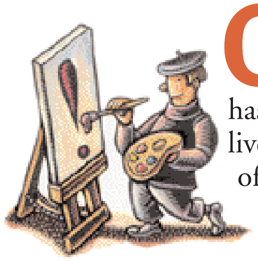
Viewpoint

Douglas Blank

Robots Make Computer Science Personal

They also make it more hands-on, real, practical, and immediate, inspiring a new generation of scientists' deep interest in the field.



LISA HANEY

Computer science has lost its appeal, and robots can help find it. Even as computing has invaded every aspect of our lives, CS as a field of study is often seen as disconnected from these same lives. So to reestablish the connection, the Institute for Personal Robots in Education (IPRE, www.roboteducation.org) is developing a personal robot, software, and curricula to help teach introductory CS courses. Imbued with the proper pedagogical philosophy and training, it can help make CS more personal.

Who is to blame for the current lack of interest? Well, me, for one. But CS education has long been a "student repellent," along with what might be viewed as asocial industry practices and unfortunate Hollywood stereotypes. Though there are islands of hope (such as Carnegie Mellon University and the Georgia Institute of Technology), overall, fewer students are enrolling in CS courses, staying with them, or moving into industry because we have washed them out of the classroom and the pipeline.

Then again, maybe the lack of interest simply reflects the state of the U.S. economy. But if you look at the numbers over the past 20 years, at least some invariants can't be explained away by economic booms and busts. Women and minorities have always been underrepresented in computing; their numbers in the U.S. peaked almost 25 years ago. Meanwhile, from 1998 to 2004 the enrollment of women in CS fell 80%, from about 1.5% to about 0.25%, according to the Higher Education Research Institute at the University of California, Los Angeles (www.gseis.ucla.edu/heri/heri.html) [4]. This trend is related to CS's reputation for being impersonal to all types of students, as reflected in the falling enrollment figures across the board, also according to the Higher Education Research Institute. Following recent research in CS gender issues, we now know much more about the weaknesses in CS education [2], most notably that students' personal values are often at odds with the environment in CS classrooms. For example, long solitary hours in the laboratory obsessing over minute details is exactly the opposite of what many students are looking for. If CS educators would confront this reality and develop the pedagogical principles needed to fix it, we could at least hope that the growing crisis in the CS pipeline might be ameliorated.

Toward this end, my colleagues and I at IPRE have embarked on a multi-year project to create new introductory CS courses and textbooks designed to be accessible and inspiring to all students. The cur-

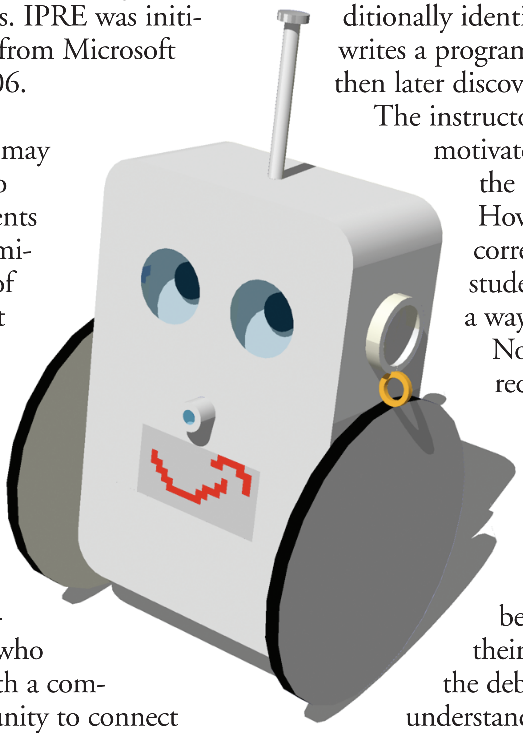
Projects like "robot dancing" represent a more natural learning environment, focusing on the creative not the merely correct.

ricula are centered around a small personal robot (about the size of a paperback book) tentatively named "Gyro" being developed at the Georgia Institute of Technology (see the figure here). Our vision is that each student would purchase one for about $150 retail at their college bookstores, using them throughout their CS explorations. IPRE was initiated through a $1 million grant from Microsoft Research, announced in July 2006.

At first glance, using robots may sound like a strange way to attract into CS those students who have traditionally been intimidated, excluded, or weeded-out of the field. One could imagine that robots would only exacerbate the problem rather than help alleviate it. However, CS is fundamentally about problem solving, and the example problems assigned to students in the classroom can profoundly influence how they perceive CS relevance and usefulness. For example, an instructor who illustrates the topic of parsing with a compiler example has lost an opportunity to connect CS with the rest of the everyday world. Exclusively using such incestuous examples is the equivalent of "computer science for the computer scientist." We must provide motivation that is instantly appreciated and understood by all.

Having an artifact—in this case a robot—provides intrinsic motivation to both the instructor and the student to explore the science and engineering behind it. Students continue for reasons (such as fun, curiosity, and to show off to family and friends) that are very different from those traditionally identified. Consider a student who writes a program that produces the output "5" then later discovers the correct output is "4." The instructor would likely attempt to motivate this student's innate interest in the art and science of debugging. However, the goal of producing the correct answer motivates only some students, while others see it only as a way to "please the instructor."

Now consider an assignment that requires students to write a program to create a dancing robot. The notion of "correct answers" is thrown out the window. In addition, the instructor doesn't have to artificially motivate debugging because the students generate their own motivation. They initiate the debugging conversation, eager to understand and fix a program not because it is perceived by the instructor as "wrong" but because the robot doesn't do what they want it to do. Projects like "robot dancing" represent a more natural learning environment, focusing on the creative not the merely correct.



Gyro's main program runs on a student's laptop. Sensor information is relayed to the laptop through a wireless link that also relays control commands back to the robot. Sensors include color camera, wheel position, and microphones. (Tucker Balch, Ben Axelrod, Daniel Walker, and Keith O'Hara at Georgia Tech, Atlanta, GA.)

Our embrace of the personal robot in the classroom is built on the shoulders of creative giants, including:

Karel. In the form of an ASCII character or toy-like robot only a few pixels tall, it moves through a grid world picking up and dropping off beepers [3];
Alice. Controlling beautifully rendered characters (such as chickens, snowmen, and ovens), it allows students to create animated stories (www.alice.org); and
Lego Mindstorms. These build-it-yourself robots are packaged as a kit (mindstorms.lego.com).

Our vision of a personal robot in the classroom adds to the conversation. It will emerge from its box, perhaps in the form of an egg or small creature rolling around on its own wheels, immediately usable by students for writing simple control programs or for instant messaging other robots and using the IMs to allow them to coordinate their robots' behaviors. They will operate in the real-world environment of walls and gravity, a place quite different from Karel's beeper-tagged grid world. In addition, the software, called Myro, we are developing for controlling robot movements, reactions, and environmental sensing can be used in real robotics research projects. (Myro is implemented in IronPython and C# running on .NET and Mono, the open source implementation of .NET.) It will help students progress from introductory coursework to more advanced concepts (such as data structures and object-oriented programming). It will be able to control a variety of simulated, educational, and commercial robots, including MobileRobots.com's Pioneer, iRobot.com's Roomba, and Lego Mindstorms. It will enhance students' understanding of what it means for software to perform within and despite the constraints and opportunities of the physical world, helping bring a sense of authenticity to the classroom, for which there is no substitute.

Guided by the philosophical principle of "pedagogical scalability," IPRE looks for tools that are simple to understand and use immediately yet provide strong pedagogical scaffolding as the student matures (pyrorobotics.org/aaai05-pyro-tutorial.pdf). One such tool is the language Python, which is intuitive, powerful, and easy to learn (www.python.org) [1]. We have been exploring its use in CS education for several years. We now hope to develop and identify a library beyond Myro, Gyro, and Python of pedagogically scalable hardware, software, and courseware tools for teaching CS.

Robots are no silver bullet for overcoming the current difficulty attracting and keeping students. But they can be used in combination with other ideas to create a more meaningful, accessible, interesting, intellectually challenging medium for teaching. We hope that they, along with personal open-ended assignments and a scalable pedagogical framework, will help attract, inspire, prepare, and keep a large and diverse group of students. Our aim is for them to find CS more satisfying, as well as a great place to refine their computational thinking, creativity, and career ambitions.

We begin testing these ideas in the classroom in the spring of 2007 at both Bryn Mawr College and the Georgia Institute of Technology, then explore the possibilities over the following three years. If all goes well, perhaps these personal robots will help CS regain the universal appeal it once had.

References

1. Guzdial, M. *Introduction to Computing and Programming in Python: A Multimedia Approach.* Prentice Hall, Upper Saddle River, NJ, 2004; see also the Pyro Robotics Project pyrorobotics.org.
2. Margolis, J. and Fisher, A. *Unlocking the Clubhouse: Women in Computing.* MIT Press, Cambridge, MA, 2003.
3. Pattis, R. *Karel the Robot: A Gentle Introduction to the Art of Programming, Second Edition.* John Wiley & Sons, Inc., New York, 1994.
4. Vegso, J. Interest in CS as a major drops among incoming freshmen. *Computing Research News 17,* 3 (May 2005); www.cra.org/CRN/articles/may05/vegso.

Douglas Blank (dblank@brynmawr.edu) is an associate professor of computer science at Bryn Mawr College, Bryn Mawr, PA. IPRE is a joint project of the Georgia Institute of Technology, Bryn Mawr College, and Microsoft Research and is directed by Tucker Balch at the Georgia Institute of Technology.