Incorporating the Virtual Programming Lab into a First Year Computer Science Module

Dr. Aidan Mooney, Dr. Susan Bergin and Emlyn Hegarty Kelly
Department of Computer Science, Maynooth University

Class Size
305 Students

Feedback Approaches
Automated feedback, Generic feedback

Discipline
Computer Science

Technologies
Virtual Programming Lab

Challenge & Aim

For many students attending third level, Computer Science is a new discipline. As part of their first year students will undertake a module in programming. Computer Science has a notoriously high failure rate at the end of first year with programming modules seen as a major stumbling block. Programming is an individual task and one which can be very frustrating with struggling students feeling isolated and often embarrassed to ask questions.

For novice programmers feedback from traditional Integrated Development Environments (IDEs) can be demoralising with potentially numerous errors frustrating students. Feedback from these IDEs tends to be high level and confusing for novice programmers.

The aim of this case study was to provide automated real-time feedback and grading to students in their introductory programming module. This feedback system would be available to the students in their weekly labs and also when they are doing self-directed study. The automated system used was the Virtual Programming Lab (VPL) which allows for tailored feedback that gives an automatic grade even if a program does not compile.

Evidence from the Literature

Quille et al. (2015) and Mooney et al. (2010) discuss the low progression rate (~74%) from first year to second year in Computer Science courses within Ireland. A large contributor to the lower progression rate in Computer Science has been identified to relate to students struggling to master fundamental concepts in their first programming module (Watson, 2014). Vihavainen (2014) showed that even when looking at statistics describing pass rates after teaching interventions, as many as one quarter of the students still fail the courses. Getting these interventions right is critically important for the retention of students.

A number of automated grading systems are available for computer programs such as Web-CAT, AutoLab and VPL. The ability to incorporate the instructor's code as part of the test frame has contributed to a growing interest and success of VPL (Thiébaut, 2015). VPL provides real-time feedback to students in a safe space. Any feedback should show students where they went wrong meaning it needs to be specific and detailed for the student to learn from. It is vital that feedback is provided in a timely fashion, even in large classes (Naylor et al., 2014). Gibbs and Simpson (2004) identified that feedback must be timely enough for students to have time to use it to improve their learning. This is something that VPL provides giving the student instantaneous, tailored and meaningful feedback.

Feedback Approach

As part of the introductory programming module students were required to complete a weekly task of writing a number of computer programs which formed part of their summative assessment within the module.

Stage 1: Teaching staff design problems based on the material covered in lectures.

Stage 2: Develop the scenarios based on the questions and develop the feedback to be returned to the students within VPL.

Stage 3: Student complete these tasks (1-3) over a number of days in their own time if they wished or during their mandatory weekly lab session. Student received automated and real-time feedback as they completed the tasks. To give tailored feedback and a suggested grade VPL looks at what the student writes in their code while they are completing the tasks and will test the students code against a number of test cases once they have finished writing the code.

Stage 4: Students complete an unseen task or question in the lab, which only became visible when they attended the session.

Outcomes

Student Response

A Moodle survey was administered to the class with 206 students responding to it. Students were firstly asked to rate how easy it was to use VPL on a scale of 1 to 10. The average response was 7.8/10 showing, that, in general, students found it easy to use.

The students were asked *"Did you encounter any technical problems when using Virtual Programming Lab?"* Some of the problems included:

The most cited technical problem related to VPL logging out the student without warning. This is something that is being investigated to improve the student experience.The students were asked *"Did you apply the automated Virtual Programming Lab feedback received to improve your code submission?"* Some of the responses included:

The student's responses showed that they were using the feedback received from VPL to modify and improve their code which was a positive finding.

The students were asked *"Do you feel that the use of Virtual Programming Lab helped you improve your coding skills?"* Some of the responses included:

Analysing the feedback from students shows that the students liked being able to get support and feedback when working outside of lab times. They also appreciated the grading system incorporated to highlight where marks were being awarded.

The students were asked *"What aspect of Virtual Programming Lab did you find most useful?"* Some of the responses included:

It is interesting to note that the positives of instant feedback and grading keep appearing for this answer. This is in line with the main aim of this project which was to provide automated real-time feedback and grading to students.

Finally the students were asked *"Do you have any suggestions for improving Virtual Programming Lab?"* Some of the responses included:

It can be observed from the feedback that most of the issues relate to style, layout and display issues. These issues, and others, are all currently being looked at as a number of in-house projects work at enhancing VPL.

Recommendations

VPL has proven to be a very effective feedback system. It should be noted that there is a significant amount of initial work required when using VPL to prepare the assignments, write the solution programs, generate the scripts to evaluate the student program, test the setup and incorporate all of this in to a new VPL activity in Moodle. However, this time is well spent as the setup for one question is easily transferable to other questions.

To achieve the most from VPL it is important to keep in mind that there is more than one way to achieve a goal. This means when you are working on a scenario for a question it is best to keep the feedback as general as possible, allowing the student several ways to reach the goal of completing the question. This method seems to work best as it allows students to explore with their code and improve their understanding. To achieve the general approach, it is useful to have another member of the teaching team check each question.

Practice makes perfect. You can never have too many practice questions. The reusability of the original questions means it is possible to add more questions with little overhead.

References

Gibbs, G. and C. Simpson, 2004. Conditions under which assessment supports students' learning. Learning and teaching in higher education. 1: pp. 3-31.

Mooney O., Patterson, V. O'Connor, M. and Chentler, A. (2010). A Study of progression in Irish higher education. Higher Educational Authority. http://www.hea.ie/content/2010.

Naylor, R., Baik, C., Asmar, C. and Watty, K. (2014). Prompts and guidelines for reviewing and enhancing feedback for students. Centre for the Study of Higher Education, The University of Melbourse.

Quille, K., Bergin, S. and Mooney, A. (2015). PreSS#, A Web-Based Educational System to Predict Programming Performance. International Journal of Computer Science and Software Engineering (IJCSSE), 4(7), pp. 178-189.

Thiébaut, D. (2015). Automatic Evaluation Of Computer Programs Using Moodle's Virtual Programming Lab (VPL) Plug-In. Consortium for Computing Sciences in Colleges (CCSCNE) 2015, Boston.

Watson C. and Frederick W.B. Li. (2014). Failure Rates in Introductory Programming Revisited. 2014 Conference on Innovation and Technology in Computer Science Education. ITiCSE '14.

Vihavainen, A., Airaksinen, J., and Watson, C. (2014). A systematic review of approaches for teaching introductory programming and their influence on success. In Proceedings of the Tenth Annual Conference on International Computing Education Research, ICER'14, pp. 19-26, New York, NY, USA, 2014. ACM

Contact

If interested in finding out more about this approach or technology, please contact Aidan Mooney at aidan.mooney@nuim.ie.

Cite as;

Mooney, A., Bergin, S., & Hegarty Kelly, E. Incorporating the Virtual Programming Lab into a First Year Computer Science Module. IN: Technology-Enabled Feedback Approaches for First-Year: Y1Feedback Case Studies in Practice: Y1Feedback. Available from: https://www.y1feedback.ie