# CSbots: Design and Deployment of a Robot Designed for the CS1 Classroom

Tom Lauwers
Carnegie Mellon University
tlauwers@andrew.cmu.edu

Illah Nourbakhsh
Carnegie Mellon University
illah@cs.cmu.edu

Emily Hamner
Carnegie Mellon University
etf@andrew.cmu.edu

## ABSTRACT

We present CSbots, an ongoing program to use robots as educational tools in the Introduction to Computer Science (CS1) course. We aim to use robotics to improve learning and retention by altering course work so that it is more relevant to students.

In our development process we use an iterative cycle composed of design, pilot, and evaluation steps. We have completed the first of these cycles, the alpha cycle, and describe the robot hardware, software, and curriculum development processes as well as key evaluation results from pilots conducted at two community colleges in Fall 2007. We discuss the implications of these results and our experiences on the in-progress beta design cycle and planned pilots.

## Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer Science education, curriculum*

## General Terms

Design

## Keywords

Robotics, CS1, Iterative Design, Curriculum

## 1. INTRODUCTION

We present the intermediate results of a program, CSbots, which aims to develop a robotic platform and accompanying activities that are aligned with the needs of the Introduction to Computer Science (CS1) curriculum and student. Robots, as physically manifested computing devices, inherently show students how computing programs that they write can impact the real world. CSbots seeks to address issues of motivation and retention in the CS1 course by using robotic technologies as a way of increasing the real-world relevance of CS1 assignments.


*SIGCSE'09,* March 3–7, 2009, Chattanooga, Tennessee, USA.


Robots have a long history as educational tools. Used at all age levels, they are generally used to motivate student interest in further study in science and technology; many educator anecdotes and at least one large scale longitudinal study [1] support this notion. Currently, most educational robotics programs occur in out of school or informal settings; these include both competitions like FIRST [13] and Botball [9], as well as arts-based and cooperative programs like Artbotics [6], Crickets [10], and Robot Diaries [5].

In the context of steeply declining enrollments [12] a number of researchers are experimenting with robots to improve retention in Computer Science; notably in the first [4] and second [8] level courses. A major study of the use of Lego Mindstorms™robots as educational tools at the US Air Force academy [4] found that students using the robots did not show improved learning or retention. This last study enumerated a number of weaknesses to using robots in computing courses: Lego Mindstorms™robots are typically too expensive for student ownership, and so students must work on robot programming assignments in labs with limited hours. Feedback is delayed due to the real-time nature of robotics, causing students to devote more time to tedious debugging, and less to developing solutions. A new effort by the Institute for Personal Robotics in Education (IPRE) proposes to eliminate these weaknesses by providing every student with a personal robot [3] designed specifically for computer science education.

### 1.1 Approach

Our approach to creating the CSbots program rests on deep partnerships with educators working in the Computer Science field and evaluation-driven iteration of our robot and curricular design.

*Partnerships.* As our design team had limited experience teaching the CS1 class, we assigned great importance to engaging with CS1 educators. Our intention was to develop a participatory design partnership with a two-way sharing of domain knowledge and skill. Our partners were involved from the start of the design process, allowing them to test and comment on early versions of the robot and software, while we had access to their existing curricula and their crucial understanding of what works with students in the CS1 class. While the details of the robot design were left up to us, we worked together to design the interface of the software framework and the curricular activities. These partnerships took time to develop and maintain, and we are pleased to have found dedicated partners in the education community.

We conducted this participatory design process with two educators teaching CS1 at community colleges: one at the

Community College of Allegheny County (CCAC) in Pittsburgh, Pennsylvania, and the other at Ohlone Community College in Fremont, California. These educators became true design partners, suggesting changes to software and hardware, coming up with assignment ideas, and according us opportunities to pilot our jointly devised curriculum.

*Iteration.* We are engaging in a multi-year design process in which we will iterate through several cycles of the following steps:

**Design.** The design step requires engineering and educator design skills, involving the creation or revision of a robotic platform and associated software, and the development of a curriculum.

**Pilot.** In the pilot step researchers and educators jointly test the designed curriculum and technology in CS1 courses, with researchers providing necessary technical support and educators checking to ensure that the curriculum is working and making in-flight modifications if required.

**Evaluation.** Learning, motivation, and retention are tracked with exams, weekly student surveys, in-class observations, and comparisons of retention rates. These data are used to inform the next design step.

We have just completed and are presenting the results from the first of these cycles, the 'alpha' cycle, the goal of which was to determine what robot features are well-suited to the CS1 course learning goals and audience. We are currently working on the design step of the second, 'beta', cycle, using the lessons learned from the first cycle to develop a robot and curriculum that is ideally suited to the course. For the beta cycle we are working with 25 high school and community college educators to further develop and pilot the curriclum and robot platform.

## 2. INITIAL EVALUATION

To ensure that our designs were grounded in the realities of Computer Science education we engaged in extensive pre-design evaluations. These evaluations included a textbook survey that sought to discover unifying themes and learning objectives among CS1 classes as well as interviews with 37 educators at as many different institutions to characterize curricular design constraints and receptiveness to using robots as educational tools. Detailed information concerning the survey methods and results has been previously published [7], [11]. The initial evaluation of the CS1 class strongly influenced our first curricular and robot design; the following is a list of key conclusions from the evaluation and how those conclusions suggested certain design decisions:

- All of the textbooks surveyed focused on programming skills and not computing concepts, and most presented a modular approach, implying that a broadly applicable curriculum should be composed of unrelated modules covering specific programming concepts. Most textbooks ordered concepts in a similar manner, so it is possible for later modules dealing with more complex concepts to assume understanding of simpler concepts.
- A major curricular change is difficult to implement bureaucratically, in most cases requiring a majority decision from the CS department faculty. Tools should therefore be adaptable to the in-place curriculum.
- Few educators are planning to change the programming language they use in the next few years. Therefore a new educational tool or curriculum should not be tied to a new or relatively rare programming language, as switching languages is considered a major curricular change. Support for either Java or C++ is required for widespread adoption.
- Educators expect students to be able to work on their out-of-class assignments at home.
- Educators are more sensitive to student costs than to costs to a department, and so we do not believe that a CS1 robotics class should require students to purchase a robot unless it is in lieu of a similarly priced textbook.

In addition to the formal textbook and educator surveys, we worked with our partner educators to detail the learning objectives, instructional activities, and assessments in their specific courses. With our partners, we went through every exam, assignment, and prepared lecture used in the previous year, mapping learning goals at both a coarse, class-wide level, and at a fine-grained, assignment by assignment level.

## 3. DESIGN STEP

We used the results of our initial evaluation and analysis to simultaneously design a curriculum, robot platform, and software for the CS1 classes of our partner educators.

### 3.1 Curriculum

It was our goal to create a curriculum that would mirror our partners' prior CS1 class at both the macro and micro levels - the learning goals of the class were to remain the same, as would the complexity and learning goals of the individual weekly assignments. Reflecting this, the exams covered the same material at the same point in the course schedule. Keeping constant much else of the class allowed us to easily measure the effect of the robotics activities on the students. We focused on Java and chose a specific textbook, "'A Guide to Programming in Java"' [2], to link to our curriculum. We took care to ensure that each assignment developed was focused on teaching CS concepts and did not rely on robot-specific conceptual understanding.

### 3.2 Robot

The end goal of our technology development process is to create a robot platform and accompanying software that has the correct feature set to support the CS1 course and audience. From our previous evaluation and others' prior work [4],[3] we know that one core element of such a robot is that every student must be able to own or loan one for use at home. We are approaching this goal in two steps; during our alpha cycle we created and piloted a design that was maximally instrumented, so as to allow us to discover which features were appealing and useful in the class. In our beta cycle we are using our evaluation of the alpha cycle pilots to create the ideally instrumented, low-cost robot that is required for the course. A major trade-off of our two step strategy is that for the alpha pilots described in this paper, our robot was too high-cost to be loaned to students outside of class and lab hours.

The platform that we used in the alpha cycle is an iRobot Create[1] and a Qwerk controller[2]. Together, these two off-the-shelf components provide a rich set of features:

- A wireless tether, allowing the use of standard Java
- Bumpers for simple obstacle detection
- Position and velocity control

[1]http://www.irobot.com/sp.cfm?pageid=305
[2]http://www.charmedlabs.com/

• Vision via a USB webcam
• Audio, including the playing of audio wav files and the generation of speech from text
• An array of programmable LEDs
• Access to real-time RSS feeds, enabling programs to respond to internet events

## 3.3 Software API

The software environment we created was heavily influenced by the feedback we received from our partners as we were designing the environment. Thanks largely to this feedback special attention was directed at ensuring that all method calls to control the robot were relatively clear, such that reading the name of the method gives a proper indication of what it will do. Students were provided with both a full Javadoc style listing of all robot class methods, as well as with a quick reference that covered the most important methods. As an example, the method *saySomething*, which causes the robot to speak a *String* argument, works very much like *System.out.print*. For further exploration, the full software environment and documentation is available for download[3].

# 4. PILOT STEP

After completing our curriculum and technology in the early summer of 2007, we prepared for two pilots, one with each of our partner educators. As we knew that our robot platform was imperfectly designed for the course, these pilots served as a testbed to see what assignments and features appealed to students more than a validation of the use of robots in CS1. Although it is never possible to guarantee that the introduction of a new element to a course won't cause major problems in the course, we attempted to minimize this risk by testing two assignments in the summer course of our CCAC partner before proceeding with the full pilots.

## 4.1 Ohlone College Pilot

The Ohlone pilot was conducted from early September to mid-December 2007. Fifteen students signed up for the class, but only four stayed in the class with a passing grade; although low, this is not an unusual retention rate for many computer science classes at both Ohlone and CCAC. Although we attempted to run a number of assignments in the course, technical problems prevented the robots from working reliably. After most of the semester had run its course, we discovered the source of the problem - the on-campus wireless network was occasionally and at random intervals throttling the wireless signals from our robots. Though this problem prevented the pilot from producing any meaningful evaluation data, we learned an important lesson about robustness: That our robot could not be any more robust than the component technologies that compose it. In our new design we plan to use component technologies that have a higher degree of reliability.

## 4.2 CCAC Pilot

The CCAC pilot was held from late August through early December 2007. 72 students in four sections began the class; 23 completed with a passing grade. Three of the sections occured during the day, and consisted of bi-weekly lectures of two hours each, the fourth section was held in the evening for four hours once per week.

In addition to lecture times, students were able to access the robots for testing and demonstrating assignments. These labs were staffed by the principal researcher, and were open from noon to six on Wednesdays and Thursdays. The lab times were arranged with the students beforehand so as to minimize scheduling conflicts with work and other classes.

About a week before each assignment was handed out, the principal researcher and partner educator would meet to discuss the upcoming assignment. We would use an assignment from the previously developed curriculum as a starting point and decide whether it was appropriate given our experience with the students to that point. In this way, we were able to make real-time adjustments to our curriculum based on our conceptions of students' abilities and interests.

# 5. EVALUATION STEP

The CCAC pilot was formally evaluated along a number of metrics; we were especially interested in student motivation, retention, and learning. We tracked retention of students after every assignment and compared it to the retention of students in CS1 courses offered by the our partner from Fall 2003 to Fall 2006. Additionally, we tracked student interest in our assignments with short surveys that were completed after each assignment, as well as by tracking what percentage of students completed all assignments compared to previous years. To assess learning, we compared the overall grades of students in the pilot course to performance by students in the four previous fall semesters.

*Retention Rates.* We compared the retention rates of the fall 2007 course to courses taught by our partner in fall semester 2003-2006. We compared the overall retention rate, and found no significant difference between our pilot and prior years. We also examined the retention rate at the week-by-week level, by determining when a student last completed an assignment for non-zero credit, but also found no real differences between the pilot and prior years.

*Interest.* We measured student interest in the robot assignments during the pilot both by comparing their assignment completion rates to prior years and by directly asking them in post-assignment surveys. Student reports of the assignments via these surveys was generally favorable. Figure

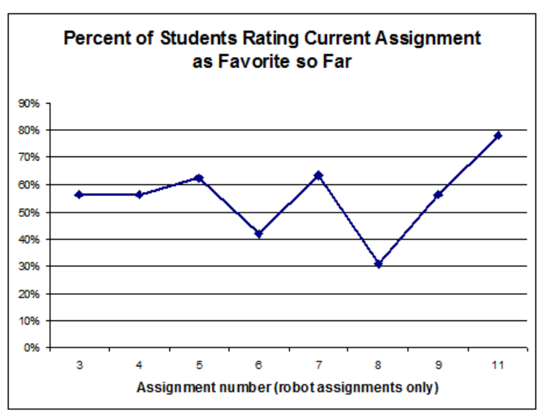


**Figure 1: Percent of students who listed the current assignment as their favorite to date**

[3]http://www.takemetoyourrobot.org/posts/tagged/csbots/

1 shows the percentage of students who, after completing an assignment, rated that assignment as their favorite assignment to that point in the class. One would expect that the trend line in this figure would fall over time, as students have more assignments to choose from. Instead we see a flat to marginally increasing trend line, indicating that students who did not drop the course stayed engaged throughout. We also compared student interest in the assignments to prior years by comparing the percentage of passing students who completed all of the assignments in the class. The grading structure both in the pilot and in prior years de-emphasized assignments, with each assignment making up only 2 to 2.5% of the total grade; as such, it is very easy to pass the class while missing an assignment. As figure 2 shows, the percent-

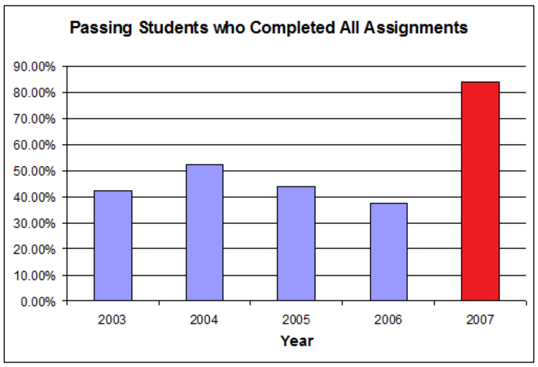


**Figure 2: Percent of passing students completing all course assignments**

age of students completing all assignments in our pilot was significantly higher than in prior years ($p<0.05$ for all years) despite the need for students to complete their assignments during limited lab hours in the pilot.

*Frustration.* We attempted to deduce potential 'sticking points' that would make the pilot assignments less engaging or more difficult by asking students to tell us the most difficult part of a just-completed assignment. We coded student responses into six categories; the percentage of expression of each is shown in figure 3. The codes were:

**conceptual** - Code for responses that gave a programming or CS concept as the answer.
**compile+syntax+IDE** - Code for responses where students had trouble with syntax, using the IDE, or figuring out compiler error messages.
**aesthetic** - Code for responses where the students responded that it was difficult thinking of non-programming creative elements; how the robot should dance, what it should say, etc.
**lab times+testing** - Code for responses dealing with the logistics of the lab setup - generally that it was difficult getting to the lab during the open hours, and that there weren't that many hours for testing.
**framework** - Code for responses dealing with usability issues in the robot software framework created for the pilot.
**nothing difficult** - Code for responses stating that nothing was difficult about the assignment.

Viewing figure 3, it becomes apparent that conceptual issues were often the most difficult part of an assignment. From a pedagogical point of view, we feel this is the desired result; students should spend the majority of their time in a computing class struggling with computing concepts. We were also sensitive to any negative impacts of the robot on the student's ability to complete their assignments; both the lab times+testing and framework codes represent difficulties that might not occur in a non-robot CS1 class. Fortunately, these responses were expressed fairly rarely, and combined make up less than 20% of reported difficulties. Reflecting the fairly robust and failure-free operation of our robot platform at CCAC no responses faulted the robot hardware,.

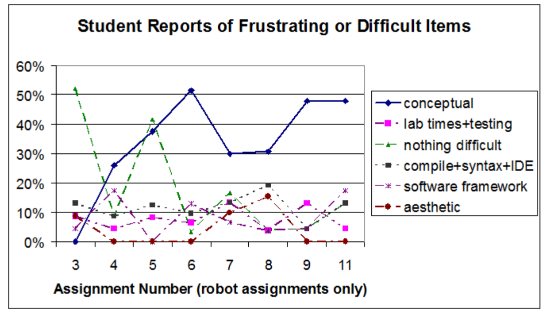


**Figure 3: Student sources of frustration**

*Relevance.* We asked students if they saw any links between the program written for the assignment and the operation of software, computers, or everyday devices. We were aiming to create assignments that would be relevant to students' lives so as to be engaging and motivating. Figure 4 shows the percentage of students who linked the assignment content to something they had previously experienced.

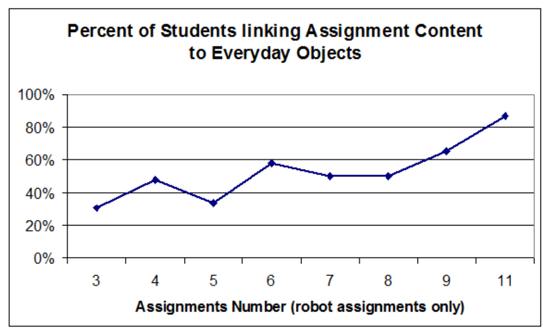


**Figure 4: Percent of students linking assignment content to external objects**

*Grades.* We compared the grades of students in our pilot to prior years to ensure that our students' learning was not hindered by the robotics assignments. The grade structure of the class was such that 75% of the grade consisted of exam grades. Exams were modified only superficially, to prevent cheating, between the pilot year and prior years; as such, we consider the performance on these exams and subsequent performance in the class as an adequate measure of comparative student learning. Figure 5 details the average grades in the 2007 pilot and prior years. Students in the pilot performed significantly better ($p<0.01$) than the five year average, and significantly better than three of the four previous years ($p<0.05$ for all three years).

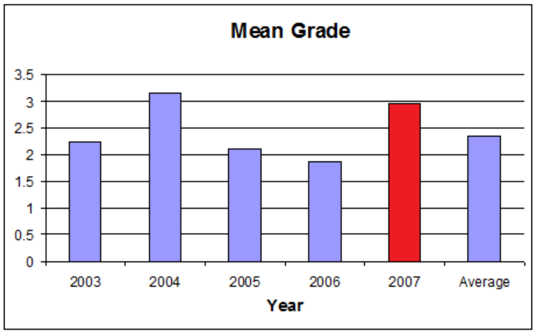


**Figure 5: Mean grades of passing students**

## 6. NEXT STEPS

We have presented the first completed stage of an iterative process to design a robot platform, software environment, and curriculum for introductory computer science education. By partnering with educators in the field and conducting significant initial evaluations of the classroom realities of CS1 education, we formed a foundation for our initial robot and curriculum design work. Our pilot results indicate that, with significant improvements to the robot design, the notion of using robots in CS1 has promise. While retention did not improve, enthusiasm and performance among passing students was significantly higher than in previous semesters even though our initial robot was imperfectly suited to CS1.

We have recently expanded our program, and are using the existing robot in the 2008-09 school year with 25 high schools across the United States whilst continuing to use it at CCAC; our new high school partners are experimenting with ways to introduce the robot into the more varied CS offerings at their different schools, and are helping us continue to develop activities and curricula that apply generally to robotics in CS. We are also now in the design step of our beta cycle, and are currently redesigning our robot platform, software, and curriculum to meet our goal of creating an ideally instrumented robot for CS1. We are planning to pilot our redesign at CCAC, other community colleges, and at the high schools of our partners, using the results of these future pilots to continue iterating our design until we have a readily disseminable curriculum and technology.

## 7. ACKNOWLEDGMENTS

We would like to thank the high school and college educators who are involved with CSbots, Chris Bartley for invaluable work on CSbots software API and Charmed Labs for providing initial hardware and support for the robot platform.

The research reported is supported in part by the Heinz Foundation, Microsoft, the National Science Foundation, and the Institute of Education Sciences (Grant R305B040063). The opinions expressed are those of the authors and do not represent the views of any funding agency.

## 8. REFERENCES

[1] T. C. A. Melchior, F. Cohen and T. Leavitt. More than robots: An evaluation of the first robotics competition participant and institutional impacts. http://www.usfirst.org/uploadedFiles/Who/Impact/Brandeis_Studies/FRC_eval_finalrpt.pdf, 2005.

[2] B. Brown. *A Guide to Programming in Java; 2nd edition*. Lawrenceville Press, 2007.

[3] J. M. Douglas Blank, Deepak Kumar and L. Meeden. Advanced robotics projects for undergraduate students. In *Symposium on Robots and Robot Venues: Resources for AI Education*, 2007.

[4] M. L. Fagin, B. Measuring the effectiveness of robots in teaching computer science. In *SIGCSE*, pages 307–311, 2003.

[5] E. Hamner, T. Lauwers, D. Bernstein, I. Nourbakhsh, and C. DiSalvo. Robot diaries: Broadening participation in the computer science pipeline through social technical exploration. In *AAAI Spring Symposium on Using AI to Motivate Greater Participation in Computer Science*, 2008.

[6] H. J. Kim, D. Coluntino, F. G. Martin, L. Silka, and H. A. Yanco. Artbotics: community-based collaborative art and technology education. In *SIGGRAPH '07: ACM SIGGRAPH 2007 educators program*, page 6, New York, NY, USA, 2007. ACM.

[7] T. Lauwers and I. Nourbakhsh. Informing curricular design by surveying cs1 educators. In *Proceedings of the 4th International Symposium on Autonomous Minirobots for Research and Edutainment*, 2007.

[8] M. McNally. Walking the grid: Robotics in cs2. In *Proceedings of the 8th Austalian conference on Computing education*, pages 151–155, 2006.

[9] D. P. Miller and C. Stein. "so that's what pi is for!" and other educational epiphanies from hands-on robotics. pages 219–243. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[10] R. M. B. R. Rusk, N. and M. Pezalla-Granlund. New pathways into robotics: Strategies for broadening participation. http://llk.media.mit.edu/papers/NewPathwaystoRobotics-06-07.pdf, 2007.

[11] I. N. T. Lauwers and E. Hamner. Csbots: A case study in introducing educational technology to a classroom setting. Technical report, Carnegie Mellon University, 2008.

[12] J. Vegso. Interest in cs as a major drops among incoming freshmen. *Computing Research News*, May 2005.

[13] M. Yim, M. D. Chow, and W. L. Dunbar. Eat, sleep, robotics. In *Robots for kids: exploring new technologies for learning*, pages 245–292. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.