Using RoboCup in University-level Computer Science Education

ELIZABETH SKLAR
Columbia University
SIMON PARSONS
Brooklyn College, City University of New York
and
PETER STONE
The University of Texas at Austin

Team-based projects have been proven in the education literature to be an effective pedagogical methodology. As a result, we have been using *RoboCup* challenges as the basis for class projects in undergraduate and masters level courses. This paper unifies 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 the related class projects in order to make the context of our investigation clear and to make it possible for others to replicate and extend our work as well as contribute to the shared resource.

Categories and Subject Descriptors: ... [**...**]: ...

General Terms: ...

Additional Key Words and Phrases: ...

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 around 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.

Author's Address: Elizabeth Sklar, Department of Computer Science, Columbia University, 1214 Amsterdam Avenue, New York, NY 10027, USA, `sklar@cs.columbia.edu`

RoboCup [robocup], initiated in 1997, was designed to bring together robotics and artificial intelligence researchers world-wide by providing a common problem, a 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* — which vary in physical size, cost, type of hardware platform and approaches to vision and software control. This remit was later expanded to include robotic urban search and rescue as well. In 2000, the RoboCupJunior 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].

In this paper, we document our experiences incorporating RoboCup activities into primarily undergraduate courses with the goal of uniting others who are doing the same. Our aim is 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 and Multiagent Systems to be a natural one, we presume that this sense arises out of our familiarity with RoboCup through longterm 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]. 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 understanding issues such as real-time process and socket programming, topics outside of 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 has been developed to ease students into the use of agents for the RoboCup soccer simulator [Heintz 2000]. Andreas Birk has developed a course on Autonomous Systems that uses the Small-Size RoboCup League for practical exercises[Birk]. José Vidal and Paul Buhler [Vidal and Buhler 2002] have developed a series of graduate-level courses on multiagent systems using the RoboCup Simulation league. They have 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 course (6.270), following from Woody Flowers' renowned "Introduction to Design" (2.70) course that was offered in the Mechanical Engineering department [Martin]. In Martin's course, students learn about the basics of building robots from kits and the term culminates with a contest. The work on this course resulted in a now popular textbook [Martin 2000]. Holly Yanco [Yanco] has adopted this course using the Botball[botball] game as the tournament at the end of the term. Maja Matarić has developed an award-winning course called "Introduction to Robotics" [Mataric] which takes a hands-on approach to teaching the fundamental concepts in the field of robotics. Students use both the Handyboard [handyboard] microcontroller and the LEGO Mindstorms Invention System [mindstorms]. 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 on "Programming Under Uncertainty" [Littman], offered in Fall 1999, taught a variety of methods for programming, as its title says, under uncertainty, including Markov Decision Processes and POMDP's, and a variety of machine learning techniques like reinforcement learning and genetic algorithms. Students in this course used LEGO robots to demonstrate their knowledge of the methodologies studied. The course ended in a project, where some of the students developed their own applications for their robots, 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 scientific study of the RoboCup learning environment. Following on the work of Sklar *et al.* [Sklar et al. 2002; **?**], which examines 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 paper is organized as follows. We begin by describing three courses, offered by each of the authors in different universities in the USA over the past four academic years. We outline the syllabus of each course, share reading lists, and project descriptions. We also present results of course evaluations of each course. Then we analyze the parallels between the coursework and evaluation results. We conclude with discussion of current and future work.

2. INTRODUCTION TO ROBOTICS.

This introductory course looks at robotics from several aspects: technically, historically and socially. Many of the technical aspects are based on Matarić's course mentioned above. The course was designed for non-engineering students to gain hands-on experiences with technology, as well as basic understanding of the field of robotics and challenges facing the field today. Part of the course is spent reading and discussing classic material that relates to robots — including non-technical literature such as science fiction, psychology, cognitive science and education. The remainder of the course takes a hands-on approach to introducing the basic concepts in robotics, focusing on autonomous mobile robots. LEGO

—Introduction
—LEGO Mindstorms construction and applications
—LEGO Mindstorms programming (Not-Quite C)
—Motors, Effectors and Actuators
—Sensors
—Control
—Behavior
—Learning
—Robotics in literature: Science Fiction

Fig. 1. Topics covered on the Introduction to Robotics course

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 RoboCupJunior rescue task), see Figure 3 (a). Second, they construct robots to play soccer and perform in a RoboCupJunior style two-on-two tournament (Figure 3 (b)).

2.1 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 Figure 1.

The course has been taught over a 14-week semester. The format involves one 75-minute lecture and one 75-minute lab per week. There was a midterm and a final exam. Students were 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.

Sklar taught this course in Spring 2001 at Boston College [Sklar]. Twenty-seven 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 and there is no engineering school in the university, so the hands-on technical experience of these students was limited. Sixteen 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

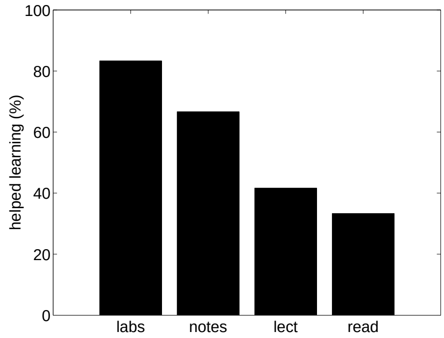
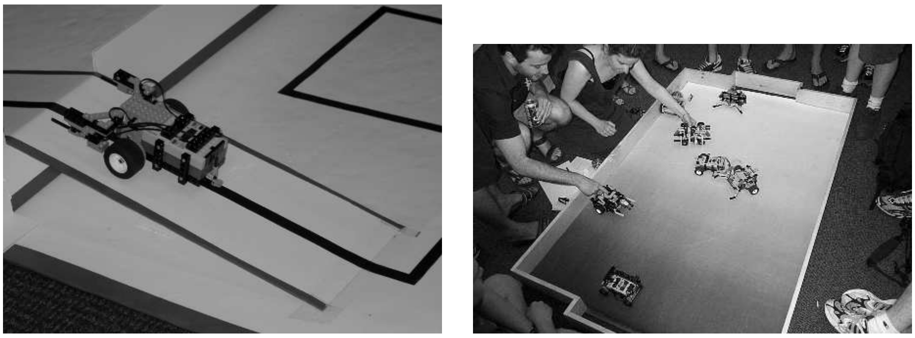


Fig. 2.
Elements of the course which students identified as helpful to them in learning the material
(Survey Results from Introduction to Robotics course, Spring 2001)



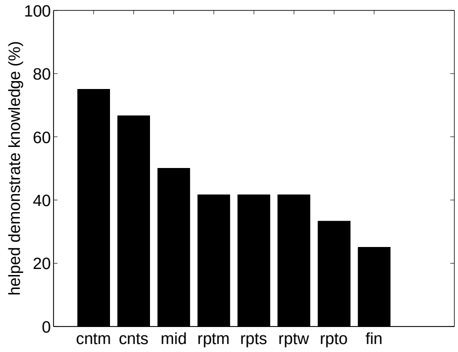
(a) Line-following maze (b) Soccer

Fig. 3. Robot contests.

the experience levels of the class was so diverse, Sklar assigned the groups, attempting to balance each group with an equal number of beginning and advanced students. Students were given some lab time during the scheduled course period in order to work on the projects. However, this was not enough time to perfect robots to perform well in the contests, so many of the students met outside of class time to work on the robots. Each student was required to submit a lab report individually, which included an assessment of the contribution of their teammates. The efforts of team members are never balanced, however most of the inequities were obvious in reviewing the lab reports, even without the peer assessment component. Students' grades were based on the lab reports, not on their robots' performance in the contests.

The two contests were held in a public space and students were encouraged to invite their friends to come and watch. Other faculty members also observed. 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 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



key:
cntm = maze contest
cnts = soccer contest
mid = midterm exam
rptm = maze lab report
rpts = soccer lab report
rptw = research report, written part
rpto = research report, oral part
fin = final exam

Fig. 4.
Elements of the course which students identified as helpful to them in demonstrating their knowledge of the material. (Survey Results from Introduction to Robotics course, Spring 2001.)

submit a brief project proposal several weeks prior to the final due date, in order to get them started and also to provide feedback about the appropriate nature of the topic. The range of topics chosen 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 ten-minute oral presentation on their 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 and many students in fact did not attend class on presentation days when they were not speaking. Course evaluation results (below) confirmed that the motivation surrounding the research project was minimal.

2.2 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 which elements of the course were helpful in learning the material and which elements of the assessment were valuable in helping them to solidify and demonstrate their knowledge of the subject. The results are shown in Figures 2 and 4.

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% and 67% responded that the two contests (maze and soccer, respectively) 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 chosen were perhaps too advanced for most of the class.

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 also learned more themselves. Negative remarks centered around requests for more lab time and less time spent on oral presentations.

3. ARTIFICIAL INTELLIGENCE.

The modern view of Artificial Intelligence (AI) [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 metaphor of intelligent agents 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 behavioral-based reactive techniques. Robots are prototypical agents that have to move around and react to their environment in pursuit of their goals. Indeed, it is hard to think of something that embodies the qualities that are required of intelligent agents better than robots do. As a result, 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. That course was built around Nilsson's "Artificial Intelligence: A new synthesis", which is not only agent-centric, but also covers reactive control architectures in detail early on. As a result, it is easy to incorporate material on approaches like the subsumption architecture [Brooks 1991] that provide a route to using AI methods in the robotics work even on a small platform like the LEGO Mindstorms. While this course was constrained by timetabling to be mainly lecture-based[1], 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 contest).

The same basic pattern has been used by Parsons to teach an introductory AI class at Brooklyn College of the City University of New York (CUNY) every semester since. Courses were run for undergraduates in Fall 2002, Spring 2003, and Fall 2003 and for Master's students in Spring 2003, Fall 2003 and Spring 2004[2]. In Spring 2003, the graduate course used the same robot simulator as described in Section 3.3 while all other iterations have used the same LEGO robots as in the robotics course. The enrollments for the various offerings are summarized in Table I. For each class we give the enrollment, the number of female students, the number of surveys we collected, the number of Computer Science and Information Science majors, and the number of graduate and undergraduate students. 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 pre-requisites that students have taken courses in programming and datastructures. Most students who take the courses have also covered discrete structures, which means that they already have met 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

[1] 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.

[2] The Fall 2003 graduate course was a slightly more advanced course.

Table I. Demographics for offerings of the introductory AI class.

Offering	Enrollment	Female	Surveys	CIS majors	Undergrad	Grad
S 2002 (U)	35	6 (17%)	19	14 (74%)	13 (68%)	6
F 2002 (U)	18	7 (39%)	18	16 (89%)	18 (100%)	0
S 2003 (U)	9	1 (11%)	6	5 (83%)	6 (100%)	0
S 2003 (G)	19	5 (26%)	13	12 (92%)	0 (0%)	13
F 2003 (U)	19	3 (16%)	15	13 (87%)	15 (100%)	0
F 2003 (G)	21	7 (33%)	16	16 (100%)	0 (0%)	16
S 2004 (G)	15	3 (20%)	15	15 (100%)	1 (7%)	14
Total	136	32	102	91	53	49

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

Fig. 5. Topics covered on the Artificial Intelligence courses

science majors, though we typically have one or two students a year who are majoring in different subjects.

The course has steadily evolved. The syllabus has been modified, is now largely based around Russell and Norvig's "Artificial Intelligence: A modern approach" [Russell and Norvig 1995], but still includes the material on reactive control mentioned above. We now provide many more resources to the students at the start of the project and, early on, organize one or two supervised lab sessions during which students can gain initial experience with the robots having instructors on hand to prevent them making common errors. On several occasions these instructors have included past graduates of the course, who seem very keen to pass on their expertise. We have also refined the contests—these refinements will be explained in detail below.

3.1 Course Structure

The current version of the course is taught over 14-weeks, structured as two 75 minute lectures a week, either delivered on two separate days (for daytime courses) or once a week one directly after other (for an evening course). In addition to the project work, the course features two exams and homeworks. The way the various offerings have been structured are best described by the material on the course web-pages [Parsons b; a]. These give the detailed syllabus for the various offerings, all of which use a subset of topics outlined in Figure 5. We have developed curricular modules for each of these topics and vary which modules we include in each offering. The web-pages also give the course schedule, homework, additional readings, lecture notes, and the detail of the robotics projects. Here we describe the organization of the projects, concentrating on aspects that are less obvious from the web site.

The projects are group efforts and account for 25% of the term grade. The projects start in around the third week of the semester, and each project runs for 5 to 6 weeks. Every student has to write a report on both projects. We find that the optimum group size is two students (more and it is easy for one to become a passenger, or to get shut out), though 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:

—Around 200 LEGO parts from which to build their robot (a subset of the 700 or so pieces that come as part of the standard LEGO Mindstorms kit), including the RCX (the Mindstorms microcontroller), two motors, and several sensors; and

—A set of robot designs and sample code to run on those robots.

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

The robot designs and sample code given to the students are taken from [Baum 2000], a book that is thoroughly recommended to anyone thinking of using the LEGO Mindstorms, and are chosen to illustrate the various issues that the students will have to deal with when participating in the contests. The set of parts is sufficient to build all the robot designs, though none of these designs will suffice alone for the challenges. The idea is to try and balance giving the students some help with the mechanical design problem, since this is the aspect of the project least connected to the subject matter of the AI course, without solving the problem for them. 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 because:

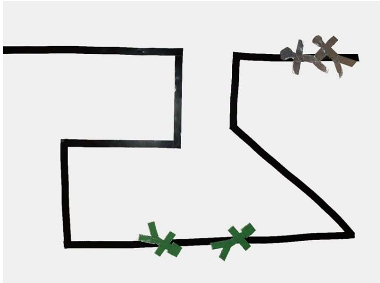
—students will have to carry the pieces to and from campus (because there is nowhere the robots can be stored on campus), and reducing the kit to something that will fit in a medium-sized plastic food-storage box reduces the burden, especially for students who commute over an hour on the subway as many of the students at CUNY do; and

—pieces will get lost, and keeping some pieces in reserve (especially some of the smaller parts) makes it possible to absorb these losses without having to purchase new kits.

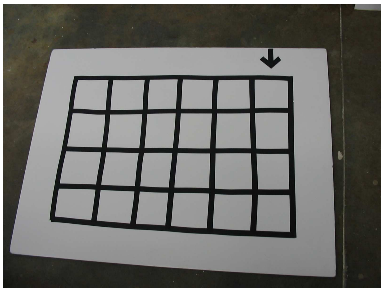
Our experience has shown that this reduced set of parts does not greatly restrict the range of designs that our students manage to create.

We are in the process of fully documenting sample robot designs, programs, and a selection of different sets of parts appropriate for different projects. These include programs to illustrate specific agent architectures—such as the subsumption [Brooks 1991] and the belief/desire/intention [Bratman et al. 1988] architectures, both of which have been implemented using the materials from the course—and robot designs that are more flexible than those given in [Baum 2000]. As we complete this work, it will be made freely available from `http://agents.cs.columbia.edu/er/`.

The contests in which the students take part are designed to test the full range of what is possible with the Mindstorms, at least using the sensors it is supplied with (albeit with the addition of an extra light sensor for the second contest), and the use of the Not Quite C (NQC) language.



(a) Maze without a gradient



(b) Gridworld

Fig. 6. Courses for other contests.

The first challenge we have been using is the same line-following task as in the robotics course. This is intended to be introductory, and therefore relatively straightforward. The main task, in the style of RoboCupJunior rescue, is to follow a black line on a white background, around a number of curves (both left-hand and right-hand). Sometimes the contest has involved climbing and descending a gradient (as in Figure 3 (a)) in order to make the task harder, but more recently we have moved toward using a flat maze as in Figure 6 (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 another wins, and to add a little complexity there are three further tasks. The first of these additional tasks is that the robot has to detect an obstacle using touch sensors—once it indicates it has detected this (by backing up), the obstacle is removed. The second additional task is to recognize colored areas in the middle of the line and to indicate they have been found by playing a tune. Since these have to be detected using the same sensors as detect the line, and give a reading between that of the black line and the white background, they introduce a trade-off between speed and accuracy of detection—failure to detect the areas results in a time penalty as does detection of areas that aren't there. The final additional task is to detect colored areas at the end of the course (which are of another color completely) and stop when they are reached.

Part of the reason the first challenge is simple is that the world the robot has to operate in is static. In contrast the second challenge we have used 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 used in Sklar's course [Lund and Pagliarini 1998][3]. The field for this game is shown in Figure 3 (b). It is carpeted with a grayscale, to give some 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. The project groups are split into two leagues and each league plays a round-robin of 3 minute games (which seem to be long enough to get a result fairly often, but short enough that the entire competition can be fitted into a class) followed by a final between the winners of the round-robins.

The problem with the soccer project is that it does not fit particularly well with an AI course. The instructions for the line-following project make it an exploration of reactive behavior, and the soccer project, while fun for the students and easily within their reach,

[3]For the Columbia offering of the course, the contest was to have a robot score in an open goal against the clock.

does not necessarily involve any additional AI techniques. As a result, in the most recent offering, we have experimented with a new challenge.

The students are confronted with a gridworld delineated with black lines like those in the first contest, Figure 6 (b), where some of the squares contain the same colored figures as in the first contest. The challenge is to survey the grid, identifying the positions of the figures, and then re-position the robot (at the arrow in Figure 6 (b)) and move to the figures in a pre-specified order in the lowest possible time. The idea behind the challenge is to bring in some of the concepts related to *search* that the students have covered in the course, combining these with the reactive techniques from the first contest (which are still required to move around the grid). Since the robots cannot localize, this is a hard challenge, but it is within the capabilities of the more able students. On occasion we have again offered students the option of an extra-credit project of building a dancing robot exactly as in the RoboCupJunior dance competition. Again this is generally great fun for the students but does not involve much AI to solve (if any).

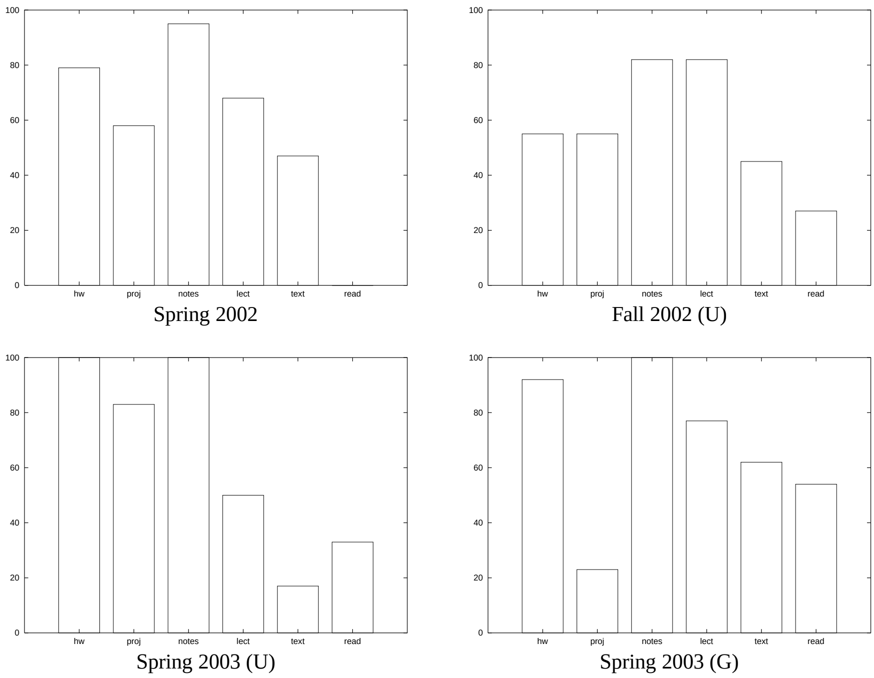
The combination of kits, challenges and language seem appropriate for the students who take these courses. It is possible, though, to go beyond what we are doing in terms of sophistication while still using the Mindstorms platform. It is possible to extend the kinds of tasks students can address by purchasing different sensors. 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] and even a compass [wiltronics]. It is also possible to extend the range of tasks by using a more sophisticated programming language. BrickOS [Baum et al. 2000; brickos] provides a more complete version of C than NQC, and one can program the Mindstorms in Java [Laverde et al. 2002] and Lisp [Klassner 2002; Klassner and Anderson 2003].

This latter work extends the range of tasks that can be solved by allowing the control programs to be run off-board (with instructions sent to the RCX using its IR communication port). A similar set-up has been used by Sklar for the RoboCupJunior E-league [**?**; eleague]. The main limitation that this approach overcomes is the 32K RAM with which the RCX is fitted. Another alternative, which has the advantage of keeping the computation on-board and the robot fully autonomous, is to run the control program on a Palm Pilot and interface that to the RCX [**?**].

3.2 Evaluation

For the Columbia offering of the course, two course evaluations were administered. One was an official evaluation done by the engineering school. The other was an informal paper-and-pencil survey given out in class, asking broadly the same questions as that given to the 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 they undertook a rating of 5 (on a 5-point scale) for interest, and two-thirds gave it a rating of 4 or 5. 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' perception of the value of the project relative to other aspects of the course. In particular, students were asked to identify which aspects of the course most contributed to helping them learn the material, and which aspects were most helpful to them in demonstrating knowledge of the material. This same informal survey was administered to students on the subsequent six offerings of the course, and overall gives us a considerable amount of data on the way that the projects are received.



key:

hw	= homework	proj	= robot project	notes	= lecture notes
lect	= lectures	text	= textbook	read	= additional readings

Fig. 7. Elements of the course which students identified as helpful to them in learning the material, Part 1

The results of the survey are given in Figures 7–10, along with combined results from all the surveys.

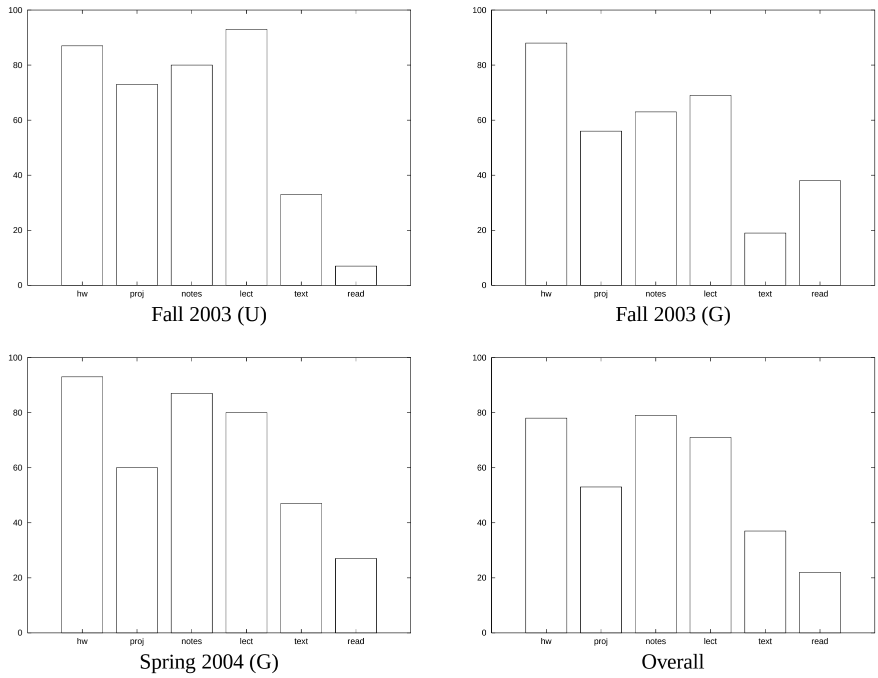
These results show fairly consistently that the project was felt to be less helpful in learning than more traditional elements like lectures, lecture notes and homework but is more helpful than additional readings or the textbook. In terms of demonstrating knowledge, the students felt that the project 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 are those for the Spring 2003 graduate course. For that offering the projects were performed using simulated robots (because of a lack of the LEGO kits). It is tempting to take the much lower figures as reflecting the fact that programming simulated robots is less satisfying than programming the real thing, but it might equally well reflect the fact that the simulator[4] presents a more challenging programming environment than the RCX[5].

Overall we think these results, and the similar results from other offerings of the course, are convincing enough to warrant continued use of robotics. Maybe more encouraging

[4]The RoboCup Soccer Simulator [simulator].

[5]Verbal feedback from the students suggested that those who mastered the simulator were very happy with the project.



key:

hw	= homework	proj	= robot project	notes	= lecture notes
lect	= lectures	text	= textbook	read	= additional readings

Fig. 8. Elements of the course which students identified as helpful to them in learning the material, Part 2

than the survey responses, though, are the free-form comments we have received, which include:

—"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."

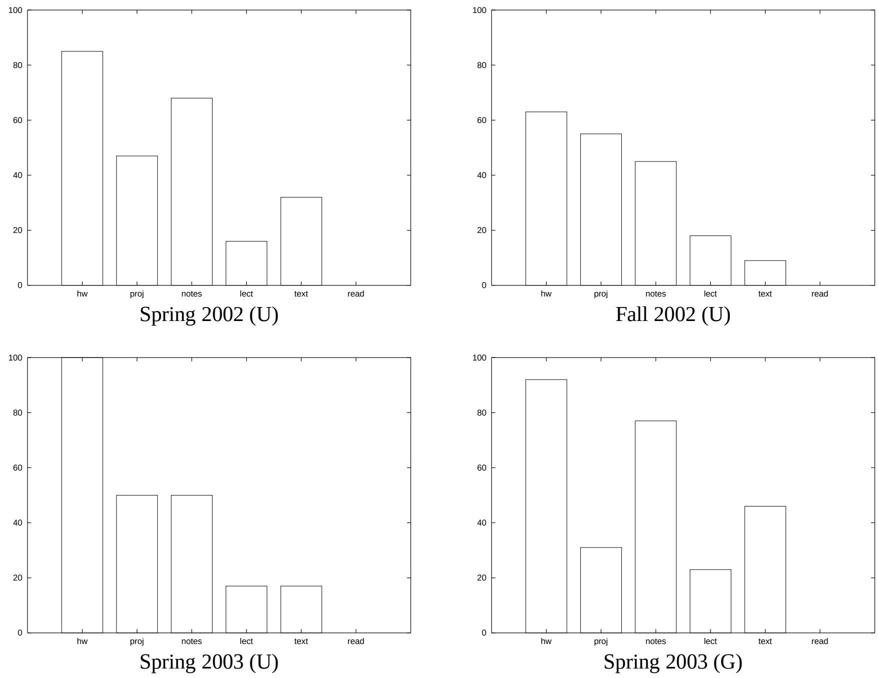
—"It helped immensely! It helped me understand some of the concepts covered in the lecture."

—"[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."

In addition, in the most recent undergraduate offering of the course, 14 of the 15 respondents used these free-form comments to indicate that the use of robotics had been beneficial, and that the project work had been fun.

It should be noted, however, that it seems that some of the positive reaction is likely because it is so unusual for students at Brooklyn College to get to do project work (in the open comment part of the survey several confessed that this was the only project they had done as undergraduates). The effect of this influence is supported by the fact that broadly similar results were generated by students who opted for a non-robotics project



key:

hw	= homework	cnt	= robot contest	mid	= midterm exam
rpt	= project report	fin	= final exam		

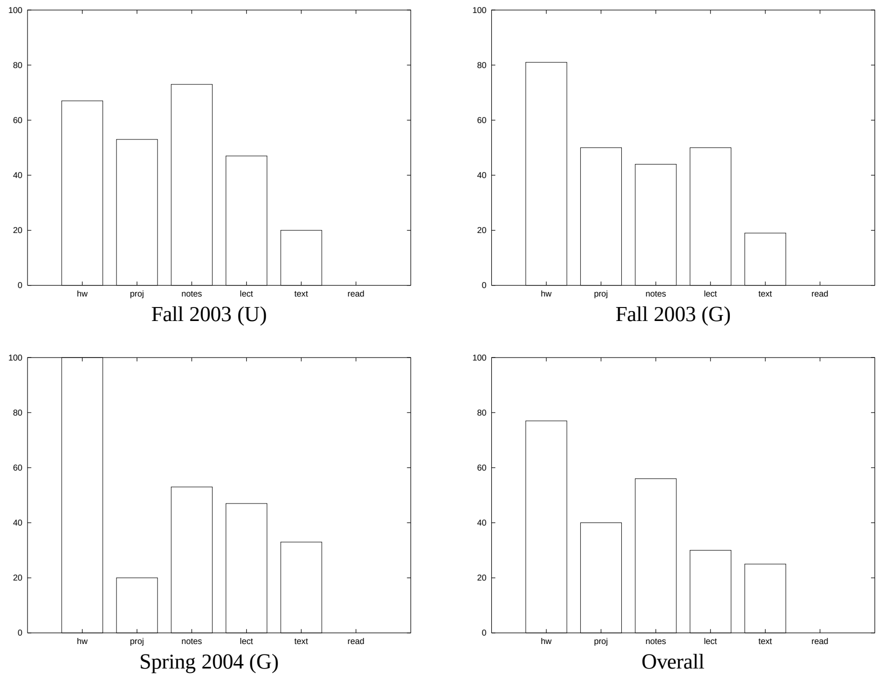
Fig. 9. Elements of the course which students identified as helpful to them in demonstrating their knowledge of the material, Part 1.

when such projects were offered in Fall 2002 (though the very small number of students in this category makes the results extremely unreliable and so they are not presented here).

3.3 Autonomous Multiagent Systems

Autonomous agents and multiagent systems (AAMAS) is one of the fields to which RoboCup participants have contributed consistently and prominently over the years. Despite being the basis for a large subfield of AI, there is no generally accepted definition of artificial intelligence *agents*. In loose terms, agents are programs that (i) sense their environment, (ii) make decisions about how to act based on these sensations, and (iii) then execute these actions. Autonomous agents do all three of these steps on their own, i.e., without a human in the loop. Multiagent systems are collections of multiple agents that interact with one another. The field of AAMAS 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].

This course provides a broad introduction to autonomous agents with an emphasis on multiagent systems. Topics include agent architectures, inter-agent communication, agent teamwork, distributed rational decision making, agent modeling, multiagent learning and entertainment agents. A full syllabus is given in Figure 11. The formal pre-requisites for



key:

hw	= homework	cnt	= robot contest	mid	= midterm exam
rpt	= project report	fin	= final exam		

Fig. 10. Elements of the course which students identified as helpful to them in demonstrating their knowledge of the material, Part 2.

—Introduction
—Autonomous agents
—Agent architectures
—Multiagent systems
—Agent communication
—Teamwork
—Robocup case studies
—Swarms and self-organization
—Applications
—Game theory
—Distributed rational decision-making
—Auctions
—Agent modeling
—Multiagent learning
—Entertainment agents

Fig. 11. Topics covered on the Autonomous Multiagent Systems course

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 about 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 these activities and the activities of a typical undergraduate student [Stone 2004]. As such, 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 attain a mastery of the AAMAS subject. However assessment is based primarily on their ability to engage in the full range of activities required of researchers.

3.4 Course Structure.

The central focus of the course is a semester-long build-up towards a class simulated robotic soccer competition in the RoboCup Soccer Server [Noda et al. 1998]. Students are assigned a series of four preliminary programming assignments designed to get them familiar with Soccer Server and some client code [Noda and Stone 2003]. By the end of these preliminary assignments, they have created a fully functional team (although not one that is particularly competent). The students are then encouraged to propose an improvement on this team as the topic for their final projects[6]. For example, one student proposed to use machine learning techniques to train a good goaltender without paying attention to the rest of the team.

The majority of the readings for this course are primary sources chosen both to 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 to submit a brief written answer to a single question pertaining to the readings the night before class. The fact that the responses are due before class allows the instructor to incorporate them into the class discussion. Another effect of the questions is that the students *do* come to class prepared to discuss the readings. As a result there have been many extended and heated class discussions.

An important component of class participation is that each student is required to moderate at least one class discussion pertaining to that week's readings. They are 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 for the students to complete. Many of them are not used to speaking in front of a class, and they 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 for their projects as well as the proposed methods for achieving them. Second, they revise their proposals and add a section on their work in progress to create progress reports. Finally, they write final reports in the format of conference papers.

Students optionally work in pairs on the final project. Teams must write their proposals and reports individually, with clear indications of what role each person played in the collaboration; and as such, 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 up the work.

The class culminates in a simulated soccer tournament. The students are told at the

[6] 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.

outset that performance in the tournament will have no negative impact on their grades (while a strong performance *can* have a positive impact). Nonetheless, the tournament is a strong motivational 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 given the fact that some students do not focus on creating winning teams. All class champions have been tested against a mid-range RoboCup entry and lost significantly: despite starting with a fairly detailed client code base, the students are not able to attain competitive world-class levels. However, given their time limitations this fact is neither surprising nor discouraging.

3.5 Evaluation.

Stone has taught this course three times, first at New York University in the Fall of 2001. Fifteen students were enrolled, only one of whom was female. All students were Computer Science majors. All were graduate students: 12 masters and 3 Ph.D. The second offering of the course was at the University of Texas at Austin during Fall of 2002 [Stone a]. Again, fifteen students were enrolled, however none were female. This time, the cohort were undergraduates. Fourteen 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 the Spring of 2004 [Stone b]. This class had 24 undergraduate students, all Computer Sciences majors, including three females. Most of the students were seniors, though the class did include ** juniors and ** sophomores (second year).

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 have also provided evidence that the students appreciate the opportunity to be exposed to the various components of scientific research. All times the course has been run, at least one student has described the course in graduate school applications as a primary motivation for going on to do research. In addition, one student from the Fall 2002 offering actively contributed 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 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 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 have included:

—“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 diferent 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 first hand."

—"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."

4. 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. Collectively, the courses have been offered five times at five different universities, providing a broad range of backgrounds, demographics and experience levels — from first-year undergraduates at a private, non-engineering 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. Thus the positive feedback across the board in regard to the robotics projects is an encouraging and significant factor.

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 reading in short written assignments and were then given the opportunity to discuss the readings formally in class, fared better than the other courses, where reading was assigned in a more traditional manner — without written reading responses and primarily the material was presented by the instructors in lectures where questions were encouraged (but infrequent) and discussion was not the central theme. Comparatively, Stone's classes had 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 our colleagues in improving the existing courses as well as designing new ones.

The challenge presented to students who were required to make oral presentations to the classes (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 is 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 coursework and the general RoboCup learning environment.

5. SUMMARY.

We have presented three university-level courses which cover topics related to robotics in various ways and are particularly focused on RoboCup challenges. Evaluations, both qualitative and quantitative, have been conducted at the end of each instantiation of each of the courses described, which includes diverse populations from inner-city schools in the northeast to a private school in the southwest. In all cases, the inclusion of a RoboCup-style contest has been a highly motivating factor, both for fun and to help students assimilate, reaffirm, apply and demonstrate their knowledge of the curricular topics being covered.

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

One of our goals with this work 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 here an account of our collective experiences incorporating RoboCup activities into undergraduate courses. We hope to encourage others to join in this community venture. Our on-line repository can be found evolving at `http://agents.cs.columbia.edu/er`. We welcome contributions and participants.

REFERENCES

ARKIN, R. C. 1998. *Behavior-Based Robotics*. MIT Press.

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.

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. Autonomous systems. `http://www.faculty.iu-bremen.de/birk/lectures/COURSES/autosys.html`.

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.

botball. `http://www.botball.org`.

BRAITENBERG, V. 1984. *Vehicles: Experiments in Synthetic Psychology*. MIT Press.

BRATMAN, M. E., ISRAEL, D. J., AND POLLACK, M. E. 1988. Plans and resource bounded practical reasoning. *Computational Intelligence 4,* 349–355.

brickos. http://brickos.sourceforge.net/.

BROOKS, R. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation 2,* 1.

BROOKS, R. A. 1991. Intelligence without representation. *Artificial Intelligence 47*, 139–59.

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 Publishing.

CORADESCHI, S. 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.

eleague. http://agents.cs.columbia.edu/eleague/.

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*. ACM Press.

handyboard. http://www.handyboard.com.

HARVEY, I., HUSBANDS, P., AND CLIFF, D. 1992. Issues in Evolutionary Robotics, Cognitive Science Research Paper Serial No. CSRP 219. Tech. rep., The University of Sussex School of Cognitive and Computing Sciences, Falmer, Brighton, UK.

HEINTZ, F. 2000. RoboSoc, a System for Developing RoboCup Agents for Educational Use. M.S. thesis, Linköpings universitet, Department of Computer and Information Science.

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*. Northern, KY, 8–12.

KLASSNER, F. AND ANDERSON, S. 2003. LEGO MindStorms: Not just for K-12 anymore. *IEEE Robotics and Automation (in press)*.

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, M. Riedmiller, and C. Sammut, Eds. Springer Verlag, Berlin, 636–644.

LAVERDE, D., FERRARI, G., AND STUBER, J., Eds. 2002. *Programming Lego Mindstorms with Java*. Syngress.

LITTMAN, M. 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.

MARTIN, F. 6.270: The MIT LEGO Robot Design Project Competition. http://fredm.www.media.mit.edu/people/fredm/projects/6270/.

MARTIN, F. 1996. Kids Learning Engineering Science Using LEGO and the Programmable Brick. In *Proceedings of the Conference of the American Educational Research Association*.

MARTIN, F. 2000. *Robotic Explorations: A Hands-On Introduction to Engineering*. Prentice Hall.

MARTIN, F., MIKHAK, B., RESNICK, M., SILVERMAN, B., AND BERG, R. 2000. To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines. *Robots for Kids: Exploring New Technologies for Learning*.

MATARIC, M. CS 445 Introduction to Robotics, a LEGO-kit-based hands-on lab course. http://www-scf.usc.edu/˜csci445/.

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 Publishers.

MATARIC, M. J. 1997. Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior. *Journal of Experimental and Theoretical Artificial Intelligence, special issue on Software Architectures for Physical Agents 9,* 2-3.

MCKERROW, P. J. 1991. *Introduction to Robotics*. Addison-Wesley Publishing Co.

mindsensors. http://www.mindsensors.com.

mindstorms. http://www.legomindstorms.com.

MINSKY, M. 1987. *Society of Mind*. Picador.

NODA, I., MATSUBARA, H., HIRAKI, K., AND FRANK, I. 1998. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence 12*, 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). To appear.

nqc. http://bricxcc.sourceforge.net/nqc/.

PARSONS, S. CIS 32/CIS 716: Artificial intelligence. http://www.sci.brooklyn.cuny.edu/~parsons/courses/.

PARSONS, S. COMS W4701: Artificial Intelligence. http://www.cs.columbia.edu/~sp/4701-2.html.

robocup. http://www.robocup.org.

RUSSELL, S. J. AND NORVIG, P. 1995. *Artificial Intelligence: a Modern Approach*. Prentice Hall.

SIMMONS, R. 1994. Structured control for autonomous robots. *IEEE Transactions on Robotics and Automation 10*, 1 (February), 34–43.

simulator. http://sserver.sourceforge.net/.

SKLAR, E. MC375: Introduction to Robotics. http://www.cs.columbia.edu/~sklar/teaching/spring2001/mc375/default.html.

SKLAR, E., EGUCHI, A., AND JOHNSON, J. 2002. Robocupjunior: learning with educational robotics. In *Proceedings of RoboCup-2002: Robot Soccer World Cup VI*.

SKLAR, E., JOHNSON, J., AND LUND, H. 2000. Children learning from team robotics: Robocup junior 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. CS378: Autonomous Agents and Multiagent Systems. http://www.cs.utexas.edu/~pstone/Courses/378fall02.

STONE, P. CS378: Autonomous Agents and Multiagent Systems. http://www.cs.utexas.edu/~pstone/Courses/378spring04.

STONE, P. 2004. RoboCup as an introduction to CS research. In *RoboCup-2003: Robot Soccer World Cup VII*, D. Polani, B. Browning, A. Bonarini, and K. Yoshida, 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.

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. http://www.wiltronics.com.au.

YANCO, H. 91.450: Robotics I. http://www.cs.uml.edu/~holly/91.450/.

Received May 2004; revised May 2005; accepted Month Year