Teaching Basic Programming Concepts using Python

Myungsook Klassen

California Lutheran University, Thousand Oaks, California, USA

mklassen@clunet.edu

Abstract

Improving retention and performance for CS majors with a weak background is crucial to increase the number of CS students. A CS0 course is designed to serve these students and desirably to recruit non CS majors before they take a rigorous first object programming class. A visual approach using 3-D animation and multimedia objects such as pictures and sounds in CS0 on different students groups is proven to be effective. In this paper, we present our study of a few different visual approaches with two different groups of students, traditional students and working adult students.

Introduction

Despite a rebound of the job market since the 2000 dot-com bubble burst, the enrollment of CS students continued to decline. Vegso[19] reported that an analysis of survey results from the Higher Education Research Institute at the University of California at Los Angeles (HERI/UCLA) indicate that the popularity of computer science (CS) as a major among incoming freshmen at all undergraduate institutions has dropped significantly in the past four years. The percentage of incoming undergraduates indicating that they would major in CS declined by over 60 percent between the Fall of 2000 and 2004, and is now 70 percent lower than its peak in the early 1980s.

It is important for CS educators to find a way to 1) recruit freshman students to CS courses and 2) to keep them in the program once they are in. It is shown that failure rates in the first programming course (CS1) are as high as 30% [3] and the majority of students drop out of the computer science major after the first programming course[14]. When the first programming language is an object oriented language, students are more challenged. Students have to learn a large amount of imperative language concepts and the concepts inherent in object-oriented languages such as abstract data types (class), inheritance, aggregation, and polymorphism.

Recent investigations have found that major factors contributing to attrition include lack of programming experience prior to entering college. Students with no prior programming experience are likely to be overwhelmed by the breadth and depth of the first object oriented programming class. To address these difficulties with CS1, schools now offer pre-CS1 courses (CS0) to CS majors and to students who wish to major in CS but don't have a solid foundation in programming concepts. CS0 typically covers problem solving, logical reasoning, algorithm design, and programming constructs with minimal or no emphasis on syntax.

To capture a student's attention with fun and interesting activities while maintaining academic rigor is a prerequisite to the teaching of any courses. Achieving such a combination in beginning computer science courses will improve CS student recruitment and retention. Research shows that use of graphics and animation is an effective teaching tool to get students interested in courses[3][4]. When programming concepts are presented through objects in a visual and animated way, students find programming fun [5]. Most students nowadays are visual learners and there was a large body of evidence supporting the idea that students learn programming concept better when given a visual representation and when solving or dealing with visual objects.

We employed a few different pedagogy in CS0. The first was to draw and to trace flow charts with paper and pencil. We used the visual approach in CS0 using Alice software[1], which was developed from Carnegie Mellon University to be a student's first exposure to programming. Then, we switched to Jython, Java version of Python [23]. In this paper, we will share our experiences and observations of using a visual approach using Alice and Jython with two different groups of students, namely traditional undergraduate students and working adult students. Our observations are anecdotal. Data has been collected since year 2006 but our program is not large enough to present meaningful quantitative analysis. We believe our observations have benefit for others in the computer science educa-

tion community, especially those considering non traditional approaches in a programming concept course for both traditional and working adult students in small liberal art colleges.

Background

There are several integrated development environments for Java that utilize a Graphical User Interface (GUI). BlueJ, NetBean, and Eclipse are a few examples. These tools are very user friendly, and make programming and debugging much easier. But Cooper claims[2] that it still adds another unnecessary level of complexity for students to master since they still lack the intuitive nature of a "point and click" interface.

Saint Joseph's University and Itaca College[2] used Alice to improve retention and performance for "at risk" introductory computer science majors and reported that the course did improve students' performance. Cooper[2] shows that students who took Alice CS0 class got better grades in CS1 than those who didn't take it. Also his data shows that comparing to students who didn't take CS0, a higher percentage of students who took CS0 went to CS2. Another study [13] shows that the retention of female students is high in CS0 courses that utilize 3D graphics, virtual worlds, and sound. Stage3Research reports [6] that their earliest pilot study showed that using the 3D animation software "Alice" made the process of writing a program much more compelling for female students.

Karel the Robot was developed more than 20 years ago, and C++ version and Java version Karel Robots [15] were developed and have been widely used as a CS0 tool. It provides a learner-centered world that can be explored by directly manipulating objects in the world with a limited set of simple commands. It is coupled with metaphors to aid in problem description and to exploit storytelling as an educational paradigm. Stanford University programming methodology class used Jkarel installed in Eclipse environment.

Jeroo[7], developed by Sanders[12] is a narrative tool which supports programming to tell a story. Unlike Alice, the program is small and does not require much memory to run. Jeroo is similar to Karel the Robot, but has a narrower scope. It focuses on just control structures, methods, and objects. It is suggested that Jeroo is much more effective if it is interleaved with a primary language by teaching it the first 3-4 weeks of a regular semester.

JPie[8] developed by Washington University, is a visual programming tool which supports the construction of programs through drag and drop. It is designed for students without programming experience and is used in CS0 with breadth first approach. It provides a gentle introduction to software design and implementation. It is very diverse and powerful to solve fun problems such as animated characters to complex problems like pong, dining philosopher, and Persian recursion. However the program is not widely adapted outside Washington University.

RAPTOR[10] provides a flow model environment where students can build programs by manipulating connections between icons or program elements. It is a flow chart IDE, and supports program development via manipulation of a flow chart. The Air Force Academy recently replaced Ada95 and MATLAB with RAPTOR in its CS0 course and reported students' better performance[11]. It is also used in Carnegie Mellon in a CS0 like course where the main goal is to teach computer science rather than computer programming.

Lego Mindstorms[9], has been used by a variety of CS educators in a wide range of courses. Programming can be done through visual programming environment such as Robolab for novice programmers.

Python is a powerful object-oriented scripting language used primarily in UNIX environments. It allows beginning students to get going quickly on interesting projects. The syntax is generally easy and the language is highly expressive for novice programmers. Centre college uses Python as a first program language for majors and non majors and reported they are generally happy with outcomes[22].

VPython, developed by David Scherer [16] allows students to write programs with visual effects, even for students with little prior programming experience. Students can do 3D programming using the VPython toolkit to model a number of subjects including gravity, friction, and tension demonstrating falling balls, weights, wind and satellites.

Jython is an implementation of Python seamlessly integrated with the Java platform. Georgia Tech computer science [3] built two sets of tools to Jython. The first one is a development environment for the students called JES. Second, a set of media tools to enable students to manipulate sounds and pictures. With media tools, students use computers for imaging manipulation, exploring digital music, viewing and creating web pages. In the Media computation course [3] at Georgia Institute of Technology, students were taught to program in the context of computer in applications and problems in society using Jython. Audio and visual aids were heavily used. They reported that only two students out of 120 dropped the course and that ninety seven percent of students in this course agreed that they

were actually learning programming.
JavaScript is another script language and shares many good features with Python. With HTML, JavaScript is interactive and visual. Its simplicity, natural web-based interfaces make it possible for novices to develop interesting and interactive programs quickly [18]. Using JavaScript fundamental concepts of programming can be easily taught. It offers a limited object oriented programming development environment. It has many built in objects which can be used to write powerful programs, but new objects and classes can not be created by users.

Language Choice Evolvement

We measure success of CS0 by the following three criteria:
1) preparedness of CS0 students for CS1 content
2) the rate of students who stay as CS major
3) the rate of students who declare their major/minor as CS after CS0 is completed.
The first goal is the most important, and is qualitatively analyzed through informal conversation with individual students while they were taking CS1 or shortly after they took CS1. We kept track of students' course takings to collect data for the second and the third measures.

Student Profile
California Lutheran University(CLU) is a small liberal arts college. The Computer Science (CS) department serves two groups of students in two separate programs. One is traditional students who come to the university right out of a high school. These students have little programming experience, or work experience in IT fields. The other group is returning working adults in the evening program. They are between 25 and 45 years old with an average age of 37 and 95% of students work full-time. Most of them have at least a few years work experience in IT fields while a small percentage of students have never worked in the IT field, but hope to do so in the future. Contrary to traditional students, almost all of returning students know what their major is and the rate of them changing their major is very low. The attrition rate in the evening program is also quite low, about 10% compared to that of traditional day students of 35%.

Alice 3D Programming
We introduced a CS0 course in year 2002 as a means for preparing weak students for CS1. Initially pseudo code and flow chart approach were used, but eventually Alice software was identified as a tool for its fun way of teaching object oriented programming by using attractive 3D animation environment and for its potential to attract non majors in computer science.
Alice was used in 2004 and 2005 for both traditional students class and working adult students class. Throughout the class we closely monitor students' reactions, attitudes, and learning outcomes. Many features of Alice software such as the colorful learning environment, built-in 3D objects, drag and drop, and animation made learning quite exciting and enjoyable, especially in the beginning. Being able to run programs from the beginning without spending much time setting up environment also contributed a high acceptance level of Alice initially. Students were able to create complex games whose codes generated by Alice were over 200 lines. One student wrote a car collision avoidance game and another student wrote a game of shooting down airplanes.
One of the drawbacks, which is also the strength of Alice, is that students spent much time creating a great picture rather than mastering a given programming concept. "Look and feel" played an important role, as much as programming. When objects are 3D and are displayed in a small screen, it takes good practice to align objects in 3D space. That distracted students from concentrating on programming concepts.
For working adults students class, there were 15 students, 13 CS majors and 2 business majors. All 13 CS majors continued to CS1, but 2 business major students never continued beyond CS0[20]. For traditional students class, students came from several majors, CS, math, multimedia, business, and liberal arts. All twelve students who declared their majors CS prior to CS0 moved to CS1 after the course was over. But no one switched to CS major after CS0. During the following course offering, there were about 20 students in the traditional CS0 class, and over a half were non CS majors, mainly from a multimedia major. We have not investigated why there was a the sudden surge of non CS major's interest in CS0.

Alice was better received by the traditional students than by the working adults students. In course evaluations, traditional students showed a high satisfaction level with the class. Course evaluation from working adult students showed a few low spots.

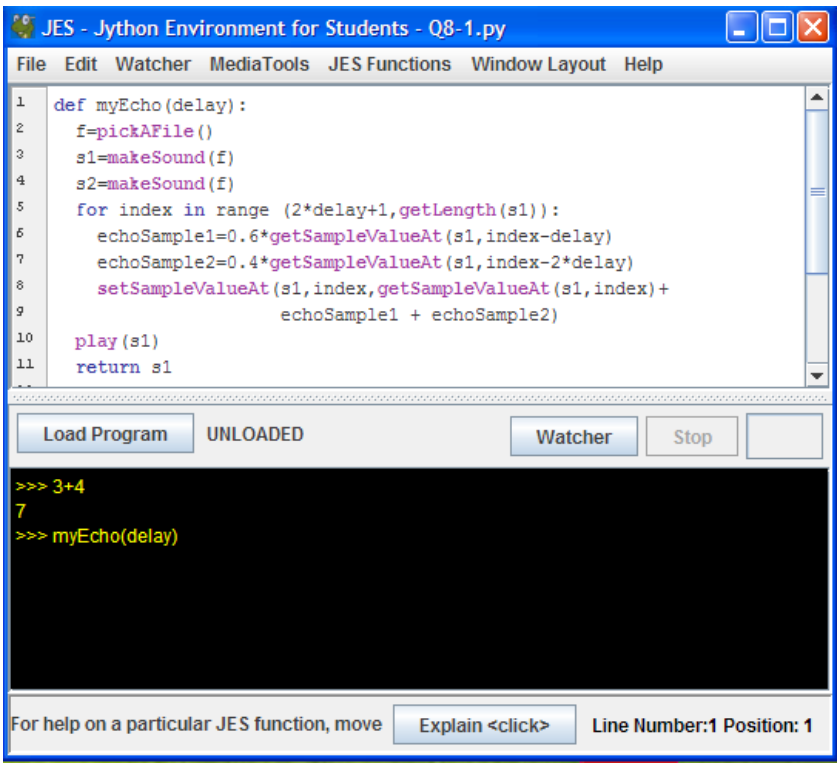
1. "Experience a high degree of intellectual and/or creative challenge" . 3.4 in the scale of 5.0. the departmental average is 4.3
2. "Find classroom activities appropriate to learning objectives for the course". 3.8 in the scale of 5.0. the departmental average is 4.3.

They just didn't feel compelled about using animation objects to learn programming concept. More "serious looking" tools and rigorous environment would serve working adults students better. The biggest problem was that fact that they didn't write a single line of code.
Using Alice for the entire semester didn't serve our main goal of providing a solid programming concept to our CS majors/potential majors for both groups of students. Alice itself alone in CS0 was not enough to prepare computer science majors for CS1. Students expressed that a transition from a "click and drop" style in CS0 to writing actual codes and dealing with syntax in CS1 was challenging. At a minimum, CS0 should do actual programming and CS0 should take a more rigorous approach and solve relatively harder and diverse problems. The most important goal, preparing students for CS1, was not met with Alice software for both groups of students, traditional students and working adults students.

Jython

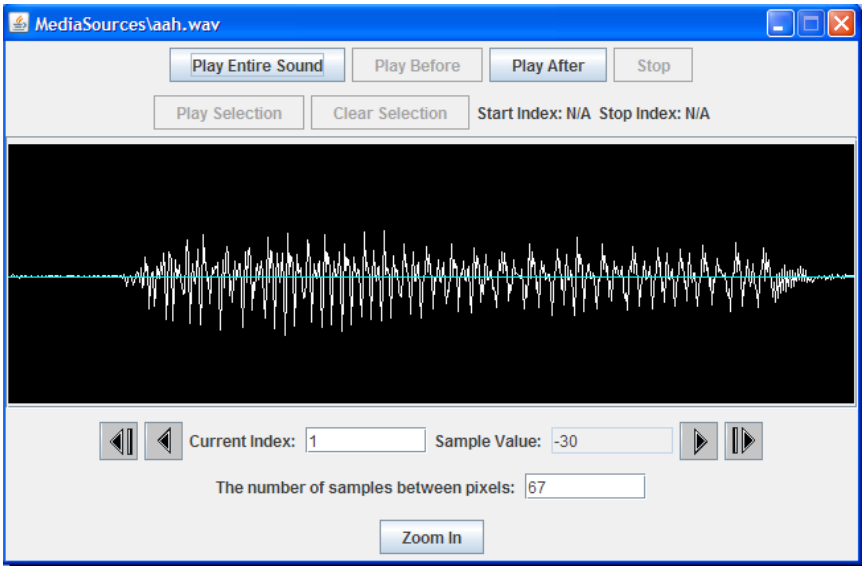
Python was chosen since it offers simplicity and practical applicability, unlike Alice, which is purely educational language. Python is a general purpose language which is suitable for developing real world applications. Python is used by large companies such as Yahoo and Google and ranks among the ten most popular languages[21]. If students do not continue to CS1, they still have a marketable programming skill of Python.

Figure 1:Jython Development environment JES



From various Python family, Jython with a good GUI environment was chosen as the next programming tool in CS0. Jython is extremely useful because it provides features of a mature scripting language while running in any environment that supports a JVM. Today, this means most major computing systems, including Microsoft Windows, Mac OS, most UNIX variants including all Linux systems, and all IBM systems.

Figure 2:JES Sound tools



The Jython Environment for Students (JES), developed by Georgia Institute of Tech[22] is a full-featured media computation environment for programming in Jython. It includes facilities for programming, debugging, and multimedia API as shown in Figure 1. An extensive multimedia API, enabling easy and rapid manipulations of sounds, images, and on some platforms, video is very intuitive and easy to use. The top part is the program area where students write functions and the bottom part is the command area where students can type python command interactively or call functions defined above.

The book "Introduction to Computing and Programming in Python: A Multimedia Approach" written by Guzdial[3] at Georgia Institute of Technology was used in class. The book uses JES environment extensively and contains many examples of different levels of difficulty. For instance, a few lines of code can convert a color picture into a black and white one. A complex example is to create a collage of an original picture, its mirror image vertically or horizontally, its b/w picture, and its scaled picture. Some picture manipulation, which is often done by Photoshop, is programmed easily with Jython. The sound tools are also quite easy to use. Students can choose a specific range of the sound, shows index and its amplitude. They can look at sounds at the sample level, record new sounds, and playback sounds. With sound tools, students can write a few lines of codes to play the sound backward or write a complex program to create delayed echo sounds and to create blending sounds.

Throughout the class, students were very engaged in writing programs to manipulate pictures and sounds. Students were able to deal with syntax problems with a moderate level of frustration for two reasons. First, they had fewer errors due to simple syntax of the Jython language and its indentation syntax. Second, JES environment colors keywords, and draws a box surrounding a block.

CS0 was taught in 2006 and in 2007 for working adult students and was taught in 2008 for traditional students. In 2006, there were 6 students, 5 CS majors and 1 Business major and all 5 CS majors moved to CS1 later. One student dropped out of CS major and from the school after CS1. In 2007, 1 out of 5 was CS major who stayed as CS major and the rest 4 students were non CS majors and they never took any more CS course beyond CS0. For traditional CS0 class, there were 5 students. Two were CS majors and they stayed as CS major and are taking CS1 currently. Three students were non CS majors and didn't intend to switch to a CS major.

We had follow up discussion with students who went to CS1. All of them expressed that CS0 prepared them well for rigor of CS1. We also learned that their transition to CS1 was smooth with Jython. The CS instructor who

taught CS0 with Alice and with Jython observed the classes carefully and concluded that Jython better served our needs. After all, students were actually writing code to learn programming concepts, so transition from CS0 to CS1 was quite smooth. There is not much difference in course evaluations from two groups of students. The traditional students and working adult students had similar experience with CS0 and expressed a similar satisfaction level. Both groups liked using pictures and sounds in programming and they had fun programming, although some programming tasks in class were quite challenging. For the third goal, using Jython didn't increase the rate of students declaring their majors to CS after they took CS0. For the second goal, we will need to observe for a little longer to measure if any students fall out of CS major.

Using Jython in CS0 is not without issues. Students don't get a chance to learn about data typing since Jython variables are associated with an identifier. Jython's dynamic typing leaves students lacking practical exposure to data type concept. JES a minimal debugging feature, comparing to some programming environments such as Eclipse and Netbean. Students had trouble finding syntax errors when problems were not obvious.

Conclusion

Our first attempt of visual approach using Alice entertained both groups of students. Although traditional students were happier with CS0 approach using Alice and gave positive course evaluation, they quickly realized that transition from "click and drop" to writing programs was too steep and they felt they were not well prepared. Working adult students had less favorable evaluation with CS0 using Alice. Overall, the approach didn't meet any of our three goals. Two questions are "Experience a high degree of intellectual and/or creative challenge" and "Find classroom activities appropriate to learning objectives for the course." Scores are 3.4, and 3.8 out of scale 5 which are below average of all other courses. Informal interviews with students who went to CS1 showed they felt CS0 didn't prepare them well for CS1 rigor.

With Jython, two same questions from the course evaluation showed much improved results with scores of around 4.2 as shown in Table 1. The course evaluations from both groups show a similar level of their satisfaction.

Table 1: course evaluation results

	Working adult students	Traditional students
Experience a high degree of intellectual and/or creative challenge	4.2	4.3
Find classroom activities appropriate to learning objectives for the course	4.3	4.4

Our informal qualitative analysis implies that Jython meets one of three objectives, preparing students for CS1. Jython is a simple yet powerful, and practical language which both groups of students appreciated. Both groups of students enjoyed working with multimedia objects, sounds and pictures. Our data does not show much improvement on recruiting non CS major students to CS major either with Alice nor with Jython. Simply serving non CS majors who are interested in knowing about programming, Alice seems better with traditional students. But working adult students care less about Alice.

It is difficult for one course to provide a solid foundation of programming concepts for CS majors and to provide an opportunity to non majors what "programming" is all about. The former goal can be better achieved using a depth first approach while the latter by a breadth first approach. Our primary concern is the former and Jython serves us better.

References

01. Alice(2008). Online, Internet. Available www:/http://www.alice.org.
02. Cooper,S. Pauch, R. and Dann, W. (2000). Using Animated 3D Graphics to prepare Novices for CS1. Proceedings of the 5th Annual CCSC Northeastern Conference 2000, Ramapo, NJ, Computer Science Education Journal.
03. Guzdial, M. and Usselman (2003) Media Computation Course for Non-Majors. ITiCSE Proceedings. P104-108 .ACM. New York.

04. Rodger, S. (2002) Introducing Computer Science Through Animation and Virtual World. 33rd SIGCSE Technical Symposium on Computer Science Education.
05. Zimmerman, G., and Eber, D. When the world collide! An interdisciplinary course in virtual reality art. 32nd SIGCSE Technical Symposium on Computer Science Education 2001. 75-79.
06. Jeroo. (2009). Online, Internet. Available WWW:http://www.jeroo.org
07. JPie. (2009) Online, Internet. Available WWW:http://jpie.cse.wustl.edu
08. Lego Mindstorms. (2009). Online. http://www.legomindstorms.com/
09. RAPTOR. (2009) Online. Available WWW: http://www.usafa.af.mil/df/dfcs/bios/mcc_html/raptor.cfm
10. Carlisle, Wilson, Humphries,& Hadfield. (2004). RAPTOR: Introducing Programming to Non-Majors with Flowcharts. Consortium for computing sciences for colleges. P52-60.
11. Sanders, D. and Dorn, B. (2003). Classroom experience with Jeroo. Journal of Computing in Small Colleges. 18,4. 308-316.
12. Kenneth J. Goldman.(2003). A Demonstration of JPie: An Environment for Live Software Construction in Java. Conference Companion, 18th Conference on Object-Oriented Programming, Systems, Languages, and Applications, pages 403-414.
13. Zaccone, R., Cooper, S., and Dann W. (2003). Using 3D Animation Programming in a Core Engineering Course Seminar. 33rd ASEE/IEEE Frontiers in Education Conference.
14. JKarel. (2006). Online. Available http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html
15. Vpython. (2009). Online. Available http://www.vPython.org.
16. Zeller, J. (2004). Python Programming: An Introduction to Computer Science. Franklin, Beedle & Associates, Wilsonville, OR.
17. Reed, D. (2001). Rethinking CS0 with JavaScript. Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education.
18. Vegso,J. (2005). Interest in CS as a Major Drops Among Incoming Freshmen. Computing Research News, Vol. 17.
19. Klassen, M. (2006). Visual Approach for Teaching Programming Concepts. Proceedings of 9th International Conference on Engineering Education.
20. Radenski,A.(2006). Python First:A lab-based digital introduction to computer science. ItiCSE.
21. Oldham,J. (2005). What happens after Python in CS1? Journal of computing sciences in colleges. Volume 20 , Issue 6.
22. Jython . (2009). Online. http:://www.jython.org