

Using RoboCup in University-Level Computer Science Education

ELIZABETH SKLAR AND SIMON PARSONS

Brooklyn College, City University of New York

AND

PETER STONE

University of Texas at Austin

In the education literature, team-based projects have proven to be an effective pedagogical methodology. We have been using *RoboCup* challenges as the basis for class projects in undergraduate and masters level courses. This article discusses several independent efforts in this direction and presents our work in the development of shared resources and evaluation instruments. We outline three courses and describe related class projects in order to make the context of our investigation clear and make it possible for others to replicate and extend our work as well as contribute to the shared resource.

Categories and Subject Descriptors: K.3.2 [Computers and Education]: Computer and Information Science Education - *Computer science education, Curriculum*; I.2.9 [Artificial Intelligence]: Robotics - *Autonomous vehicles, Sensors*; I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence - *Intelligent agents, Multiagent systems*

General Terms: Documentation, Measurement

Additional Key Words and Phrases: Educational robotics, RoboCup.

1. INTRODUCTION

The notion of *educational robotics* – the use of robotics as a hands-on learning environment – is becoming increasingly common, with the rise in popularity of low-cost and accessible robot kits [Sklar and Parsons 2002]. Creative instructors are finding ways to teach a myriad of science topics using these hands-on technologies and engaging students of all ages. Tournaments are being organized using the robots, and the energy, enthusiasm, and motivation displayed by students is unsurpassed. We have found the RoboCup challenges – especially Soccer Simulation and RoboCupJunior Soccer and Rescue – to be particularly conducive to college-level classroom use. We believe that the ability to demonstrate theoretical models and complex algorithms with a hands-on, accessible medium strengthens the learning experience for students.

RoboCup [RoboCup 2005], initiated in 1997, was designed to bring together robotics and artificial intelligence researchers world-wide by providing a common problem for which a solution would require both advances in many fields and a collective approach to research in those fields [Kitano et al. 1997]. The chosen arena was robotic soccer, currently played by autonomous robots in several categories (“leagues”) distinguished by

Authors' addresses: E. Sklar and S. Parsons, Department of Computer and Information Science, Brooklyn College, CUNY, Brooklyn, NY 11210; email: sklar@sci.brooklyn.cuny.edu; parsons@sci.brooklyn.cuny.edu; P. Stone, Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712-1188; email: pstone@cs.utexas.edu.

Permission to make digital/hard copy of part of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date of appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Permission may be requested from the Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036, USA, fax:+1(212) 869-0481, permissions@acm.org

© 2005 ACM 1531-4278/04/0600-ART1 \$5.00

differences in physical size, cost, type of hardware platform, and approaches to vision and software control. This was later expanded to include robotic urban search and rescue as well. In 2000, the RoboCupJunior [RoboCupJunior 2005] division was formed, with the goal of introducing young students (primary through high school) to RoboCup and providing them with an exciting and motivating way to learn about technology through hands-on experiences [Sklar et al. 2002, Sklar and Eguchi 2004].

In this article, we document our experiences incorporating RoboCup activities into advanced undergraduate and early graduate courses, with the goal of uniting with others who are doing the same. Our long-term aims are two-fold: one, to create a repository for related curricular materials; and two, to build a common instrument and database for evaluating the RoboCup learning environment. Another motivation is to provide a means for enabling others to replicate and expand our efforts. While we have found the link between RoboCup and traditional coursework in introductory robotics, artificial intelligence (AI), and autonomous agents and multiagent systems (AAMAS) to be a natural one, we presume that this sense arises out of our familiarity with RoboCup through long-term involvement with the initiative. In developing our repository, we are hoping to make the notion of incorporating RoboCup into such coursework a relatively easy task for uninitiated instructors, by providing syllabi, reading lists, and project descriptions.

We are not the first to experiment with the RoboCup paradigm in undergraduate classrooms. Silvia Coradeschi and Jacek Malec presented results of their early experiences at RoboCup in 1998 [Coradeschi and Malec 1998]. They used the RoboCup soccer simulation in a course on artificial intelligence programming [Coradeschi 2000]. Their course involved some lecture and seminar time, but was primarily devoted to lab work. The students were given reading materials on the RoboCup initiative and were also expected to write two reports during the course. Their project involved programming an agent to perform in the RoboCup soccer simulator. The instructors' evaluation of the effort was mixed; they found that students spent perhaps too much time on understanding issues such as real-time process and socket programming – topics outside the normal AI curriculum. However, the students were surveyed, and agreed that the idea of integrating RoboCup in the course was a good one. Subsequently, the RoboSoc library was developed to ease students into using agents for the RoboCup soccer simulator [Heintz 2000]. Andreas Birk developed a course on autonomous systems that uses the Small-Size RoboCup League for practical exercises [Birk 2001]. José Vidal and Paul Buhler [Vidal and Buhler 2002] developed a series of graduate-level courses on multiagent systems using the RoboCup simulation league. They created a platform called “Biter” which provides a framework for agent development and simplifies the task of building RoboCup agents for educational purposes.

We are also not the first to use robot kits in an undergraduate classroom. In 1989, Fred Martin created the MIT robot design project course¹, following from Woodie Flowers' renowned introduction to design course², which was offered in the mechanical engineering department [Martin 1989]. In Martin's course, students learn about the basics of building robots from kits and the term culminates in a contest; the work in this course resulted in a textbook. [Martin 2000]. Holly Yanco [Yanco 2002] adopted this course using the botball [Botball 2005] game as the tournament at the end of the term. Maja Mataric' developed an award-winning course called “Introduction to Robotics” [Mataric'

¹ course number 6.270

² course number 2.70

1998], which takes a hands-on approach to teaching fundamental robotics concepts. Students use both the Handy board microcontroller [Handy board 2005] and the LEGO Mindstorms invention system [Mindstorms 2005]. The syllabus covers in detail the basic components of robotics from a technical standpoint, and the course ends with a contest where robots play a ball game in a hexagonal field.

A few people have developed courses using hands-on robotics that do not focus on teaching robotic topics as the main subject. Michael Littman's course, called "Programming Under Uncertainty" [Littman 1999], offered in Fall 1999, taught a variety of methods for programming under uncertainty, including Markov decision processes and POMDPs, and a variety of machine-learning techniques like reinforcement learning and genetic algorithms; students used LEGO robots to demonstrate their knowledge. The course ended with a project where some of the students developed their own applications for their robots, ranging in sophistication from line-following tasks to making breakfast.

Aside from constructing a shared repository of course materials, we are interested in creating an assessment instrument and in conducting a comprehensive evaluation of the pedagogical value of educational robotics in general and RoboCup activities in particular. While we have empirically witnessed increased excitement, interest, and motivation on the part of our students, we are working to formalize these observations with a scientific study of the RoboCup learning environment. Following on the work of Sklar and colleagues [Sklar et al. 2002; Sklar and Eguchi 2004], which examined this phenomenon at the primary and secondary school-age levels, we are interested in trying to pinpoint the educational value of robotics and the RoboCup initiative at the undergraduate level.

Uniting multiple instructors from disparate universities means that we can not only share experiences, but also collect course evaluation data on a grander scale, thereby contributing jointly to a larger database of student experiences with robotics and RoboCup as a learning arena. This database will allow us to perform analysis across a broader cohort, with a range of academic as well as cultural backgrounds.

This article is organized as follows. We describe three courses, offered by each of the authors at different universities in the USA over four academic years. We outline the syllabus of each course, and share reading lists and project descriptions. We also present results of course evaluations of each class and then analyze the parallels between the coursework and evaluation results. We conclude with discussion of current and future work.

2. COURSE: INTRODUCTION TO ROBOTICS

This introductory course looks at robotics from several aspects, i.e., technically, historically, and socially (many of its technical aspects are based on the work of Mataric' [1998]). The course was designed for non-engineering students majoring in computer science; it gives hands-on experiences with technology and provides basic material on the field of robotics and the challenges facing the field today. Prerequisites for the course are at least an introductory programming course in C, although it is better if students have more programming experience and some background in data structures. Part of the course is spent reading and discussing classic material that relates to robots, including non-technical literature, e.g., science fiction, psychology, cognitive science, and edutainment. The remainder of the course takes a hands-on approach to introducing the basic concepts in robotics, focusing on autonomous mobile robots. LEGO Mindstorms robots are used, and students must complete two projects with them. First, they must build robots to execute a line-following task, culminating in a maze contest (based on the

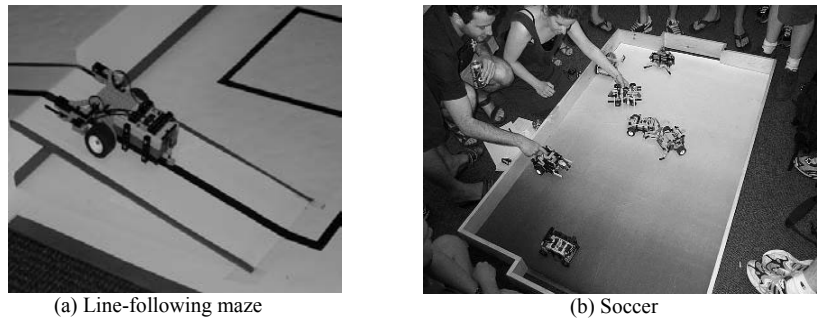


Fig. 1. Robot contests.

RoboCupJunior rescue task); see Figure 1(a). Second, they construct robots to play soccer and perform in a RoboCupJunior style two-on-two tournament (Figure 1(b)).

4.1 Course Structure

The course begins with an introduction to robotics and agent-based artificial intelligence [Russell and Norvig 1995]. Then some history is covered, followed by the basics of building and programming with LEGO Mindstorms [Martin 1996; Martin et al. 2000], using Not Quite C [Baum 2000; Baum et al. 2000; NQC]. The LEGO robots are used as examples for the remaining course topics, which introduce the general areas in robotics: effectors, sensors, and control [McKerrow 1991; Martin 2000]. The area of control is covered in more depth, discussing various architectures including deliberative, reactive, hybrid, and behavior-based [Brooks 1986; Arkin 1998; Birk 1998; Mataric' 1997]. Learning is also discussed [Mataric' 1994; Harvey et al. 1992; Watson et al. 1999]. Other areas presented include artificial life [Colorni et al. 1992; Balch et al. 2001], edutainment [Kitano et al. 1997; Sklar et al. 2000], cognitive science and psychology [Minsky 1987; Braitenberg 1984], and science fiction [Asimov 1950]. The full syllabus for the course is given in Table I.

The course is designed to cover a 14-week semester. The format involves one 75-minute lecture and one 75-minute lab per week. There is a midterm and a final exam. Students are expected to submit written lab reports documenting their efforts, both in terms of software and hardware development. They are encouraged to record results of tests made and changes to their designs. Students also prepare a research project, consisting of both a written report and an oral presentation given to the class.

Table I. Topics Covered in the Introduction to Robotics Course

<input type="checkbox"/> Motors, Effectors and Actuators	<input type="checkbox"/> LEGO Mindstorms construction
<input type="checkbox"/> Sensors	<input type="checkbox"/> LEGO Mindstorms programming
<input type="checkbox"/> Control	<input type="checkbox"/> Robotics in Artificial Life, Cognitive Science and Psychology
<input type="checkbox"/> Behavior	<input type="checkbox"/> Robotics in Literature (Science Fiction) and Edutainment
<input type="checkbox"/> Learning	

Sklar taught this course in Spring 2001 at Boston College [Sklar 2001]; 27 students were enrolled, three of whom were female. All were undergraduates, and there was a mix of ages: first year (1 student), second year (3), third year (10), and fourth year (13). The Computer Science Department at Boston College is in the School of Management. Since there is no engineering school in the university, the hands-on technical experience of these students was limited; sixteen members of the class were computer science majors, the rest came from biochemistry (1 student), communication (1), economics (4), history (2), marketing (1), mathematics (1), and physics (1).

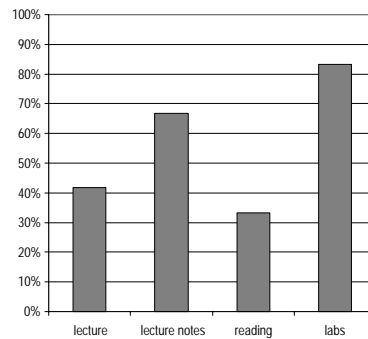
The students were placed in groups of three for working on the robotics projects. Since the experience levels of the class were so diverse, Sklar formed the groups, attempting to balance each group with an equal number of beginning and advanced students. Some lab time was provided during the scheduled course period in order to work on the projects. However, this was not enough time to perfect the robots enough to perform well in the contests, so many students met in the lab, outside of class time, to work on their robots. Each student was required to submit an individual lab report and include an assessment of the contributions of their teammates. The efforts of team members were never balanced; but in reviewing the lab reports, most of the inequities were obvious, even without the peer assessment. Student grades were based on the lab reports, not on their robots' performance in the contests.

The two contests were held in a public space, students were encouraged to invite their friends to come and watch (as shown in Figure 1), and other faculty members also came to observe. The excitement of the crowd and the visibility of the event motivated students to work harder after the first (maze) contest in preparing for the second (soccer) contest.

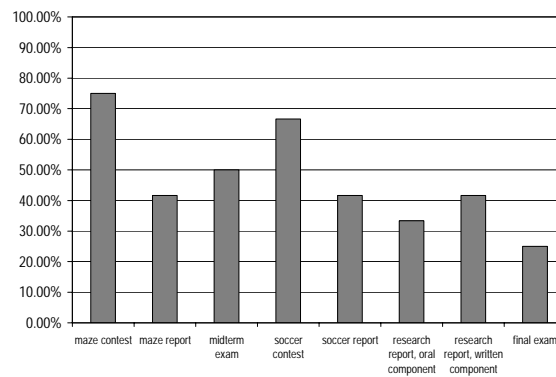
The research projects presented a major challenge for these students, who were not typically asked to do any writing in computer science classes. They were required to submit a brief project proposal several weeks prior to the final due date in order to get started and to provide feedback about the appropriateness of the topic. The range of chosen topics was quite broad, from the use of nanotechnology in surgical robots to the history of robots dating back to ancient Greece. Each student gave a 10-minute oral presentation about his or her chosen topic. This was difficult for many students who were not used to speaking in front of a class. Although discussion following the presentations was encouraged, very little actually occurred; many students did not even attend class on presentation days when they were not speaking. Course evaluation results (below) confirmed that the motivation surrounding the research project was minimal.

4.2 Evaluation

Students were given a survey at the end of the course; 44% of the class responded. The survey collected demographic information and also queried the students about their learning experience. They were asked to identify the elements of the course that were helpful in learning the material and the elements of the assessment that helped them to solidify and demonstrate their knowledge of the subject. The results are shown in Figure 2. Overwhelmingly (83%), the students felt that the labs (i.e., building and programming the robots) were helpful for learning the material, whereas only 33% said that the reading was helpful; 75% (in the maze contest) and 67% (in the soccer contest) responded that the two contests were valuable in helping them solidify and demonstrate their knowledge of the material. This confirms our intuition that the hands-on components provide more effective learning experiences than other aspects of coursework, particularly at the introductory level. We speculate that the readings were perhaps too advanced for most of the class.



(a) Elements of the course which students identified as helpful to them in learning the material



(b) Elements of the course which students identified as helpful in demonstrating their knowledge of the material

Fig. 2. Survey results from the Introduction to Robotics course, Spring 2001.

Student comments were overall quite positive, including the following statements:

- “Great course... loved the [relaxed] atmosphere and hands-on experience. I’d recommend the course to any CS major.”
- “I think the class idea is great. It is a great hands-on experience to try out. The labs were very fun times.”

There were many comments that the mixed-age group was helpful for all students, as the inexperienced students learned from the more advanced, and in assisting others, the advanced students also learned more themselves. Negative remarks centered around requests for more lab time and less time spent on oral presentations.

3. COURSE: ARTIFICIAL INTELLIGENCE

The modern view of artificial intelligence [Russell and Norvig 1995] is that it is the study of intelligent *agents* – autonomous computing systems that perceive their environments and act upon them in a way that both responds to changes in their environments and works toward underlying goals. The intelligent agents metaphor is a way of bringing

together the many strands of work carried out under the banner of AI and presenting them to students in a convincing way. For example, thinking of an agent exploring an environment is a natural way to introduce search techniques, and considering how agents must respond to changes in their environments clearly shows the advantage of behavior-based reactive techniques. Robots are prototypical agents that have to move around and react to their environment in pursuit of their goals. Thus, it is highly appropriate to explore areas of a typical AI syllabus using robotics projects.

Our first effort at this exploration was in the Spring of 2002, when Parsons taught the introductory AI class at Columbia University. The course was built around Nilsson's book, *Artificial Intelligence: A New Synthesis* [Nilsson 1998], which is not only agent-centric, but also covers reactive control architectures in detail early in the syllabus. As a result, it is easy to incorporate material on approaches like subsumption architecture [Brooks 1991] that provide a route to using AI methods, even on a small robot platform like the LEGO Mindstorms. While this course was constrained by time to be mainly lecture-based,³ it ran similar contests to those Sklar used at Boston College (with an additional extra credit project in which students choreographed their robots to move to music in a RoboCupJunior-style dance performance).

The same basic pattern has been used by Parsons to teach introductory AI classes at Brooklyn College, (CUNY) most semesters since Fall 2002. Courses were run for undergraduates in Fall 2002, Spring 2003, and Fall 2003, and for Masters students in Spring 2003, Fall 2003,⁴ and Spring 2004. In Spring 2003, the graduate course used the same robot simulator described in Section 4, while all other iterations have used the same LEGO robots as in the robotics course. The enrollment for the various offerings are summarized in Table II. For each class, we give the total enrollment, the number of female students, the number of surveys we collected, the number of computer science/information science majors, and the number of graduate and undergraduate students. It is important to note that the latter three figures are derived from the surveys, not from the enrollment.

All versions of the course, both undergraduate and graduate, are advanced electives and have as prerequisites courses in programming in C and data structures. Many students who take the AI course have also covered discrete math, which means that they already have seen many of the concepts in the parts of the course that deal with propositional and predicate logic. Most of the students who take the course are computer science majors, although we typically have one or two students a year who major in different subjects.

The course has evolved steadily; the syllabus has been modified, and is now largely based on the book by Russell and Norvig [1995], entitled, *Artificial Intelligence: A Modern Approach*, but the course still includes the material on reactive control mentioned above. We now provide many more resources to the students at the start of the project. Early on, we organize one or two supervised lab sessions, during which students can gain initial experience with the robots, and since there are instructors on hand, students are less likely to make common errors. On several occasions the instructors were past graduates of the course, and were very keen to pass on their expertise. We have also refined the contests (the refinements are explained in detail below).

³ Though it should be noted that this does not necessarily rule out regular practical work. The LEGO kits are handy enough that they can easily be used in a classroom, and with inexpensive laptops liberate robotics from the laboratory.

⁴ The Fall 2003 graduate course was slightly more advanced.

Table II. Course Offering Demographics

Offering	Enrollment	Female	Surveys	CS majors*	Under-grad*	Grad*
S 2001, BC (U)	27	3 (11%)	12 (44%)	16 (59%)	27 (100%)	0 (0%)
F 2001, NYU (G)	15	1 (7%)	13 (87%)	15 (100%)	0 (0%)	15 (100%)
S 2002, CU (U/G)	35	6 (17%)	19 (54%)	14 (74%)	13 (68%)	6 (32%)
F 2002, BC-CUNY (U)	18	7 (39%)	18 (100%)	16 (89%)	18 (100%)	0 (0%)
F 2002, UTA (U)	15	0 (0%)	12 (80%)	15' (100%)	15 (100%)	0 (0%)
S 2003, BC-CUNY (U)	9	1 (11%)	6 (67%)	5 (83%)	6 (100%)	0 (0%)
S 2003, BC-CUNY (G)	19	5 (26%)	13 (68%)	12 (92%)	0 (0%)	13 (100%)
F 2003, BC-CUNY (U)	19	3 (16%)	15 (79%)	13 (87%)	15 (100%)	0 (0%)
F 2003, BC-CUNY (G)	21	7 (33%)	16 (76%)	16 (100%)	0 (0%)	16 (100%)
S 2004, BC-CUNY (U/G)	15	3 (20%)	15 (100%)	15 (100%)	1 (7%)	14 (93%)
S 2004, UTA (U)	24	3 (13%)	24 (100%)	24 (100%)	24 (100%)	0 (0%)
Total	217	39	163	161	119	64

Notes:

* = percentage of students who completed the surveys (not overall percentage of enrollment)

' = some students were Computer Engineering majors

Location Key:

BC = Boston College

BC-CUNY = Brooklyn College, CUNY

CU = Columbia University

NYU = New York University

UTA = University of Texas at Austin

Class Level Key:

(U) = undergraduate

(G) = graduate

(U/G) = mixed undergraduate and graduate

Table III. Artificial Intelligence Course Topics

<input type="checkbox"/> Introduction to AI	<input type="checkbox"/> Knowledge representation
<input type="checkbox"/> What is an agent?	<input type="checkbox"/> Rule-based expert systems
<input type="checkbox"/> Introduction to robotics	<input type="checkbox"/> Propositional logic
<input type="checkbox"/> Reactive control	<input type="checkbox"/> Predicate logic
<input type="checkbox"/> Behavior-based control	<input type="checkbox"/> Common-sense reasoning
<input type="checkbox"/> Perceptrons	<input type="checkbox"/> Means-ends planning
<input type="checkbox"/> Machine vision	<input type="checkbox"/> Partial-order planning
<input type="checkbox"/> Heuristic search	<input type="checkbox"/> Reinforcement learning
<input type="checkbox"/> Adversarial search	

4.3 Course Structure

The current version of the course is taught over 14 weeks, structured as two 75-minute lectures per week, either delivered on two separate days (for daytime courses) or once a week in one 120-minute session (for an evening course). The course features two robotics projects, plus two exams and a number of homework assignments. The way the various offerings are structured is best described by the material on the course web pages [Parsons 2002; 2005]. The web pages give detailed syllabi for each course offering; all

the syllabi use a subset of the topics outlined in Table III. We have developed curriculum modules for each of these topics and vary the modules we include in each offering. The web pages also give the course schedule, homework, additional readings, lecture notes, and the details of the robotics projects.

The projects are group efforts, and account for 25% of the term grade. The projects start around the third week of the semester, and each project runs for 5 to 6 weeks. Every student has to write a report on the two robotics projects. We found that the optimum group size is two students (if there are more, then it is easy for one student to become a passenger or to get shut out); although limitations on the number of robots has often forced us to have three or more in a group. Each group is supplied with a kit that contains the following:

- around 200 LEGO parts from which to build their robot (a subset of the 700 or so pieces that are a part of the standard LEGO Mindstorms kit), including the RCX (the Mindstorms microcontroller), two motors, and several sensors (light, touch, and rotation sensors); and
- a set of robot designs and sample code to run on the robots.

There is an initial instructional lab session in which the students build and program a very simple robot in order to become familiar with the basic concepts. The lab room is also made available outside of class-time for the students to get together and practice running their robots under the same conditions in which they will be evaluated in the contests. Where time permits, other instructional lab sessions are also scheduled, but every such session means one less lecture (and several lecture sessions are spent running the contests at the end of the projects).

The robot designs and sample code given to the students are taken from Baum [2000] (a book we thoroughly recommend to anyone thinking of using the LEGO Mindstorms), and chosen to illustrate various issues the students will have to deal with in the contests. The set of parts is sufficient to build all the robot designs; although, for the contests, none of the designs will suffice without modification. The idea is to try and balance the help given to students with mechanical design (since this aspect of the project has the least connection to the subject matter of AI) and providing a complete solution. Similarly, the code covers many of the necessary aspects without coming close to solving the problems. We do not give the students the entire LEGO Mindstorms kit for the following reasons:



Fig. 3. Sample student robot designs.

- Students have to carry the pieces to and from campus (since there is no place the robots can be stored on campus and because some students like to work on

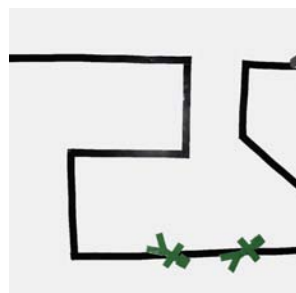
their robots at home). Reducing the kit to something that will fit in a medium-sized box reduces the transportation burden, especially for students who commute over an hour on the subway as many of the students at CUNY do.

- Pieces will get lost. Keeping some pieces in reserve (especially some of the smaller ones) makes it possible to absorb these losses without having to purchase new kits each term.

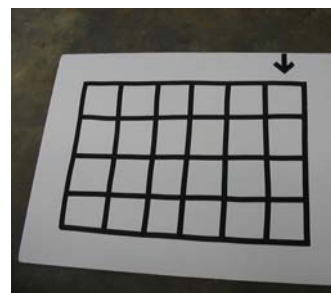
Our experience has shown that this reduced set of parts does not greatly restrict the range of designs that our students manage to create. For example, Figure 3 shows two different designs that were created by students who took the course.

We are in the process of fully documenting sample robot designs and programs and selecting different sets of parts appropriate for different projects. These include programs to illustrate specific agent architectures such as subsumption [Brooks 1991] and the belief/desire/intention architectures (BDI) [Bratman et al. 1988], both implemented using the materials from the course; we also include robot designs that are more flexible than those given in Baum [2000].

The contests in which the students take part are designed to test the full range of what is possible with Mindstorms and the use of the Not Quite C (NQC) language. The first challenge in the contest is the same line-following task as in the introduction to robotics course described in Section 2. It is intended to be introductory, and therefore relatively straightforward. The main task, in the style of RoboCupJunior rescue [Sklar 2004], is to follow a black line on a white background around a number of curves (both left-hand and right-hand turns). In order to make the task harder, the contest has sometimes involved climbing and descending a gradient (as in Figure 1(a)), but more recently we have moved toward using a flat maze as in Figure 4(a); this seems more appropriate for the level of the students taking the course. The fastest robot to follow the line from one end to the other wins. To add a little complexity, there are three further tasks: the first is that the robot has to detect an obstacle using touch sensors, i.e., once it indicates it has detected this (by backing up), the obstacle is removed. The second additional task is to recognize colored areas (shaped like human “victims,” after the RoboCupJunior rescue challenge) in the middle of the line and to indicate they have been found by playing a tune. Since the colored areas give a reading between that of the black line and the white background and have to be detected using the same sensors as detect the line, they introduce a trade-off between



(a) Maze without a gradient



(b) Gridworld

Fig. 4. Sample contests.

speed and accuracy of detection. Failure to detect the victims results in a time penalty, as does detecting victims that are not there. The final task is to detect colored areas (made of

a reflective material that appears to the light sensor as brighter than white) at the end of the course and for the robot to stop when the bright regions are reached.

The reason the first challenge is simple is in part because the robot's environment is static. In contrast, the second challenge most often involves a dynamic world: groups have to build robots to play a one-on-one version of the same RoboCupJunior robot soccer game [Lund and Pagliarini 1998] used in Sklar's course.⁵ The field for this game is shown in Figure 1(b). It is carpeted in grayscale (black at one end and white at the other, changing gradually from black to white), to give directional information that can be read with a light sensor. The ball used in the contest emits infra-red light, and so is relatively easy to detect using a second light sensor. The contest takes the form of a soccer tournament in which the project groups are split into two pools; each pool plays a round-robin of three-minute games (which seems to be long enough to score goals most of the time, but short enough to fit a class period) followed by a final match between the winners of the two round-robin pools.

The problem with the soccer project is that it does not fit a course on AI particularly well. The instructions for the line-following project make it an exploration of reactive behavior, and while the soccer project is fun for the students and easily within their reach, it does not necessarily involve any additional introductory AI techniques. As a result, in the most recent offering, we have experimented with a new challenge.

The students are confronted with a grid-world (shown in Figure 4(b)) delineated by black lines like those in the first contest (Figure 4(a)); some of the squares in the grid-world contain the same "victims" as in the first contest. The challenge is to survey the grid, identify the positions of the victims, and then re-position the robot (at the arrow in Figure 4(b)) and move to the figures in a prespecified order in the shortest possible time. The idea is to bring in some of the concepts related to *search* that the students have covered in the course, and combine them with the reactive techniques from the first contest (which are still required to move the robots around the grid). Since the robots cannot localize, this is a hard challenge; but completing it is within the capabilities of the more able students.

On occasion we have offered students the option of an extra-credit project, i.e., building a dancing robot exactly as in the RoboCupJunior dance competition. While this is generally great fun for the students, it does not involve much AI (if any).

The combination of the Mindstorms kit, NQC programming language, and challenges seem appropriate for the students who take these courses. But it is possible to go beyond what we are doing in terms of sophistication while still using the Mindstorms platform. This can be done by purchasing different sensors, which will extend the kinds of tasks students can address. For example, LEGO sells a rotation sensor that can be used to supply odometry information, and it is possible to purchase inexpensive infra-red range finders [Mindsensors 2005] and even a compass [Wiltronics 2003]. It is also possible to extend the range of tasks by using a more sophisticated programming language. BrickOS [Baum et al. 2000, BrickOS 2004] provides a more complete version of C than NQC, and one can also program the Mindstorms in Java [Laverde et al. 2002] and Lisp [Klassner 2002; Klassner and Anderson 2003].

The Lisp interface expands the range of tasks that can be solved with RCX by allowing control programs to run off-board (with instructions sent to the RCX via its IR communication port). A similar set-up has been used by Sklar for the RoboCupJunior E-

⁵ For the course offered at Columbia, the contest is to have a robot score in an open goal; performances were compared based on time to score, if a goal was scored.

league [Anderson et al. 2003]. This approach overcomes the 32K RAM (the main drawback) with which the RCX is fitted.

4.4 Evaluation

Two evaluations were administered for the Columbia offering of the course. One was an official evaluation done by the engineering school; the other was an informal paper-and-pencil survey given out in class, which broadly asked the same questions as those given to students on Sklar's course. The results of the engineering school's evaluation showed that 55% of the 33 students who responded gave the robotics project a rating of 5 (on a 5-point scale) for interest, and two-thirds gave it a rating of 4 or 5. Twenty-one percent (21%) of the same cohort of students gave the project a rating of 5 for the amount learned during the project and 58% rated it 4 or 5.

The informal survey probed more into the students' perceptions of the value of the project relative to other aspects of the course. In particular, as in Sklar's Boston College evaluation, students were asked to identify which aspects of the course contributed most to helping them learn the material and which were most helpful to them in demonstrating their knowledge of the material. The same informal survey was administered to students in the subsequent six offerings of the course, and overall gave us a considerable amount of data on the projects. The results of the survey are given in Figures 5 and 6, along with combined results from all the surveys.

The results show fairly consistently that the project was felt to be less helpful in learning AI than more traditional elements like lectures, lecture notes, and homework, but was more helpful than additional readings or the textbook. In terms of demonstrating their knowledge, the students felt that the project was more helpful than the final exam, and some felt it to be as helpful as the midterm exam.

The one set of results that is inconsistent with these findings is the one for the Spring 2003 graduate course. In this course the projects were performed using a robot soccer simulator⁶ (the same one as in the courses described in Section 4). It is tempting to take the much lower figures as a reflection of the fact that programming simulated robots is less satisfying than programming the real thing, but it might equally well reflect the fact that simulator presents a more challenging programming environment than the RCX.⁷

Overall, we think these results, and similar ones from other offerings of the course, are convincing enough to warrant continued use of robotics. But perhaps more encouraging than the survey responses were the free-form comments we received, including the following:

- "When working with the robot, I learnt that nothing is perfect in the real world. A lot of times the outcome is very unexpected."
- "It reminded me of why I want to stay away from hardware as much as possible."
- "It helped immensely! It helped me understand some of the concepts covered in the lecture."

⁶ The RoboCup Soccer Simulator [2004].

⁷ Verbal feedback from the students suggests that those who mastered the simulator were very happy with the project.

Using Robo-Cup in University-Level Computer Science Education

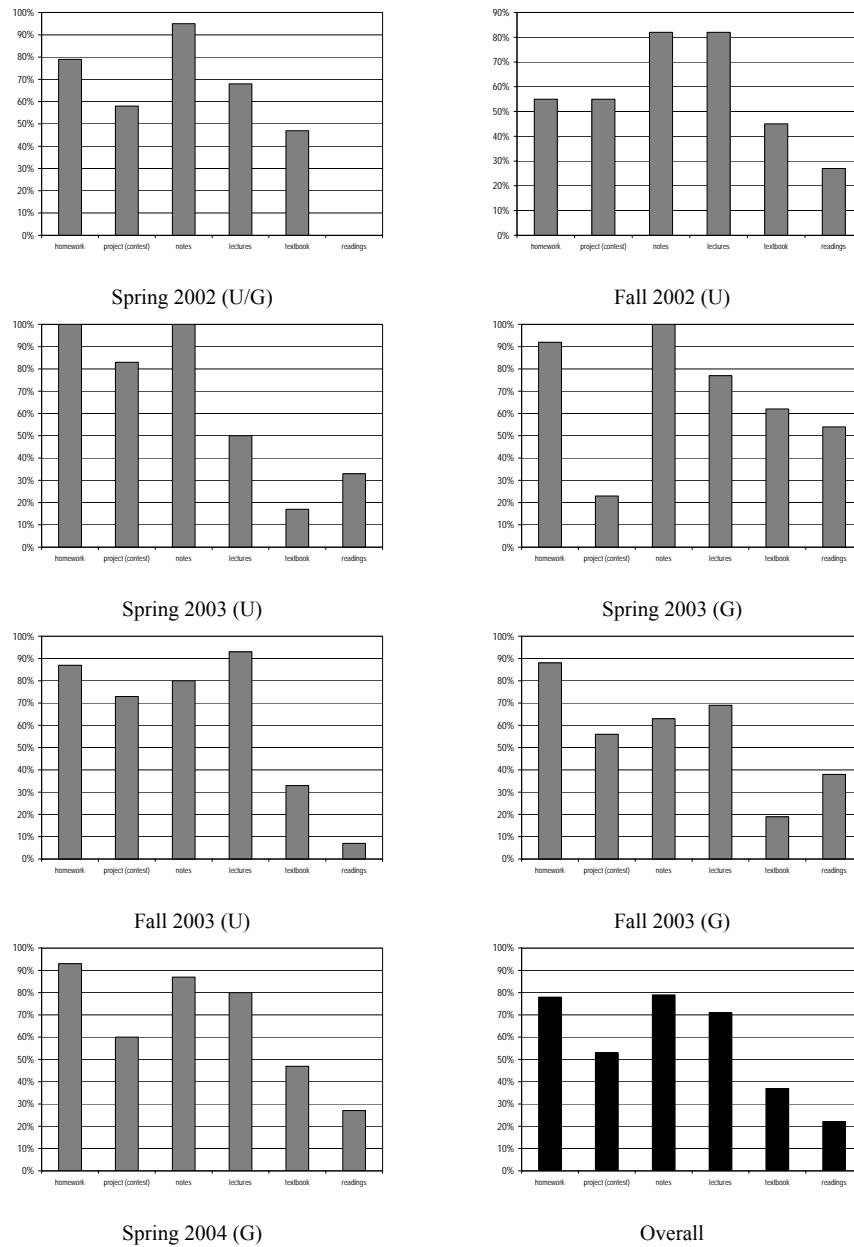


Fig. 5. Elements of the course identified by students as helpful in learning the material.

- “[The] project helped [me] to realise how important [it] is to divide complicated tasks into smaller ones in order to solve [them].”
- “It is nice to put theory to practice.”

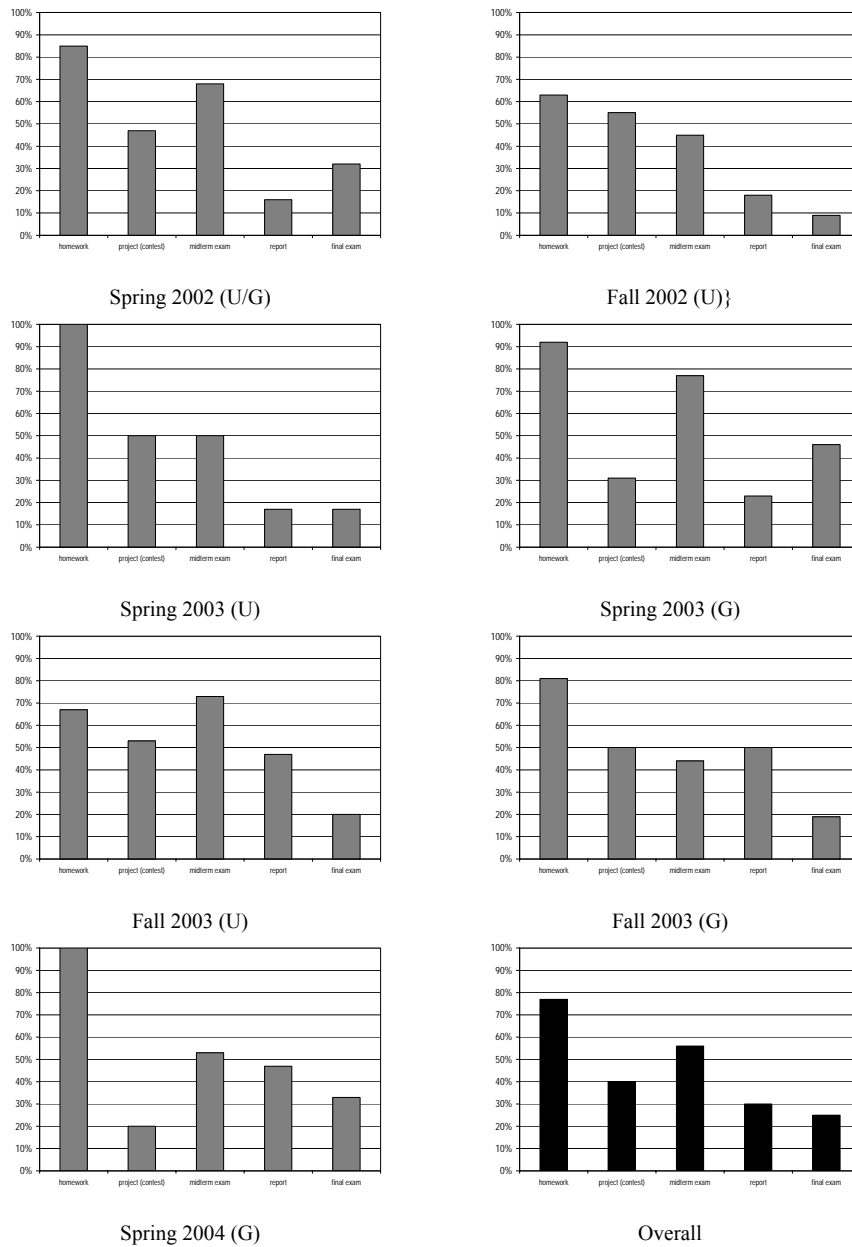


Fig. 6. Elements of the course identified by students as helpful in demonstrating their knowledge of the material.

In addition, in the most recent undergraduate offering of the course, 14 of the 15 students responded via free-form comments that the use of robotics was beneficial and that the project work was fun.

It should be noted that some of the positive reactions may be because it is unusual for students at Brooklyn College to do open-ended project work (in the open comment part of the survey, several students confessed that this was the only undergraduate project they had participated in). This observation is supported by the fact that broadly similar results were generated by students who opted for a non-robotics project when such projects were offered in Fall 2002 (although the very small number of students in this category makes the results extremely unreliable, and so they are not presented here). Comments of a similar nature appeared in surveys from the University of Texas at Austin undergraduate cohort, described in Section 4.

4. COURSE: AUTONOMOUS MULTIAGENT SYSTEMS

Over the years, autonomous agents and multiagent systems (AAMAS) is a field to which RoboCup participants have contributed consistently and prominently. Multiagent systems are collections of multiple autonomous agents that interact with one another. The AAMAS field covers a wide variety of research foci and applications, including software-based information processing, robotic control of multiple agents, entertainment agents, and tutoring agents [Gini et al. 2002]. Our course provides a broad introduction to autonomous agents with an emphasis on multiagent systems. Topics include agent architectures, interagent communication, agent teamwork, distributed rational decision-making, agent modeling, multiagent learning, and entertainment agents. A full syllabus is given in Table IV. The formal prerequisites for the course are stated as “Good programming skills, preferably in C and/or C++. Some background in artificial intelligence is recommended but not essential.” In practice, this restricts the course to upper-level computer science students.

In addition to teaching AAMAS, the course aims to introduce undergraduates to the full spectrum of research activities engaged in by professional computer science researchers, emphasizing the difference between their activities and those of typical undergraduate students [Stone 2004b]. The course includes an open-ended programming project, readings from the research literature, public speaking and writing requirements, and opportunities to collaborate with peers. In order for students to succeed, they need to master the AAMAS subject. Assessing their capacity to do so is based primarily on their ability to demonstrate their knowledge by engaging in the full range of activities required of researchers.

4.5 Course Structure

The central focus of the course is a semester-long class build-up towards a simulated robotic soccer competition using the RoboCup Soccer Server [Noda et al. 1998]. Students are assigned a series of four preliminary programming assignments designed to familiar-

Table IV. Topics Covered in the Autonomous Multiagent Systems Course

<input type="checkbox"/> Introduction	<input type="checkbox"/> Applications
<input type="checkbox"/> Autonomous agents	<input type="checkbox"/> Game theory
<input type="checkbox"/> Agent architectures	<input type="checkbox"/> Distributed rational decision-making
<input type="checkbox"/> Multiagent systems	<input type="checkbox"/> Auctions
<input type="checkbox"/> Agent communication	<input type="checkbox"/> Agent modeling
<input type="checkbox"/> Teamwork	<input type="checkbox"/> Multiagent learning
<input type="checkbox"/> RoboCup case studies	<input type="checkbox"/> Entertainment agents
<input type="checkbox"/> Swarms and self-organization	

ize them with the Soccer Server and some client code [Noda and Stone 2003]. By the end of these preliminary assignments, the students have created a fully functional team (although not one that is particularly competent). The students are then encouraged to propose some improvement to the functioning of the team as the topic for their final projects.⁸ For example, one student proposed using machine-learning techniques to train a good goal-tender without paying attention to the rest of the team.

The majority of the readings for this course are primary sources chosen to both introduce particular topics and to engender some controversy (e.g., reactive [Brooks 1991] versus deliberative [Simmons 1994] agent architectures). To encourage the students to complete the readings in a timely fashion, they are required the night before class to submit a brief written answer to a single question pertaining to the readings. The fact that the responses are due before class allows the instructor to incorporate them into the class discussion and the students to come to class prepared to discuss the readings; resulting in many extended and heated class discussions.

An important component of class participation is the requirement that each student moderate at least one class discussion pertaining to that week's readings. The student is instructed to either defend a controversial statement or pose a question and be prepared to defend either side, depending on how the class reacts. This activity turns out to be one of the most difficult to complete. Many students are not used to speaking in front of a class, and have rarely been put in the position of facilitating discussions as opposed to defending specific positions.

The course requires a good deal of writing from the students. As above, the students are asked to provide weekly written responses to questions related to the readings. They receive feedback pertaining to the clarity and soundness of their responses. More significantly, the students are expected to write three documents pertaining to their final projects. First, they write project proposals defining their goals as well as propose methods for achieving them; second, they revise their proposals and add a section on the work-in-progress to create progress reports; finally, they write final reports in the format of conference papers.

On the final project, students have the option of working in pairs. Teams must write their proposals and reports individually, with clear indications of what role each person played in the collaboration; and hence more is expected from them as a final product. The robotic soccer project lends itself to such collaboration nicely, since there are many different ways in which the students can divide the work.

The course culminates in a simulated soccer tournament. The students are told at the outset that performance in the tournament will not have a negative impact on their grades (while a strong performance can have a positive impact). Nonetheless, the tournament is a highly motivating factor for the students. Visitors are invited to the event, and the students present their approaches orally and field questions as their teams are playing.

The performance spread among the teams is often very large, especially due to the fact that some students did not focus on creating winning teams. All class champions were tested against a mid-range RoboCup entry and lost significantly; the students were not able to attain competitive world-class levels despite starting with a fairly detailed client code base. However, given their time limitations, this fact is neither surprising nor discouraging.

⁸ They are also given the option to propose a programming project in a multiagent domain of their choice; but typically, few, if any, students choose to do so.

4.6 Evaluation

Stone has taught this course three times: first at New York University (NYU) in the Fall of 2001. Fifteen students were enrolled, only one of whom was female. All students were (as undergraduates) computer science majors, all were graduate students: 12 master and 3 doctoral. The course was offered the second time at the University of Texas at Austin during Fall 2002 [Stone 2002]. Again, 15 students were enrolled, but none were female. This time the cohort was made up of undergraduates: 14 were seniors (fourth year) and one was a junior (third year). Most students were computer sciences majors, with the remainder majoring in computer engineering. The third, and most recent offering of the course was also at UT Austin during Spring 2004 [Stone 2004a]. This class had 24 undergraduate students, all computer sciences majors, including three females. Most of the students were seniors, though the class did include some sophomores and juniors.

Course evaluations and surveys were administered at the conclusion of all three courses. At NYU, the course was rated 4.5 out of a possible 5 (highest rating). At UT Austin, the course was rated 4.6 out of a possible 5 on both occasions. Student comments also provide evidence that the students appreciated the opportunity to be exposed to the various components of scientific research. Every time the course was run, at least one student described the course as a primary motivator for wanting to do research in graduate school. In addition, one student from the Fall 2002 offering contributed actively to the UT Austin entries in the RoboCup 2003 competition, including the world champion simulator coach team [Kuhlmann et al. 2005].

An informal survey was administered in the middle of the term to the UT Austin cohort. Students were asked to rate the programming assignment on a scale of 1 to 5. Thirty-three percent (33%) gave it the highest rating; 57% gave it the second highest, while 10% scored it average, and no students entered low marks. Students also rated the reading assignments on the same scale. Twenty percent (20%) gave the highest rating; 60% gave the second highest, 13% scored it average, and 7% gave the lowest rating.

Some of the comments from students follow:

- “I really like reading from research literature. Just the fact that it’s actual research provokes curiosity.”
- “The discussions we have in class are quite unique, I haven’t had such involving discussions in any class before.”
- “Format of the class is perfect. I’ve waited through three years of college for a class like this... I like the class so much that my other classes now disappoint me.”
- “The only thing I dislike about the class is that we are limited in our application of our knowledge. Our education in AI is directed at implementing a RoboCup soccer agent. I feel that if we were able to apply our knowledge to other aspects, we would gain an even better understanding of artificial intelligence.”
- “The simulator code was kinda tricky to understand.”
- “I really enjoyed this class. It is definitely the type that you get out of it what you put into it. I wasn’t sure how doing the readings every week would be, but at the end I looked forward to the next week’s readings. I also liked the end goal of the tournament, it’s a helpful motivation to work on the final project. And with flexible times, it was enjoyable to work on, instead of a pressured assignment to finish. I *really* liked the research aspects because it was an insight different than most other classes.”

- “Speaking in front of class is good and rare experience. One of the few courses where one can test his feelings about research. I feel my writing has also improved.”
- “This was a great course and one of the most rewarding things is seeing my team score a goal through seemingly intelligent behavior.”
- “This is perhaps one of the most entertaining computer sciences classes I’ve taken since I’ve been here. I’ve had a lot of fun, and learned an immense amount of stuff.”
- “I think the open project is an excellent idea. Especially with the proposal phase and everything; it’s quite similar to the limited experiences I’ve had trying to get projects going/funded and I think it’s a great opportunity to practice/see it firsthand.”
- “The problems are not well defined, but that is not necessarily a bad thing. It allows us to be creative which usually creates an improved understanding of the task.”
- “It’s good that we have relative autonomy over what kind of project we can implement.”

5. DISCUSSION

The three courses described above offer an interesting comparison, not only in terms of content and presentation but also in regard to the cohorts of students enrolled in them. Collectively, the courses have been offered 11 times at 5 different universities, providing a broad range of backgrounds, demographics, and experience levels – from first-year undergraduates at a private, nonengineering college to fourth-year undergraduates at a large state university, and including graduate students from both an urban public university and two large private universities (Table II contains a summary of these statistics). Thus the positive feedback from all participants in regard to the robotics projects is encouraging and significant.

Comments about the reading materials were typically less enthusiastic, as the figures presented in the previous section indicate. Stone’s course, where students were required to respond to the readings with short written assignments and then given the opportunity to discuss the readings formally in class, fared better than other more traditional courses. In these courses the material was presented by the instructors primarily in lectures without written responses, where questions were encouraged (but infrequent), and discussion was not central. Stone’s classes had comparatively fewer students, so more effective discussion was possible. Nonetheless, several students from all the courses commented that they wished there had been better connections between the readings and the project work. This type of feedback is valuable to us and to our colleagues for improving the existing courses, as well as for designing new ones.

The challenge to students who were required to make oral presentations to the class (in both Sklar’s and Stone’s courses) is also notable. No matter what career path is ultimately taken, students need to know how to communicate their ideas. The development of oral presentation skills is important and should be encouraged, despite students’ dislike of this aspect of the courses. Perhaps more creative ways of oral reporting can be incorporated into all the courses to make it more effective and palatable.

The evaluations performed on Stone’s course so far were limited to generalized questions about whether students liked the course and standard questions about the instructor and workload. While this level of information is useful to administrators, we are interested in gathering more specific data on the students’ learning experiences, as

illustrated by the survey shared by Sklar and Parsons. The discrepancy in evaluation methodology from one course offering to another is one of the factors that has spurred us to create the repository mentioned here. The repository will include a standard instrument for measuring the effectiveness of specific course-work and the general RoboCup learning environment.

6. SUMMARY

We have presented three university-level courses that cover topics related to robotics in various ways, with a particular focus on RoboCup challenges. Evaluations, both qualitative and quantitative, were conducted at the end of each of the courses (the student populations were diverse, from public and private institutions, inner-city and suburban settings, in northeastern and southwestern locations). In all cases, the inclusion of a RoboCup-style contest was a highly motivating factor, both for fun and to help students assimilate, reaffirm, apply, and demonstrate their knowledge of the subject matter.

We will continue our efforts in the existing courses described above as well as in the development of new courses. Using robotics as a basis for demonstration, Sklar and Parsons are currently adapting materials for an introductory computer science curriculum; Parsons introduced projects that use Lego robots into introductory Java programming classes in Fall 2004 and Spring 2005.

One of our long-term goals is to build an on-line space for sharing curricular materials and to develop a unified instrument and database for evaluating the RoboCup learning environment. As examples of the type of information we are seeking and archiving, we have presented in this article an account of our collective experiences incorporating RoboCup activities into undergraduate courses. We hope to encourage others to join in this community venture. Our evolving online repository can be found at <http://www.sci.brooklyn.cuny.edu/~sklar/er>. We welcome contributions and participants.

REFERENCES

- ANDERSON, J., BALTES, J., LIVINGSTON, D., SKLAR, E., AND TOWER, J. 2003. Toward an undergraduate league for RoboCup. In *Proceedings of RoboCup-2003 Conference: Robot Soccer World Cup VII*.
- ARKIN, R. C. 1998. *Behavior-Based Robotics*. MIT Press, Cambridge, MA.
- ASIMOV, I. 1950. *I, Robot*. Doubleday, Garden City, NY.
- BALCH, T., KHAN, Z., AND VELOSO, M. 2001. Automatically tracking and analyzing the behavior of live insect colonies. In *Proceedings of the 2001 Conference on Autonomous Agents* (Montreal, Quebec).
- BAUM, D. 2000. *Dave Baum's Definitive Guide to LEGO Mindstorms*. APress.
- BAUM, D., GASPERI, M., HEMPEL, R., AND VILLA, L. 2000. *Extreme MINDSTORMS: An Advanced Guide to LEGO MINDSTORMS*. APress.
- BIRK, A. 1998. Behavior-based robotics, its scope and its prospects. In *Proceedings of The 24th Annual Conference of the IEEE Industrial Electronics Society*. IEEE Press.
- BIRK, A. 2001. Autonomous systems. <http://www.faculty.iu-bremen.de/birk/lectures/COURSES/autosys.html>
- BOTBALL. 2005. <http://www.botball.org>
- BRAITENBERG, V. 1984. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, MA.
- BRATMAN, M. E., ISRAEL, D. J., AND POLLACK, M. E. 1988. Plans and resource bounded practical reasoning. *Computational Intelligence* 4 (1988), 349-355.
- BRICKOS. 2004. BrickOS at SourceForge. <http://brickos.sourceforge.net>.
- BROOKS, R. A. 1986. A robust layered control system for a mobile robot. *IEEE J. of Robotics and Automation* 2, 1 (1986).
- BROOKS, R. A. 1991. Intelligence without representation. *Artificial Intelligence* 47 (1991), 139-159.
- COLORNI, A., DORIGO, M., AND MANIEZZO, V. 1992. An investigation of some properties of an ant algorithm. In *Proceedings of the Parallel Problem Solving from Nature Conference (PPSN 92)*. R. Maner and B. Manderick, eds. Elsevier, New York.
- CORADESCHI, S. 2000. TDDA 14: Artificial intelligence programming. <http://www.ida.liu.se/~silco/AIP/>.

- CORADESCHI, S. AND MALEC, J. 1998. How to make a challenging AI course enjoyable using the RoboCup soccer simulation system. In *RoboCup-98: Robot Soccer World Cup II*. Lecture Notes in Artificial Intelligence (LNAI). vol. 1604. Springer Verlag, New York.
- GINI, M., ISHIDA, T., CASTELFRANCHI, C., AND JOHNSON, W. L., EDS. 2002. *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*. ACM Press, New York.
- HANDY BOARD. 2003. The handy board. <http://www.handyboard.com>.
- HARVEY, I., HUSBANDS, P., AND CLIFF, D. 1992. Issues in evolutionary robotics. Cognitive Science Research Tech. Rep. Paper Serial CSRP 219. Univ. of Sussex School of Cognitive and Computing Sciences, Brighton, UK.
- HEINTZ, F. 2000. RoboSoc, a system for developing RoboCup agents for educational use. M.S. thesis, Dept. of Computer and Information Science, Linköpings Univ.
- KITANO, H., ASADA, M., KUNIYOSHI, Y., NODA, I., AND OSAWA, E. 1997. RoboCup: The Robot World Cup Initiative. In *Proceedings of the First International Conference on Autonomous Agents (Agents-97)*.
- KLASSNER, F. 2002. A case study of LEGO Mindstorms suitability for artificial intelligence and robotics courses at the college level. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*. 8-12.
- KLASSNER, F. AND ANDERSON, S. 2003. LEGO Mindstorms: Not just for K-12 anymore. *IEEE Robotics and Automation*. (June 2003).
- KUHLMANN, G., STONE, P., AND LALLINGER, J. 2005. The UT Austin Villa 2003 champion simulator coach: A machine learning approach. In *RoboCup-2004: Robot Soccer World Cup VIII*. D. Nardi et al. eds. Springer Verlag, Berlin, 636-644.
- LAVERDE, D., FERRARI, G., AND STUBER, J. 2002. *Programming Lego Mindstorms with Java*. Syngress.
- LEGO MINDSTORMS. 2005. <http://www.legomindstorms.com>.
- LITTMAN, M. 1999. CPS196: Programming under uncertainty. <http://www.cs.duke.edu/~mlittman/courses/cps196/>.
- LUND, H. H. AND PAGLIARINI, L. 1998. Robot soccer with Lego Mindstorms. In *RoboCup-98: Robot Soccer World Cup II*. Lecture Notes in Artificial Intelligence (LNAI), vol. 1604. Springer Verlag, New York.
- MARTIN, F. 1989. 6.270: The MIT LEGO Robot Design Project Competition. <http://web.mit.edu/6.270/www/2004/about/history.html>.
- MARTIN, F. 1996. Kids learning engineering science using LEGO and the programmable brick. In *Proceedings of the Conference of the American Educational Research Association (AERA-1996)*.
- MARTIN, F. 2000. *Robotic Explorations: A Hands-On Introduction to Engineering*. Prentice Hall, Englewood Cliffs, NJ.
- MARTIN, F., MIKHAK, B., RESNICK, M., SILVERMAN, B., AND BERG, R. 2000. To Mindstorms and beyond: Evolution of a construction kit for magical machines. In *Robots for Kids: Exploring New Technologies for Learning*.
- MATARIC, M. J. 1994. Reward functions for accelerated learning. In *Machine Learning: Proceedings of the Eleventh International Conference*. W. W. Cohen and H. Hirsh, eds. Morgan Kaufmann.
- MATARIC, M. J. 1997. Behavior-based control: Examples from navigation, learning, and group behavior. *J. Experimental and Theoretical Artificial Intelligence*. Special Issue on Software Architectures for Physical Agents 9 (1997), 2-3.
- MATARIC, M. J. 1998. CS 445 introduction to robotics, a LEGO-kit-based hands-on lab course. <http://www-scf.usc.edu/~csci445/>.
- MCKERROW, P. J. 1991. *Introduction to Robotics*. Addison-Wesley, Reading, MA..
- MINDSENSORS. 2005. <http://www.mindsensors.com>.
- MINSKY, M. 1987. *Society of Mind*. Picador.
- NILSSON, N. J. 1998. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann, San Francisco.
- NODA, I., MATSUBARA, H., HIRAKI, K., AND FRANK, I. 1998. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence* 12 (1998), 233-250.
- NODA, I. AND STONE, P. 2003. The RoboCup soccer server and CMUnited clients: Implemented infrastructure for MAS research. *Autonomous Agents and Multi-Agent Systems* 7, 1&2 (July 2003).
- NQC. 2005. Not Quite C. <http://bricxcc.sourceforge.net/nqc/>.
- PARSONS, S. 2002. COMS W4701: Artificial intelligence. <http://www.cs.columbia.edu/~sp/4701-2.html>.
- PARSONS, S. 2005. CIS 32/CIS 716: Artificial intelligence. <http://www.sci.brooklyn.cuny.edu/~parsons/courses/courses.html>.
- ROBOCUP. 2005. <http://www.robocup.org>
- ROBOCUP JUNIOR. 2005. <http://www.robocupjunior.org>
- ROBOCUP SOCCER SIMULATOR. 2004. The RoboCup soccer simulator. <http://sserver.sourceforge.net>.
- RUSSELL, S. J. AND NORVIG, P. 1995. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ.

Using Robo-Cup in University-Level Computer Science Education

- SIMMONS, R. 1994. Structured control for autonomous robots. *IEEE Trans. on Robotics and Automation* 10, 1 (Feb. 1994), 34-43.
- SKLAR, E. 2001. MC375: Introduction to robotics. <http://www1.cs.columbia.edu/~sklar/teaching/spring2001/mc375/default.html>.
- SKLAR, E. 2004. A long-term approach to improving human-robot interaction: RoboCupJunior Rescue. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.
- SKLAR, E. AND EGUCHI, A. 2004. RoboCupJunior – Four years later. In *Proceedings of the Eighth International RoboCup Symposium*.
- SKLAR, E., EGUCHI, A., AND JOHNSON, J. H. 2002. RoboCupJunior: Learning with educational robotics. In *Proceedings of RoboCup-2002: Robot Soccer World Cup VI*.
- SKLAR, E., JOHNSON, J. H., AND LUND, H. H. 2000. Children learning from team robotics: RoboCupJunior 2000 Educational Research Report. Tech. Rep., The Open University, Milton Keynes, UK.
- SKLAR, E. AND PARSONS, S. 2002. RoboCupJunior: A vehicle for enhancing technical literacy. In *Proceedings of the AAAI-02 Mobile Robot Workshop*.
- STONE, P. 2002. CS378: Autonomous agents and multiagent systems. <http://www.cs.utexas.edu/~pstone/Courses/378fall02>.
- STONE, P. 2004a. CS378: Autonomous agents and multiagent systems. <http://www.cs.utexas.edu/~pstone/Courses/378spring04>.
- STONE, P. 2004b. RoboCup as an introduction to CS research. In *RoboCup-2003: Robot Soccer World Cup VII*. D. Polani, eds. Springer Verlag, Berlin.
- VIDAL, J.M. AND BUHLER, P. 2002. Teaching multiagent systems using RoboCup and biter. *The IMEJ of Computer-Enhanced Learning* 4, 2 (2002).
- WATSON, R. A., FICICI, S. G., AND POLLACK, J. B. 1999. Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC-99)*.
- WILTRONICS. 2003. Wiltronics Research Pty Ltd. <http://www.wiltronics.com.au>.
- YANCO, H. 2002. 91.450: Robotics I. <http://www.cs.uml.edu/~holly/91.450/>.

Received May 2004; revised May 2005; accepted June 2005.