Using Robotics to Motivate Learning in an AI Course for Non-Majors

Andrea Pohoreckyj Danyluk
Williams College
Department of Computer Science
47 Lab Campus Drive
Williamstown, MA 01267
andrea@cs.williams.edu

Abstract

This paper describes an undergraduate course in Artificial Intelligence aimed at students who are not Computer Science majors. A significant component of the course is a weekly laboratory in which students build and program simple vehicular robots. While the laboratory exercises are quite distinct from the topics discussed in class, they serve several important purposes. The lab provides a setting in which students can build confidence; the fun of working with robots motivates students to be more engaged in lecture; and students learn some fundamental programming concepts in addition to material about AI.

Introduction

This paper describes an undergraduate course in Artificial Intelligence aimed at students who are not Computer Science majors. A significant component of the course is a weekly laboratory in which students build and program simple vehicular robots. While the laboratory exercises are quite different from the topics discussed in class, they serve several important purposes. They help to give some students, particularly those who are afraid of science, confidence in their ability to handle a science course. They give students enough of a taste of programming that they can determine whether they would like to take additional courses in Computer Science. They motivate students to come to class to learn about topics that, while interesting, are less "flashy" than robotics. They give students some sense of the difficulties inherent in the complex and exciting discipline of Artificial Intelligence.

We begin with a discussion of the objectives for this course, giving a brief history behind its creation. We then describe the course curriculum, format, and assignments, providing details about the set-up of the laboratory. We discuss the benefits of our course design. We then conclude with open issues and ideas for future offerings of this course.

Objectives for a Non-Major Course in CS

The standard introduction to our discipline is a programming course. While CS1 courses aim to teach students about the fundamental principles behind the design of algorithms and data structures, they are more typically nuts and bolts courses that teach programming in a given language of choice. Students typically do not gain an appreciation for the ideas and questions that define the discipline of Computer Science.

We can argue that learning to program is useful for Computer Science majors as well as for students who expect to apply programming skills to other disciplines such as Physics, Engineering, and Economics. However, it is less clear that it is the right type of course for the average student who would simply like to learn more about Computer Science.

This course is one of three non-major Computer Science courses offered at Williams College. Each of these is aimed at giving students an introduction to a specific topic area: networks (Murtagh 2001), graphics, and artificial intelligence. A core set of goals for these courses is as follows:

- they should introduce students to the fundamental questions and issues of some aspect of Computer Science;
- they should teach students through problem solving and programming – these are not intended to be "gut" courses.

To a large extent, the three courses also share the following objectives of the AI course for non-majors:

- to motivate students to become Computer Science majors;
- to give students enough practice with programming and problem solving that they can determine whether they have the aptitude to go on in the major;
- to make all students feel comfortable in the course, regardless of their background in science.

In essence, the goal is to inspire students to become excited about our discipline by giving them a sense of what it is all about, rather than introducing them only to programming. The course needs to be sufficiently rigorous that students can appropriately determine whether continuing is right for them. But it needs to be a sufficiently friendly environment that even science-phobic students can comfortably complete the course.

The Course Structure

The following sections describe the structure of our course, including lecture topics and lab assignments. Detailed course information, such as the syllabus, lecture notes, and lab write-ups can be found at:

http://www.cs.williams.edu/~andrea/cs108

Many of the ideas for this course were taken from a similar course offered at Swarthmore (Meeden ; Kumar and Meeden 1998).

Course Format

The course is taught over twelve weeks. It meets for three fifty-minute lecture sessions each week. There is also a weekly lab session. In some semesters the lab has met for ninety minutes and in others it has met for three hours.

While class sessions are primarily lectures, approximately six sessions each semester are devoted to discussion of readings. Typically, there are three sessions in the first half of the semester and three at the end of the course.

The same lab assignments have been given regardless of the length of the scheduled lab. Students are typically able to complete a lab exercise in the scheduled time when they are given three hours. With only ninety minutes, students return during evening TA hours to complete their labs. Completing a lab exercise generally involves demonstrating a particular behavior with a robot, submitting the program that implements the behavior, and submitting a brief lab report in which the student discusses various ideas for their implementation as well as successes and stumbling blocks.

In addition to lab exercises, students do readings for each class. The readings include textbook readings, position papers on various aspects of AI, philosophy, and psychology, as well as fiction. During discussion weeks, students submit a brief response to the readings for the week. Students also complete two problem sets.

Students work in groups of three on the lab exercises, though their lab reports, reading responses, and problem sets are to be done independently.

Robotics Platform

In this course students build simple vehicular robots around the Handy Board (Martin) robot controller. Our lab has ten Handy Boards – one for the instructor, six to seven for groups of three students each, and two to three to be swapped for student boards, when necessary.

Each group's workstation consists of a Macintosh computer, Interactive C software, a Handy Board, and a variety of building materials including LEGOs, motors, and touch and light sensors. Each student group is also provided with a toolkit that includes all the materials necessary to appropriately wire the motors and sensors for the Handy Board, such as wire, pins, soldering iron, wire strippers, pliers, and so on.

Kumar and Meeden have written a resource guide for creating this type of robot laboratory (Kumar and Meeden). The guide also contains many ideas for laboratory exercises.

The Classroom Curriculum

We spend roughly half the semester on topics specifically related to robotics and the programming that students will be doing in lab. The second half of the semester is devoted to more general topics within AI.

Topics covered in the first half of the semester include:

- possible applications of robotics,
- challenges for robotics (such as unpredictability of actions in the real world)
- types of effectors for both locomotion and manipulation
- types of sensors
- classical architectures and STRIPS-style planning
- behavioral approaches
- navigation and motion planning
- configuration spaces

We also devote six lectures to programming in Interactive C. We cover this material on an as-needed basis, providing just enough information in class for students to be able to complete their lab exercises.

Topics covered in the second half of the course include units on knowledge representation and reasoning, search, game playing, and machine learning. We also discuss topics such as robots in literature and film, predictions about the future of robotics and AI, and objections to AI.

Laboratory Exercises

Laboratory exercises for this course serve several purposes, as indicated earlier. We describe the exercises here and discuss the merits of these exercises in the next section.

The students' first lab exercise is to solder and wire the motors and sensors that they will be using for the remainder of the semester. While we could provide pre-wired motors and sensors, we have found this lab to be very effective in making students feel comfortable with the hardware they will be using.

In the second lab, students test the motors and sensors they wired the previous week. They learn how to download Interactive C to the Handy Board. They also learn how to use simple commands to turn motors on and off, and to get sensor readings.

In the third lab, the students are asked to build the actual vehicle around the Handy Board. We provide them with a standard design, but students are also free to experiment with other designs. Once their vehicle is built, students download a test program to the vehicle and run it.

At this point students are ready to begin weekly implementations of a variety of behaviors. These include:

- light seeking and avoidance – In this lab, students implement two behaviors described in (Braitenberg 1984). These are fear and agression. Two light sensors are placed on the front of the vehicle. In both cases, the intensity of the light sensed controls the speed of the vehicle. More intense light makes the motors speed up; less intense light makes them slow down. In one case, the right sensor is linked to the right motor, and the left sensor is linked to the left motor. In the second case, the right sensor controls the left motor and vice versa. The behaviors of the vehicle in each of these cases will appear to be fearful and aggressive, respectively.
- bumper sensing and obstacle avoidance – In this lab, students implement a simple wandering behavior. If their

robot bumps into something while wandering, it should back away, adjust its direction away from the obstacle, and continue to wander. Students place two touch sensors on the front of their vehicle. It is also suggested that they place at least one touch sensor on the back of their vehicle.

- combined light seeking with bumper sensing – For this lab, students use the touch sensors from the previous week and at least three light sensors, arranged in any way they would like. Students are asked to implement light seeking behavior, similar to the aggression behavior in an earlier lab. However, they should assume the presence of obstacles in this case. As a result, while seeking light, their vehicle must also be able to adjust when bumping into objects.
- a simulation of feeding behavior, with light as a food source – This lab is a further extension of previous exercises. In this lab, students are to think of light as a food source. As a result, their robot is to seek out light to nourish itself. To feed, the robot finds a light source and stops near that source for a specified period of time. After feeding, it wanders without seeking food, as it is "full." Once it becomes "hungry" (again, after a specified period of time), it must find a light source fairly quickly, otherwise its energy (as measured by maximum motor speed) decreases.
- navigation of a simple maze – This is a three-week assignment. In the first week, students implement a number of functions that they can use to deliberately control their robots. These include functions to move a specific number of inches, to turn ninety degrees to the right or left, and so on. In the second week, students are shown a maze. They write two programs to navigate the maze – one based upon the functions they wrote a week earlier and a second that uses touch sensors to "feel out" the maze. In the third week, the robots compete to see which can successfully (and most quickly) navigate the maze.

In the first few weeks, the lab exercises are quite closely tied to the lectures. For example, while the students are working on wiring motors and sensors in lab, they are also learning about different types of effectors and sensors in lecture. When testing their motors and sensors, they are learning about programming in Interactive C. When they program their first simple behaviors, we are discussing behavioral approaches in class and are comparing them to classical planning. At this point, the lectures and labs diverge.

Interestingly, students do not have trouble with the lab topics being different from lecture topics in the second half of the semester. There are a few possible explanations for this. One is that the course still has a unified theme. Another is that the labs are simply so enjoyable, that students don't find them uncomfortable. While we have not done a systematic study to address this specifically, end-of-course surveys provide evidence for this. Many students comment that the labs are great fun and their favorite part of the course.

It is also useful to note that there is more than a simple divergence in topic after the first few weeks. The robot exercises don't particularly build on any specific AI themes. We view them as exercises for practicing fundamental programming skills. The robots are linked to the course topic in the students' minds, but they are primarily a tool for teaching and developing basic skills in an interesting way.

Benefits of Robotics Exercises

As indicated in the Introduction, the robot labs serve several important purposes. Our confidence in the following points is based on a combination of anecdotal evidence and the course surveys that students fill out at the end of each semester. We can compare the survey results of this version of the course with a previous version. In the earlier course, the emphasis was entirely on classical topics within AI, such as knowledge representation, logic, search, and planning. There was no robotics component. The weekly lab involved writing very simple programs (or pieces of larger programs) in Lisp.

First, the earliest robotics labs help to *build confidence*. Students typically remark after the soldering lab that it was the best lab they have ever done. While soldering is a fairly straightforward and simple activity, many of our students have never done it. As it is a component of robot building, the students perceive it as important. Once they have become familiar with the tools and have completed the task, they leave with a sense of confidence in their ability to handle something as complicated as a hardware lab. This sense is supported by Computer Science majors. Those who are around the lab tend to remark that they've never had the opportunity to do anything like that (and that they would like to).

Second, students get enough of a taste of programming that they can determine whether they would like to go on in Computer Science. The programming constructs needed to complete the labs include constant and variable declaration, function definition, conditionals, and loops. Students also learn about writing processes to handle specific types of control and learn about the use of shared variables for communication among processes. In the last offering of the course, there were eighteen students enrolled. Half were juniors and seniors and had already declared their majors. Of the remaining nine, two have gone on to become Computer Science majors.

While students do some fairly sophisticated programming by the end of the semester, they are not overwhelmed by programming. Because they work in groups, students are never left to sink or swim on their own. There are certainly situations in which a strong student can take over a group, but this has been rare in our experience. On the contrary, students seem more willing to try their own ideas when they can do so with the support of their peers.

Third, the robot lab exercises motivate students to come to class to learn about topics that, while interesting, are less "flashy" than robotics. It is important to keep in mind that students taking this particular course are not those who intend to go on in Computer Science. They are there either to fulfill a science requirement or to learn about something they perceive as "cool." In the early non-robotics version of the course, students grudgingly admitted that they had learned valuable things about what AI really is (as opposed to their notion from science fiction movies and books), but they wished that there had been something more like what they read about. The new version captures the interest and enthusiasm of these students and keeps it. This enthusiasm translates to the lectures as well as the lab.

Finally, because they are working with robots in the physical world, students gain an appreciation for some of the practical difficulties of robotics.

New Directions and Future Work

The current version of the course was first offered in the spring of 1999. It is time to update and revise the course. One reason for doing so is simply that there have been many innovations in the field of robotics since that time. There are other compelling reasons to modify the course, however.

In 2001, the faculty of Williams College voted to implement a new requirement for all students. Beginning with the class of 2006, all students are required to take one course that is designated as a course that builds skills in quantitative and formal reasoning. This course does so in its current version, but we feel that we can make it even more rigorous. The faculty vote to implement this requirement has reinforced our belief in the importance of teaching these skills to our students.

At this point, we plan to assign more problem sets and anticipate cutting some of the discussion from the course. We are also planning to introduce new laboratory exercises.

References

Braitenberg, V. 1984. *Vehicles*. Cambridge, MA: MIT Press.

Kumar, D., and Meeden, L. A robot laboratory for teaching artificial intelligence resource kit. World Wide Web, URL is http://mainline.brynmawr.edu/Robots/ResourceKit/.

Kumar, D., and Meeden, L. 1998. A robot laboratory for teaching artificial intelligence. In *Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education (SIGCSE-98)*.

Martin, F. The handy board. World Wide Web, URL is http://lcs.www.media.mit.edu/groups/el/Projects/handy-board/.

Meeden, L. World Wide Web, URL is http://www.cs.swarthmore.edu/~meeden/.

Murtagh, T. P. 2001. Teaching breadth-first depth-first. In *ITiCSE 2001, Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*.