Ubiquitous Web-Based Programming

Anna Maria Luxton
School of Mathematical and Computing Sciences
Victoria University of Wellington, New Zealand
annie@mcs.vuw.ac.nz

Abstract

Learning to program can be very hard. Without an existing background in the subject and without adequate resources, students, especially women, may find themselves experiencing much frustration at their lack of understanding. A ubiquitous usable learning environment, that pictorially depicts run-time information in the form of animated visualisations, could support the learning of some common programming techniques. It would also help provide that previous experience essential to becoming a proficient programmer. This report presents JavanOwl, a web-based educational programming environment designed to provide all the above.

1. Introduction

The amount and complexity of information that students in introductory Computer Science courses are expected to absorb is exceedingly high. For most students, the level of understanding required is a lot to handle. For those without any prior experience and little confidence with computers, this task can be quite overwhelming. Unfortunately, since most New Zealand schools do not offer Computer Science as a subject, perhaps due to the high cost of maintaining such a course, the majority of students entering introductory Computer Science courses at tertiary level fall into this latter category of being educationally disadvantaged. According to Marian Gunsher Sackrowitz and Ann Parker Parelius [3], prior familiarity with programming concepts is the most useful predictor of success in introductory Computer Science courses, and that a high level of success is difficult to achieve without this prior knowledge. Despite the efforts of teachers and tutors to communicate complex concepts through diagrams and analogies, educationally disadvantaged students who lack prior experience with programming show a lower persistence level. Our experience is that retention strategies do not address these earlier problems [2]. This trend affects women in particular, as they tend to have less exposure and less confidence with computers than men do upon entering Computer Science courses.

In order to raise the level of retention at introductory Computer Science courses we must provide a way for beginners anywhere to attain some prior familiarity with programming concepts. In this paper, we introduce JavanOwl, an accessible, user-friendly programming environment designed to teach, that automatically makes use of software visualisations. We discuss the general idea of JavanOwl, together with the advantages that could gained through the use of such an application.

2. JavanOwl Technology: How it Works

JavanOwl is a web-based educational programming environment designed to help beginners learn how to program. Being web-based, it provides a cheaper and ubiquitous alternative to traditional applications. Users will be able to use it from any computer, no matter how old and cheap, with Internet connectivity. This reduces the need for owning expensive resources such as a computer, software, textbooks and teacher time. Using the system allows users to create, edit, save, retrieve, and run both example programs provided and their own code. The user enters their code through a simple form and then runs it. If the code has errors, the browser will display an adequate error message generated with the help of a Java parser. If the code is error-free, the browser will be redirected to a split-screen page displaying their code in an editor and beside it, the output of the code (figure 1). Users can then edit their code and re-run it. This provides immediate feedback which is important in the process of learning to program. Users will write stand-alone code similar to simple Java code. This gives an OO-first approach which may help students overcome the difficulty they encounter when attempting to shift their understanding of programming to include OOP concepts. JavanOwl also aims to integrate a visualisation tool based on Aspect Oriented Programming [4] to create run-time visualisations of the code being executed. Visualisations are widely used as teaching aids to help students understand the

concepts being taught [1], particularly in Computer Science where visual depictions help communicate the complex inner workings for program code. A built-in web-forum in JavanOwl is designed to allow a collaborative community of idea and file sharing to arise.

JavanOwl has been implemented using HTML, JSP, JavaBeans, JDBC and an SQL database, all on the server side. It has been designed to be as portable as possible by using no plug-ins and keeping the client-side as lightweight as possible. This reduces the need for special client-side support and avoids browser compatibility problems, allowing the application to be run on as many browsers as possible.

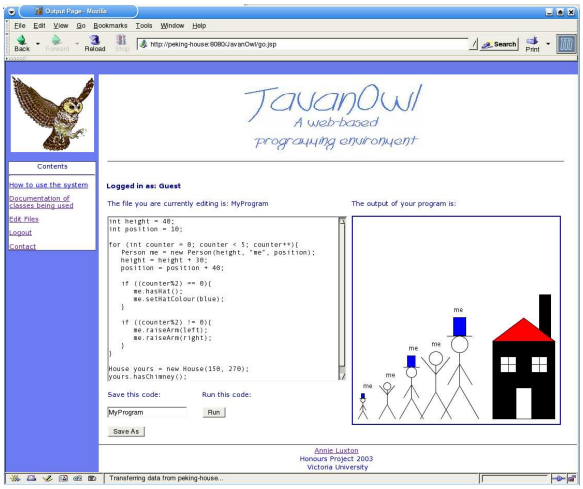

Figure 1. JavanOwl: output page

3. JavanOwl in Use: How it Will Help

JavanOwl can be used as a teaching aid in classrooms or in conjunction with introductory Computer Science courses at tertiary level. Alternatively, picture the following scenario. Student X attends a local public highschool. She wishes to learn how to program in Java, but is told by her teacher that programming is not taught at their school. Student X is advised to:

- buy herself a relevant textbook, download or buy all the relevant software and install it on a PC either at home or at school, and begin working on it herself. Student X finds that textbooks are very expensive and that she cannot possibly afford one. The PC she has at home is not 'hers', so she cannot install software on it, and she is afraid of doing something wrong on the ones at school. She feels lost and confused. Without some teacher aid, she doesn't know where to begin the rather long and arduous task of learning how to program.
- try using JavanOwl. Student X accesses the JavanOwl application online from the school computer during her lunch break, creates an account and begins by looking at some sample code. Student X realises she is able to access her account on JavanOwl from any computer, and so spends some time at internet cafes playing with code. Soon, student X finds the courage to try writing some code of her own, experimenting with different programming constructs and techniques. She uses the visualisation tool to help her grasp how the code works and if she has any questions, she posts a message to the web-forum and waits for a reply from a tutor or a fellow student. Student X writes some fun programs (a phone-book, for example) and passes her program's URL onto friends for them to try running. Student X has now successfully gained the experience with programming that will increase her chances of succeeding at an introductory Computer Science course. This is the vision of how JavanOwl will help education in Computer Science.

4. Conclusion

The motivation for a web-based educational programming environment that utilises software visualisations has been outlined, and JavanOwl has been introduced. JavanOwl aims to be the 'Hotmail' of programming environments, a ubiquitous online service that will create a safe and collaborative programming community for beginners. It will reduce the need for the expensive resources involved in teaching/learning to program. It explores the use of visualisations to better explain complex programming concepts. A prototype of JavanOwl is currently being evaluated and extended.

Acknowledgement: advisors Robert Biddle, James Noble.

References

[1] K. Brodlie, J. Wood, and H. Wright. Scientific visualisation - some novel approaches to learning. *Integrating Technology into Computer Science Education*, June 1996.

[2] J. Brown, P. Andreae, R. Biddle, and E. Tempero. Women in introductory computer science: experience at victoria university of wellington. In *Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education*, pages 111–115. ACM Press, 1997.

[3] M. Gunsher Sackrowitz and A. Parker Parelius. Women in the introductory computer science courses. In *Proceedings of 27th SIGCSE Technical Symposium on Computer Science Education*, March 1996.

[4] R. Khaled, J. Noble, and R. Biddle. InspectJ: Program monitoring for visualisation using aspectJ. In M. Oudshoorn, editor, *Proceedings of the 26th Australasian Computer Science Conference*, Adelaide, South Australia, 2003. Australian Computer Society.