Using Autonomous Robotics to Teach Science and Engineering

Randall D. Beer[1,2], Hillel J. Chiel[2,3] and Richard F. Drushel[2]

[1]Dept. of Computer Engineering and Science
[2]Dept. of Biology
[3]Dept. of Neuroscience
Case Western Reserve University
Cleveland, OH 44106
Email: beer@alpha.ces.cwru.edu
www: http://vorlon.ces.cwru.edu/~beer/

Appearing in the June, 1999 issue of *Communications of the ACM*

Please address all correspondence to:

Randall D. Beer
Dept. of Computer Engineering and Science
Case Western Reserve University
Cleveland, OH 44106

Introduction

If you walked into our Autonomous Robotics class[1] at Case Western Reserve University on a typical day, you might be surprised to find thirty college students from a variety of engineering and science disciplines sitting on the floor surrounded by LEGOs™. Are students being given college credit for playing with LEGOs? Appearances can be deceiving. In fact, this course tackles serious issues in engineering and science education. In this course, students design, build, program and test their own autonomous robots that participate in a public competition. This course uses robotics to foster a hands-on, interdisciplinary, teamwork-oriented approach to the synthesis and analysis of integrated real-world systems, as well as teaching new approaches to robot control.

Created in 1995, our Autonomous Robotics course grew out of ongoing research on biologically-inspired robotics at CWRU (Beer et al., 1997). The design of this course draws heavily on technology developed at MIT, first for K-12 education (Resnick, 1993) and later for an undergraduate course similar to ours that has been running since 1990 (Martin, 1994). Related courses have been developed at the University of Edinburgh (Donnett and Smithers, 1991) and elsewhere. Two features of our course distinguish it from these other courses. First, our final competition is considerably more technically demanding. Second, we address a much broader set of educational goals. Our course attracts students from computer engineering and science, biology, electrical engineering, neuroscience, systems engineering, biomedical engineering, and physics. In this article, we describe the educational goals of the course, its overall design, the final competition, and student assessment.

Educational Goals

Why is the transition from student to professional difficult even for the best students? In the majority of programming that most computer science and computer engineering students do in

---

[1] Our course web page, as well as pointers to similar courses and related information on the web, can be found at http://eecs.cwru.edu/courses/lego375.

industry, software is only one component of a much larger integrated system. This software must be designed with these other components firmly in mind and through close collaborative interactions with the people responsible for these other components. For example, implementing embedded control software for an automobile's fuel injection system is very different from implementing a depth-first traversal of a binary tree in a data structures class. More generally, engineering and science students often find the transition from student to professional difficult for four major reasons: (1) students are not trained to deal with problems that require an integrated approach; (2) they are rarely exposed to the real-world issues that such problems pose; (3) they are rarely encouraged to solve problems through teamwork, or to bring to bear information from multiple disciplines; and (4) they rarely have an opportunity for critical thinking.

Integrated approaches are essential for solving problems in engineering and science. A tacit assumption of many engineering students is that if each piece of a complex project works in isolation, the complete system will work as a unified whole. This is almost never true in practice. Unexpected problems emerge unless one takes into account the special properties of each piece of the system, and their interactions. There is a growing need to deal with such problems. Increasingly, software is part of an embedded system such as an airplane, a microwave oven, or a copy machine. In such systems, it is essential to integrate (and trade off between) the design and implementation of mechanics, electronics, and control. Similarly, the education of biology students emphasizes the reductive analysis of animals into cells and molecules, but not how such components as sensory structures, musculature, and neural control are integrated into complete functioning biological systems. There is a growing awareness, however, that overall function emerges from the coupling of neural control, peripheral biomechanics, and the environment (Chiel and Beer, 1997).

Outside of the classroom, engineering and science problems share certain real-world characteristics. Unlike textbook problems, there may be no single right answer. Moreover, the world rather than a professor decides whether an engineering design or a scientific hypothesis are correct. One can almost never ignore resource limitations of time, money, and materials. Finally, real world devices never follow the ideal models. For example, a program that assumes that sensors always deliver exact and valid values and that motors always deliver the commanded speed and torque will be helpless in the face of noise, spurious inputs, unreliable outputs, or breakdowns. Students lacking hands-on experience significantly underestimate the importance of these real world issues.

Increasingly, progress on engineering and science problems is made by collaborative interdisciplinary teams, rather than by a single individual working in isolation. Designing an airplane requires team members with expertise in fluid dynamics, control software, mechanics, materials, and computer simulation. Similarly, understanding how an animal's nervous system produces a particular behavior requires team members with expertise in electrophysiology, molecular biology, biomechanics, behavior, and computer simulation. Unfortunately, current educational practice emphasizes individual problem solving within well-defined disciplines. As a consequence, students may not master the interpersonal skills necessary for effective teamwork, and they are unlikely to learn to overcome the language barriers that make interdisciplinary communication difficult.

Students often feel that education is something that is done to them, rather than something they are actively doing for themselves because they are not encouraged to think critically. Not only does this limit how well they learn, but it often prevents them from preparing for the lifelong learning they will need as professionals in engineering or science. Students are not entirely wrong to take this attitude. Current educational practice encourages passive listening in large lectures, rather than active engagement. It encourages finding the single answer that the professor wants, rather than creatively exploring multiple possibilities. It encourages rote retention and repetition, rather than critical analysis of material (Chiel, 1996).

Learning to Build a Robot

To address these educational problems, it is essential to identify subject matter that naturally and simultaneously engages all of these issues. This has determined our decision to teach a course in which teams of students from different disciplines design, build and program their own animal-like autonomous robots. At the end of the semester, students participate in a public competition in which teams of two robots collect and discriminate plastic eggs, depositing them in one of two nests distinguished by differently polarized light beacons.

The semester begins by dividing the students into groups of three. While we allow students to choose their own groups, some care must be taken in the composition of these groups because the course attracts students from such a variety of backgrounds. We try to ensure that at least one member of each group is a skilled programmer and that at least one member is mechanically adept. Each group is assigned a station consisting of a table and chairs, a computer, and a robot kit (Figure 1). The lab currently contains a mixture of desktop and portable Macintosh computers and can support up to 30 students per semester (10 groups of 3). Aside from group assignment, the major activity of the first day of class is to completely inventory each kit in order to verify that all materials are present.

Students receive a standard kit that includes all materials that will be used throughout the semester (Figure 2). In total, there are more than 1300 parts in each kit (total cost $400-$500). “Plug-and-play” modularity is essential to our educational philosophy, allowing students to easily combine these components in an almost endless variety of ways. Accordingly, the majority of these components are LEGO building materials, including a large variety of plates, beams, pins, axles, bushings, wheels, pulleys, gears, differentials, cams, propellers, and treads. NonLEGO building materials include springs, rubber bands and polarizing filters. The kit also includes several DC motors, a servomotor, incandescent lights, LEDs and a variety of light sensors, touch sensors, bend sensors, and breakbeam sensors. The motors and sensors are inexpensive off-the-shelf and surplus components that have been modified with LEGO mountings so that they are readily reusable and can seamlessly interface with the rest of the building materials.

One of the most important components in the kit is the 6.270 microcontroller board, developed at MIT (Martin, 1992; cost $300). This board is based on a Motorola 68HC11 microprocessor with 32K RAM. It supports up to 8 digital and up to 16 analog inputs, and can drive up to 6 DC motors, a servomotor, and 2 incandescent lamps or LEDs. Two pushbutton switches, four DIP switches and a variable resistor allow for user input, and a 16×2 character LCD display and a piezoelectric buzzer are provided for user output. The board is powered by rechargeable batteries, allowing for fully autonomous operation of student robots. The board is programmed in an interpreted subset of C known as Interactive C (IC) and contains a variety of libraries, including support for multi-tasking. Although assembled 6.270 boards are no longer commercially available, a similar microcontroller known as the Handy Board is available from a variety of vendors (URL: http://lcs.www.media.mit.edu/groups/el/projects/handy-board/).

During the first half of the semester, students perform a series of structured exercises that are designed to familiarize them with the components of the kit. They begin by learning to program the 6.270 board in IC and learning to build a variety of mechanical structures with LEGO. Their first design project is to build a motorized LEGO device that can locomote in some way other than using wheels or treads. In addition to gaining experience with building robust, fast and powerful geartrains using LEGO, this assignment gives students an opportunity to creatively explore the capabilities of the materials in the kit. Throughout these structured exercises, we provide a series of "mini-lectures" that provide necessary technical information, describe different approaches to robot control, and provide relevant biological background. For example, for the novel locomotion device assignment, we give an overview of the many different ways in which animals locomote (Alexander, 1992). This inspires students to try things they might not have otherwise thought possible, and has led to an amazing variety of LEGO devices that walk, hop, swim, flop, tumble, slither, or inch their way across the floor.

Students next undertake a series of exercises that expose them in turn to each of the specific design issues that are raised by the final egg hunt competition. First, they build and program a servo-mounted sensory platform that can orient to a polarized light source under different conditions. Second, they build and program a device that can discriminate between pastel and black plastic eggs. These sensory assignments raise important tradeoffs between mechanics, electronics and software control. For example, the software for both of these exercises can be very simple or very complex depending on the choice of light sensor and the design of its physical enclosure.

Third, students build a motorized platform that can keep moving while avoiding obstacles. In addition to providing further experience in LEGO geartrain design, this assignment begins to raise significant design choices. For example, there are many different ways of detecting obstacles (e.g., bump sensors, bend sensors, stall sensors) and each method has different performance and software complexity implications, which students must discover for themselves. Fourth, students combine their earlier light orientation device with their obstacle avoidance robot so that the robot can find and approach a polarized light. This assignment begins to raise significant integration issues in both mechanical and software design, as well as posing new problems (e.g., finding a light to orient to and approach). Finally, students incorporate their egg discriminator to produce a robot that can retrieve one pastel egg and deliver it to a nest. At this point, students have tackled all of the core problems raised by the final competition. However, their robots often perform quite poorly because the mechanical and software design of each component occurred largely in isolation from the rest. In fact, some students are unable to complete this final assignment because assumptions made in constructing an earlier component are simply incompatible with the operation of a later component. *This is intentional.* We have found that only by experiencing these difficulties for themselves do students fully appreciate the importance of integrative issues when designing a complete device.

At mid-semester, the structured exercises end and we hand out the complete rules for the egg hunt competition. The three-student groups are paired into teams that will field two robots in the final competition. Based on the performance of the individual groups in the first half of the semester, we try to balance the teams as much as possible. The first task of these new teams is to design a joint strategy for the final competition. This requires that the three-student groups, which have typically learned to work very well together, develop a new group dynamic within the larger team. Once these deliberations are complete, each team must formally present their design to us and defend it. While we rarely reject a well-thought-out design, we try to emphasize to students that simple and robust designs usually outperform elaborate and brittle ones in the final competition.

Once a design has been approved, students build their contest robots over a four-week period. Given that most students have already completed a prototype egg-hunt robot by mid-semester, this construction period might seem too long. However, based on what they have learned in the first half of the semester, students invariably want to build a new robot, changing some of their earlier design decisions and designing for a complete integrated system that fits their team's overall strategy. During the first half of the semester, students discover that poorly structured software is ineffective for managing the growing complexity of robot behavior. Therefore, students often put considerable effort into developing a software architecture that allows the many behavioral capabilities required of a contest robot to be smoothly integrated and tested. Some of the architectures that students have utilized include (1) a finite state machine architecture, (2) a multiprocessing architecture, in which each behavior runs as a separate process that communicates with other behaviors via shared memory, and (3) a subsumption architecture, in which higher-level behaviors can override or suppress lower-level behaviors (Brooks, 1991).

At the end of the semester, we hold a series of mock contests that closely simulate the conditions of the final competition. Participation in these mock competitions is required. Although students are encouraged to test their robots throughout the semester, this informal testing is often haphazard and ineffective. Robots are often alone in the arena, with students "feeding" eggs to them one at a time, manually pointing them in the general direction of a nest, and nudging the robots whenever they become stuck. In contrast, during the actual competition, there are four robots in the arena at once, the distribution of eggs is constantly changing, and students are not allowed to touch the robots during a round. Based on the often disappointing performance of the robots in these mock competitions, there is usually a flurry of debugging and redesign in the days leading up to the final competition.

The Egg Hunt Competition

The final competition serves as a capstone for the students' activities throughout the entire semester (Figure 3). It is therefore important that it reflect our educational goals. We wanted the contest to be rich and open-ended, so that many different strategies and designs are possible. We wanted a task that contained a mixture of cooperation and competition, and whose duration was long enough for these complexities to be manifest. We also wanted it to be technically challenging, so that students would be forced to work together to push the limits of their materials and their abilities, yet feasible within the limitations of the kit. In order to foster a collegial rather than a cut-throat atmosphere, we decided not to award any monetary prize to the winners. Since the competition is a public event, students are already highly motivated to field successful robots. Last, but not least, the contest should be fun for both students and spectators.

For all of these reasons, we decided to make the final competition a robot egg hunt. Teams of two robots compete to collect as many pastel plastic eggs as possible while avoiding plastic eggs painted flat black. The egg hunt occurs in a closed arena with nests at each end (Figure 4). The main floor of the arena is covered by a gray carpet, and the walls of the arena are painted flat black. The walls and floor of the nests are painted white. At the back of each nest is a polarized light beacon; one nest is vertically polarized and the other is horizontally polarized. At the front of each nest is a slight lip that prevents eggs from rolling in or out of the nest on their own.

At the beginning of a round, each team is randomly assigned a nest, the robots are placed in front of their home nests, and 40 pastel and 10 flat-black plastic eggs are uniformly distributed throughout the arena. Each round lasts for 10 minutes and students are not allowed to touch their robots during this time. At the end of a round, each team's score is calculated by counting the number of eggs in their home nest: *score* = (*# of pastel eggs*) - 4*(*# of black eggs*). The scoring is designed so that a robot that fails to discriminate between pastel and black eggs will receive a score of 0 on average. Ties are resolved by a sudden death playoff in which the first egg deposited in a nest determines the winner of the round. The overall competition is structured as a modified double-elimination tournament with initial seeds chosen by the instructors based on performance in the mock competitions.

Within these broad constraints, an incredible range of strategies and specializations are possible (Figure 5). The simplest strategy is for both robots on a team to collect pastel eggs and return them to their home nest, either one at a time or in groups. Or one robot on a team can collect black eggs and deposit them in their opponent's nest. Or collectors can switch which nest they go to depending on the type of egg they find. A variety of nest blocking strategies are also possible. For example, a robot on one team can try to block their opponent's nest, thereby preventing them from depositing any pastel eggs in it. Or they can block their own nest in order to prevent their opponents from depositing any black eggs in it. In this latter case, the egg collector on the team must circumvent the blocking, for example by lifting eggs over the wall. In addition, because the 6.270 board contains an onboard timer, robots can switch strategies partway through a round. For example, because the movements of the four robots tends to push eggs against the arena walls, some robots switch to wall-following midway through a round. At the beginning of each round, some students use the DIP switches on the 6.270 board to select different strategies depending on their opponent's strategy. Over the past few years, we have seen examples of all of these strategies, with varying degrees of success. With over 70 egg hunt robots constructed to date, we have never seen the same design twice.

In order to be successful, a typical egg collection robot must solve quite a range of difficult problems. Because these robots have no vision, finding an egg involves random or patterned search. Robots are often equipped with a scoop to increase the likelihood of encountering an egg. As a robot drives around, an egg will eventually roll into the discrimination area. The presence of an egg is usually signaled when it breaks the beam between an infrared emitter/detector pair. This often triggers a gate to close so that the egg is not lost. Once an egg has been detected, its color is determined by a reflectance sensor, often augmented with an incandescent light to achieve reliable discrimination. If the egg color is one the robot can handle, it must next scan for the polarized beacon from the appropriate nest. If the desired beacon cannot be found, the robot must either search for it or find the other nest beacon and move in the opposite direction. Often, robots will drive backwards to a nest to prevent the scoop from picking up additional eggs of unknown color. Due to errors in steering, robots must constantly correct their course as they approach the nest. Because the nest floor is white, robots usually employ another light sensor to determine when they are actually in the nest. Once there, the egg must be released and the robot must get out of the nest and begin to search for another egg. Throughout a round, a robot must use some form of obstacle detection and avoidance to prevent it from becoming entangled with other robots (including its teammate!). When dealing with obstacles, timeouts and randomization are extremely important to prevent ineffective behavioral loops.

Assessment

Given the unusual design of this course, assessing student performance is difficult. Because things can go wrong during the egg hunt through no fault of the students, we have chosen to focus our assessment on the design process itself rather than the performance of the final robot. Therefore, the primary basis for student assessment is a design notebook that each student is required to keep throughout the entire semester. This design notebook is intended to be a daily record of the student's activities within the course. It includes not only detailed mechanical sketches and program source listings of final designs, but also a complete record of the initial ideas and their subsequent testing and refinement. These design notebooks are augmented with a more subjective assessment of each student's participation in the activities of their group and team. Finally, each group participates in a videotaped exit interview in which each student must present and defend one aspect of the final robot's design.

Conclusion

The course has been extremely successful. To keep up with student demand, the course is offered in both the Fall and Spring semesters, yet there is still a waiting list each semester. The public egg hunt competition typically attracts over a hundred spectators, as well as local news media. To excite students about careers in engineering and science, our course has been central to several outreach activities. A module using these materials has been developed for a freshman engineering course at CWRU. Tours of the laboratory are regularly used to recruit high school juniors and seniors to CWRU. A summer program for high school science teachers introduces them to the use of these materials in their classrooms. Demonstrations of student robots to elementary school students has generated enormous enthusiasm for engineering and science. In order to give students the opportunity to explore more of their creative ideas, we are constantly trying to extend the technology available to them. For example, we are developing a radio communication interface for the 6.270 board that will allow two robots on an egg hunt team to send simple messages to one another.

At the outset, we identified four educational goals: integration, real-world issues, interdisciplinary teamwork, and critical thinking. Our autonomous robotics courses addresses all of these goals. Building a robot requires that students integrate control, electronic and mechanical systems into a working device, confronts them with the interactions between different subsystems, and affords them the opportunity to trade off between the different subsystems in constructing their robot. This course also encourages biology students to confront the issues involved in getting a physical agent to operate reliably in a realistic environment by giving them the opportunity to “build their own animal”. Thus, the course provides a unique perspective on the many problems that nervous systems actually solve.

The problem of building an autonomous robot also engages the issues of real-world problem solving, multidisciplinary teamwork, and creative and critical thinking. Building an actual robot, rather than programming a simulation, immediately confronts the student with the non-ideality of real-world devices, as well as providing immediate feedback about the success or failure of their ideas. By having students work in teams, the course encourages them to pool their individual expertise, allows them to specialize on specific subtasks, and gives them experience in developing the interpersonal skills to articulate and defend their views, but ultimately reach a consensus that is best for the group as a whole. Because the course attracts students from a variety of disciplines, students learn that very different perspectives can be helpful for solving a hard problem, and they are motivated to learn each other's language to break down disciplinary barriers. There are three ways in which students are encouraged to take ownership of their education in this course, and to think critically: first, the excitement of building a working device; second, the desire to do well in the public competition; and third, the recognition that there is no single correct solution, which encourages creativity.

In addition to these broad educational goals, the course also exposes students to exciting new research approaches in robotics and neurobiology. These range from biologically-inspired approaches to robotics (Beer et al., 1997) to computational and physical models of the neural basis of behavior in animals (Beer, 1990). Indeed, our primary incentive for developing the Autonomous Robotics course was the high degree of interest among undergraduate students in ongoing research in this area at CWRU. Thus, the course also attracts bright undergraduate students to pursue graduate study in these areas of research. Indeed, some of the same materials used in this course have also begun to be used in robotics and neurobiology research (Donnett and Smithers, 1991; Webb, 1993; Morse et al., 1998).

Acknowledgments

The initial development of this course was funded by a grant from the Howard Hughes Medical Institute, and recent offerings have been supported by funds from the Case Alumni Association and General Motors. We would like to thank Norm Rushforth, Lee White and Bob Edwards for their support of the course. In addition, we would like to thank Leslie Picardo, Alan Calvitti, Brian Yamauchi, and a dedicated staff of undergraduate "alumni" of the course for their invaluable volunteer help in setting up and running the course.

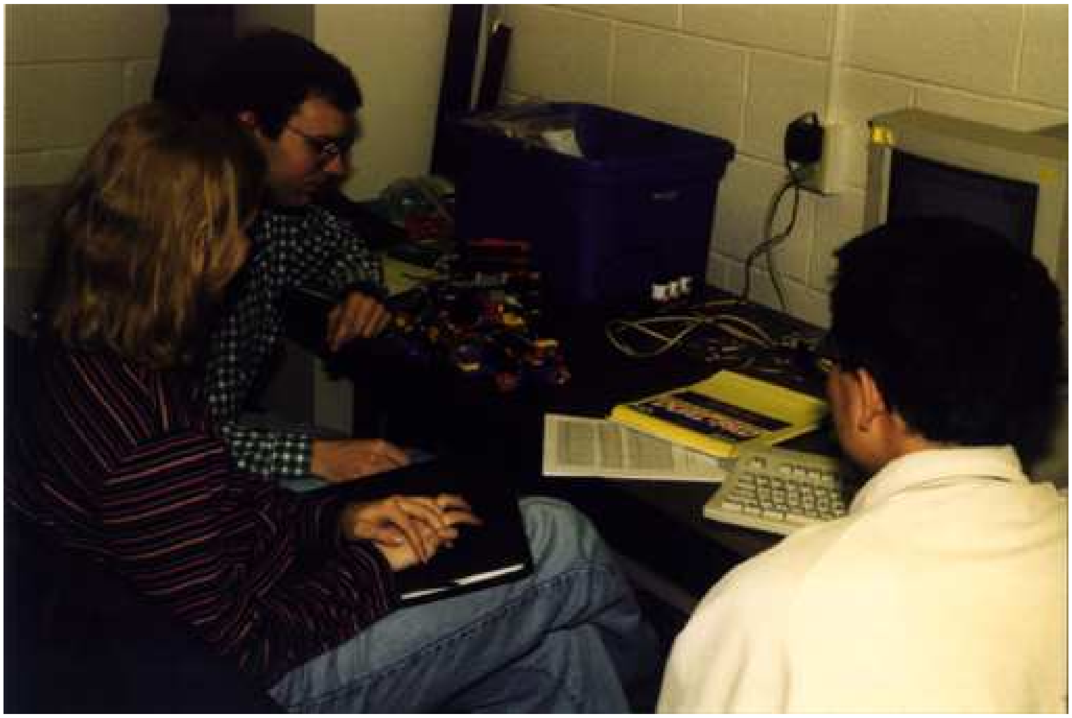

Figure 1: A Typical Student Station

Figure 2: Contents of the Robot Kit

Figure 3: Contest Day

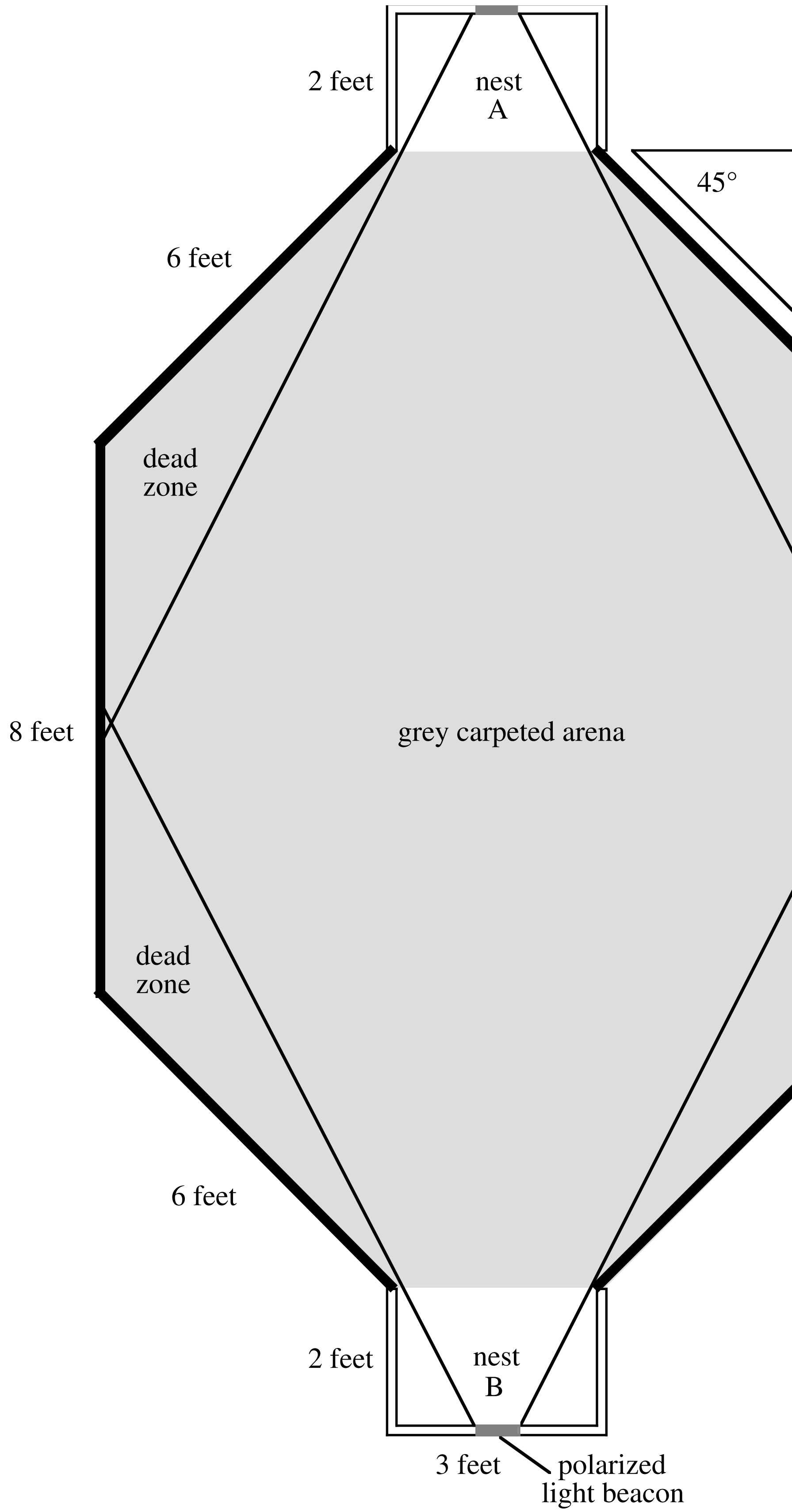

Figure 4: The Egg Hunt Arena

Figure 5: (Top) Student Robots. (Bottom) A close-up of one of the winning robots.

References

Alexander, R.M. (1992). *Exploring Biomechanics: Animals in Motion*. W.H. Freeman.

Beer, R.D. (1990). *Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology.* Academic Press.

Beer, R.D., Quinn, R.D., Chiel, H.J., and Ritzmann, R.E., (1997). Biologically-inspired approaches to robotics. *Communications of the ACM* **40**(3):30-38.

Brooks, R.A. (1991). New approaches to robotics. *Science* **253**:1227-1232.

Chiel, H.J. (1996). Critical thinking in a neurobiology course. *Bioscene* **22**(1):3-14.

Chiel, H.J. and Beer, R.D. (1997). The brain has a body: Adaptive behavior emerges from interactions of nervous system, body and environment. *Trends in Neurosciences* **20**: 553-557.

Donnett, J. and Smithers, T. (1991). Lego vehicles: A technology for studying intelligent systems. In J.-A. Meyer and S.W. Wilson (Eds.), *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior* (pp. 540-549). Cambridge, MA: MIT Press.

Martin, F.G. (1992). *The 6.270 Robot Builder's Guide*. Cambridge, Massachusetts: MIT Media Laboratory.

Martin, F.G. (1994). *Circuits to Control: Learning Engineering by Designing LEGO Robots.* Ph.D. Dissertation, Program in Media Arts and Sciences, Massachusetts Institute of Technology.

Morse, T.M., Ferrée, T.C. and Lockery, S.R. (1998). Robust spatial navigation in a robot inspired by chemotaxis in *C. elegans*. To appear in *Adaptive Behavior*.

Resnick, M. (1993). Behavior construction kits. *Communications of the ACM* **36**(7):64-71.

Webb, B. (1996). A cricket robot. *Scientific American* December: 94-99.