From Mad Libs to Tic Tac Toe: Using Robots and Game Programming as a Theme in an Introduction to Programming Course for Non-Majors

Jennifer S. Kay

Computer Science Department, Rowan University
201 Mullica Hill Road
Glassboro, NJ 08028, USA
kay@rowan.edu

Abstract

Computer Science has a bad reputation among non-CS majors. This paper describes three assignments from a gentle introduction to programming course for non-majors that uses robots and simple game programming as a hook to get students interested in the subject. In each of the assignments presented, what might be considered a trivial twist to an instructor was a key factor in making an otherwise standard project into something that is more engaging.

Introduction

It's a lot of work to fight Computer Science stereotypes: Whether it's the geeks who haven't showered in weeks hacking in the basement, the socially inept males playing raunchy video games at 3am, or even the Robot Wars machines chopping each other into bits, there are a lot of people who find it difficult to identify with the computer scientist.

Often the only chance we have to change this impression is in a Computer Science course for non-majors that students end up in through major or general education requirements. Worse still, our course could have been chosen as the lesser of two evils – it's programming or math, and "I'm not a math person."

So how do we make the classes students are forced into more engaging? Some, like (Distasia & Way 2007) and (Guimaraes & Murray 2008) use video-game like frameworks. Others use simpler games like tic tac toe to introduce students in courses as early as CS0 to topics in AI (Parker 2003), or in CS1 and CS2 to programming (Lorenzen, & Heilman 2002). (Dodds 2008) uses a wide variety of AI assignments including games and (mostly simulated) robotics. There are also many innovative educational software tools such as the Alice Programming Environment (www.alice.org) and hardware tools including education-specific boards like the MIPPET (Lobo et al 2000), and of course, inexpensive robots like those from LEGO Mindstorms (mindstorms.lego.com) and IPRE (www.roboteducation.org).

Introduction to Programming Using Robots

Introduction to Programming Using Robots is one of three "CS ½" courses we offer as general education electives in the computer science department at Rowan. The goal of these courses is to provide students with a good understanding of three key programming concepts: conditionals, loops, and functions. If time permits, other topics may be added, the most common of which is arrays.

Introduction to Programming using Robots was taught for the first time in Fall 2008. We use the Parallax Scribbler with the IPRE add-on board (Balch et. al., 2008), and the IPRE text book (Kumar, 2008). The language of instruction is Python, supplemented with Myro robot commands, which are imported just like other Python modules.

The Assignments

Mad Libs

In the traditional Mad Libs game (www.madlibs.com) players are asked for a list of parts of speech (nouns, verbs, etc.) that would be filled into appropriate blank spaces in a story that only the organizer would see in advance. The resulting silly story would then be read out loud.

The class Mad Libs project stayed quite close to the original idea. The students created 5 files: a story file, where the end of a line corresponded to a "blank" in a Mad

Libs story, and four parts of speech files. The program would ask the player to select parts of speech from the lists to fill in each "blank." Instead of just presenting their final stories on the screen, we used Myro speech synthesis tools to read the resulting story out loud. This seemingly trivial addition made all the difference in class enthusiasm; listening to a computer read out silly stories with its emotionless monotone voice makes quite an impression!

Graphics: Shapes and Motion

Myro includes some very simple commands for drawing points and shapes. We spent one lecture early in the semester trying out a number of these commands, including the idea of generating apparent motion by drawing and erasing shapes. Students were given a very open-ended assignment for the next week: use the tools that we have studied to create something fun. There must be some motion, and they must use at least 5 shapes. As an added incentive to be creative, students were told that the author of the project that was rated highest by the class would get a small "extra credit" award.

While there were, of course, some very basic projects, many of the results were surprisingly creative and detailed, including a ticking clock, swimming fish, and a ping-pong-ball-throwing stick figure.

The incorporation of motion seemed to move this project from the standard "homework assignment I've got to get done" into something worthy of cheers and excitement.

Robot Tic Tac Toe

We started the topic of game programming with a simple variant on the classic game of Nim in which players alternate taking one or two stones from a pile. The last player to take a stone loses. There is a known solution to win this variant that is easy for the students to understand. We wrote Nim as a group without using robots, and then moved on to Tic Tac Toe.

Tic Tac Toe proved to be a wonderful final project for our class. The project has two parallel tracks that we alternated work on: the standard computer game of Tic Tac Toe, and a robot player.

The computer game is much like that presented in any other introductory programming class. We have had to have discussions about winning strategies, board symmetry, etc.

The robot portion of the game involved large Tic Tac Toe boards made up of 9 squares (with very thick edges). Each Square is big enough for the robot to fit inside. Students used the (somewhat flakey) line sensors on the bottom of the robots to detect the lines to enable the robot to determine its location on the Tic Tac Toe grid. An easy setup that was interesting and challenging for the students.

Conclusions

Python is a fantastic language for non-majors to learn. It is simple, and yet powerful and used in real-world applications. Its design makes some topics that are difficult to learn in other languages, like lists and functions, so natural as to be graspable very early on in the semester.

Integrating robotics and games into a CS ½-level course in Python is practical and enjoyable for the students. Non-majors learn a bit about computer science and programming. An unexpected bonus for us was the recruitment of some new majors as a result of the class!

References

Alice.org, http://www.alice.org, accessed November 2008.

Balch, T., Summet, J., Blank, D., Kumar, D., Guzdial, M., O'Hara, K., Walker, D., Sweat, M., Gupta, G., Tansley, S., Jackson, J., Gupta, M., Muhammad, M. N., Prashad, S., Eilbert, N., and Gavin, A. 2008. *Designing Personal Robots for Education: Hardware, Software, and Curriculum*. IEEE Pervasive Computing 7, 2 (Apr. 2008), 5-9.

Distasio, J. and Way, T. 2007. *Inclusive computer science education using a ready-made computer game framework.* In Proceedings of the 12th Annual SIGCSE Conference on innovation and Technology in Computer Science Education (Dundee, Scotland, June 25 - 27, 2007). ITiCSE '07. ACM, New York, NY, 116-120.

Dodds, Z. 2008. *AI assignments in a CS1 course: reflections and evaluation.* J. Comput. Small Coll. 23, 6 (Jun. 2008), 262-271.

Guimaraes, M. and Murray, M. 2008. *An exploratory overview of teaching computer game development.* J. Comput. Small Coll. 24, 1 (Oct. 2008), 144-149.

Institute for Personal Robots in Education, http://www.roboteducation.org, accessed November 2008.

Kumar, Deepak, Editor, *Learning Computing with Robots*, Institute for Personal Robots in Education, Fall 2008.

LEGO.com MINSTORMS NXT Home, http://mindstorms.lego.com/, accessed November 2008.

Lobo, A. F., Baliga, G. R., Bergmann, S., Stone, D., and Shah, A. 2000. *Using real-world objects to motivate OOP in a CS1 lab*. Journal of Computing in Small Colleges, vol. 15, no. 5, pp. 144-156, May 2000.

Lorenzen, T. and Heilman, W. 2002. *CS1 and CS2: write computer games in Java!* SIGCSE Bull. 34, 4 (Dec. 2002), 99-100.

Mad Libs Official Website, www.madlibs.com, (accessed November 2008).

Parker, D. L. 2003. *Machine "Learning" at all course levels*. J. Comput. Small Coll. 18, 3 (Feb. 2003), 126-134.

Acknowledgements

This work was made possible by an award from the Institute for Personal Robots in Education.