and tend to be math-phobic. A standard programming language is not introduced in this course because students are unprepared for the precision and rigor it entails. However, students gain the experience of programming with the accompanying debugging and compiling via an algorithmic simplified language (Robolang) that they use to write program which are then compiled and downloaded to their robots.

This course carries only 3 standard semester credit hours as opposed to CS 1 and most of our other CS courses, which carry 4 credits. Students have 4 contact hours a week in the course including a 50-minute lab meeting, but expectations are considerably lower than they would be for a 4 credit offering, in particular with regard to lab/programming work.

Curriculum

In order to develop not only this new robotics course but other specialized version of courses that introduce computer science we first identified key topics currently covered in CS 0. These include: problem solving, algorithms, history, ethical and social issues, databases, computer system components, networking, WWW, software/OS, binary representation, basic logic, limitations of computing, AI, and simulation. These topics correspond approximately to those presented in most CS 0 texts, some topics underlie the rest (problem solving and algorithms) while others receive a cursory overview (AI).
From this list we identified core topics that all of our specialized introductory courses would cover: problem solving, algorithms, history (of whatever topic is the course focus), binary representation, basic logic, and social and ethical issues (again, focused on the topic area). In order to reinforce the breadth and unifying CS concepts in the robotics course a CS 0 book was required. There was no suitable robotics book as we do not use a traditional programming language for the robotics; rather, we are using our own algorithmic language. In addition, there was not enough focus on robot construction to warrant the adoption of a book focused on building robots.

Hands on from the start

We elected to have each student build his or her own robot, without any expectation as to whether this would be good or bad. Once students started programming the robots they worked in teams, developing programs in groups of 2 or 3 yet downloading the programs onto each individual robot. As more complex tasks were posed, including preparing for a competition, teams selected one robot to use for task completion yet continued (purely for fun since this was outside the required projects for the lab) to alter the other robots belonging to the team. While building robots was frustrating there was clearly value in getting one completed and getting it to do things.

History

Material for the history of robotics unit (a combination of history of CS and history of AI with more material from the recent past on robotics) was assembled by the instructor based on a number of Internet and book resources. It was difficult to assimilate the vast amount of material and some traditional history of CS material was excluded. Some historical events such as the construction of mechanical animals and people could be seen to foreshadow the development of androids and piqued the interest of students. Material dropped from a traditional history of CS includes an in depth look at different generations of computers.

Problem Solving and Algorithms

Problem solving and algorithms are the foundation of the course. Skills in these areas are developed in both the laboratory and classroom through hands on and programming exercises. The programming language used is Robolang, which is associated with Robotran, a program that translates Robolang programs into Lejos source code. (http://www-cs.canisius.edu/~rmmeyer/ROBOTRAN/home.html).
All of the software for the course has been developed in our department. Robolang is a simple algorithmic language that is easy for students to understand. It provides a rich set of primitives for problem solving and is designed specifically for use with the Lego RCX.

The following are a couple representative Robotran programs. The first is a simple loop to (endlessly) move the robot forward, albeit slowly (three seconds forward, two seconds back).

```
program prog4
     while true
        go forward 3 seconds
        stop
        go backward 2 seconds
     end
```

The second program tests the value of a touch sensor connected to S1. If the sensor is triggered the robot backs up and turns right a random amount. The random function in Robolang by default returns a value in the range 0-99. The units for turns may be specified, here degrees are used.

```
program prog14
     S1 is a touch sensor
     go forward
     while true
        if S1 == 1 then
          stop
          go backward 1 second
          var angle = random
          turn sharp right angle degrees
          go forward
        end
     end
```

Methodology: Students are first presented with a task for the robot to complete. They work in teams to come up with a strategy for completing the task. The strategy is then formalized using Robolang. Robolang programs are translated into Lejos, compiled, and downloaded onto the robot and run.

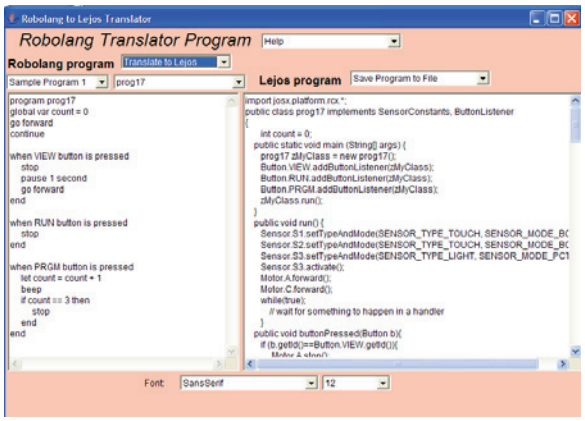

Figure 1. Screen shot of Robotran GUI showing Robolang on the left and Lejos code on the right.

The first tasks we assign involve having the robot draw letters. The first robots the students build (called "penbots") can raise and lower a pen using a third motor (the other two are used for movement). They draw on large pads of paper. Other tasks (without the pen attachment) include line following, staying within a ring, and maze navigation. We have found that letter drawing is an ideal initial problem to work on: the results are immediate and visual.

Binary Representation and Logic

Currently logic is covered in lecture with hands on exercises and through the use of Logisim in the lab (Burch). This is one area where robotics is not directly used to teach a concept. It should be noted that students very much enjoy using Logisim and mastery of the topic is extremely high among students, so this is a successful course component as it stands.

The idea of different representations of information and translation among representations is also reinforced in the lab through the use of Robotran. In particular, the GUI for Robotran displays both the Robolang algorithm and the corresponding Lejos source code (Figure 1). The value in this approach is that students understand that what actually is "understood" by the computer, in this case the RCX, is not the same as the language they use to think and problem solve in. Students must compile and download their programs, reinforcing the idea of a hierarchy of formats and translation among them.

AI and Robotics

Concepts of AI are introduced in a number of ways, including literature and the popular press. Students take turns presenting "Robots in the News" where they have to find something in the news that focuses on robotics. These stories have ranged from nursing home robots in Japan to the Scooba. We read Asimov's *I Robot* and Lem's *Cyberiad* and watched the I Robot film as well as an episode of the cartoon series "Futurama" (Obsoletely Fabulous). Once students gain hands on experience with robots and learn about the types of robots that actually exist in the world, it gives them a better and more critical understanding of the way robots are portrayed by writers and artists.

Ethical and Social Issues

The Futurama episode touches on issues that include human aggression vs. "killer robots", the lack of souls in robots, and Luddism. This, coupled with a videotaped lecture by philosopher Stephen Petersen on "The Ethics of Robot Servitude", was used to introduce ethical and societal issues, including three different approaches to considering robot "personhood" (or lack thereof) that are broadly applicable to any social issues: utilitarianism, deontology, and naturalism.

Logistics

The course was scheduled to meet Tuesday/Thursday, making lecture periods 1.25 hrs in length. The laboratory was scheduled for Thursday immediately following the lecture for an additional hour. This allowed us the flexibility to spend varying amounts of time up to 2.25 hours in the lab on Thursday which has proved invaluable.

Competition
A competition proved to be the highlight of the course for students. They spent approximately a month preparing for a 3-part competition that included a drag race, maze, and sumo wrestling contest. While initially concerned and reluctant due to the other competitors, who were drawn from an advanced physics lab (junior and senior physics majors), an introductory engineering course, and a team of upper-level CS majors, the students placed fourth overall out of 12 teams and won the sumo challenge!

Results

The 11 students included 4 CS majors, 1 CS minor, 2 math education students, an art student, and the rest undeclared or other liberal arts. Interestingly, the students who took programming courses in high school proved the slowest at completing tasks in the lab. It was difficult for them to work with a simple algorithmic language even though they clearly have not grasped many programming concepts. They tend to be poor at following directions, which is an essential component for success in the course.

The following table summarizes class survey results.

Question	SA	A	N	D	SD
Building the robots was worthwhile	91	9	0	0	0
The robots helped me understand programming	82	18	0	0	0
The robots were more interesting than just pure programming	73	18	9	0	0
The course stimulated my interest in robotics	64	36	0	0	0
I am interested in doing something with robots in the future based on my experiences in this class	18	46	27	9	0
This class made me more interested in computer science	18	36	45	0	0
The class made me realize that I don't like programming	9	9	18	27	36
The robots helped me understand algorithms	36	36	9	18	0
I would rather have just programming the robots and not built them	10	0	18	18	55
This class stimulated my interest in AI	27	55	18	0	0
I would like to more with robots in future CS courses	45	18	36	0	0

Table 1. Survey Results CSC 109 Fall 2006 (SA=strongly agree; A= agree; N=neutral; D=disagree; SD=strongly disagree) presented as percentage of students. Note that percentages may not add up to 100 due to truncation of decimal values < 0.5

The table shows a number of interesting results, though the class size was small (11 students). Students clearly liked the experience of actually building the robots. A majority (72%) felt that the robots helped them understand algorithms, which is the most important course goal.

The class sparked a significant interest on the part of students in robotics (100%) and AI (82%) but not as much in computer science (54%). Over a third of the students discovered they didn't like to program.

The four intended CS majors have stayed (thus far) in the major along with the intended CS minor. While this first offering of the course did not attract any new majors it seemed to give students a good idea of what computing was "about" in a way that allowed them to engage with newspaper stories, literature and film. This unexpected result shows that the course, may serve the students and our field well by connecting basic ideas of computing to daily interactions with our digital world.

Future Work

There are a number of aspects of the course that need tuning or further development. For example, if Robotran could be expanded so that it could show translated forms of the Lejos programs, including assembly language and byte code, it could provide a seamless and compelling way for students to understand binary representation. A complete system allowing for assembly programming would be ideal. It is not, however, clear that this is required since students are happy with Logisim.

Robot resources are scarce and we have developed a simulator for the RCX to help allay this problem. Unfortunately we have not yet effectively integrated it into the course, in part due to the complexities of creating software that a typical techno-phobic student can install and use.

References

Burch, Carl, *Logisim: A graphical system for logic circuit design and simulation,* Journal of Educational Resources in Computing *2*:1, p. 5-16, 2002.

Fagin, Barry, *Using Ada-based robotics to teach computer science*, ACM SIGCSE Bulletin, 32:3, p.148-151, Sept. 2000.

Greenwald, Lloyd, Dodds, Zachary, Howard, Ayanna, Tejada, Sheila and Weinberg, Jerry (editors), *Accessible Hands-on Artificial Intelligence and Robotics Education: Papers from 2004 AAAI Spring Symposium TR SS-04-01,* AAAI Press, 2004.