Teaching Android development using eLearning material in a traditional Java programming course

Karsten Lundqvist

University of Reading

Corresponding author: k.o.lundqvist@reading.ac.uk

Abstract

World-wide some 152 million smart phones were sold in the first quarter of 2012 with a 49.9% increase from the year before (IDC, 2012). An important aspect of this popularity is the wide range of specialised programs known as apps that phone owners can readily acquire and install. For instance the number of apps on Google's play market for the Android platform has almost doubled within a year to just under 500,000 apps (Appbrain.com, 2012) and the amount of apps is often used in advertising as a major selling point for phones (Apple, 2012). It is therefore important for students' employability to gain skills in app development. Educationalists are starting to teach aspects of app development, for example developing workshops to aid learning the skills necessary to use the Google App Inventor tool (Abelson, Chang, Friedman, Lomas, & Wolber, 2010) (Hsu, Rice, & Dawley, 2012).

In this paper the author reports on their experiences of embedding an online tutorial on mobile Android programming development into the curriculum of a traditional second level undergraduate Java programming course. The distribution of the resulting marks did not follow the traditional distribution of a normal curve, and reflections on this unexpected distribution are provided.

The original online tutorial

In the summer of 2011 it had become clear that mobile phone development was becoming increasingly important, and that there was a lack of this in our curriculum. An online tutorial (Lundqvist, 2011) was created for a two day extra-curricular workshop; the tutorial was based on the author's experience of Android games development and consultancy. The tutorial was presented as five posts that takes a student with basic programming skills from first steps to a having developed a simple ball game. A dozen

ISSN: 2044-3315

students were expected to undertake the course, but forty-four signed up. Most students readily created the simple game, and some started to develop more complex apps that they continued working on in the following academic year.

Embedding into Java Programming Course

The Java programming course had previously been taught as a 10 credit (single term) module to second year Computer Science students, and it had been decided to increase it to a 20 credit (full year) module to allow students to improve their programming skills, coupled with including mobile phone programming in the curriculum. The e-Learning material developed for the extra-curriculum was re-used to provide 6 hours self-study which could be undertaken within supported lab classes. The lab work was complemented with 6 hours of lectures presenting challenges and pitfalls of general Android development. The students were then set an assignment to develop a personalised version of the traditional 'breakout' game.

There were eighty-three students on the module. The lab sessions each had three teaching assistants and a lecturer to support the self-learning session. These sessions were not well utilised, with only ten to twenty students using them, however from the server statistics it was clear that the students were using the online material extensively. For instance the number of pageviews increased by 110 pageviews on average per day in the period the students worked on the assignment. There are no other known reasons to explain this increase. Assuming this peak in usage primarily occurred because of the course work, approximately 6400 extra pages were shown to students, resulting in an average of ~75 pageviews/student. Therefore an average student would visit each page on the online tutorial in total ~15 times.

Results and Reflections

Of the 83 students 11 did not return the coursework and 14 students had very low marks (fail or non-honours level), however 20 received 1st class level and 16 got 2.1 class level marks. This resulted in a reversed normal distribution with peaks on each end of the scale and an almost "perfect" average of 50%, with fewer than expected students receiving marks around this average. Seemingly the good students had managed to become better than normal and the lower end students had struggled.

ISSN: 2044-3315

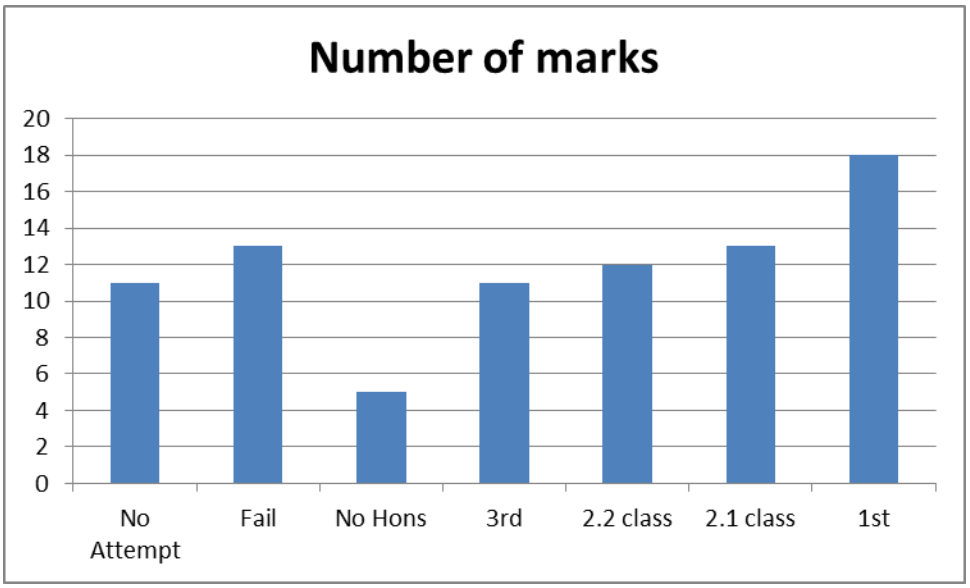

Figure 1: Distribution of marks

There are several factors that might describe why this happened, which will be discussed in the following.

Enthusiastic students

There were a group of students who were thrilled to develop for mobile phones, especially a game, as evidenced in feedback forms, informal discussion after classes, observed discussions among students, and evidenced by colleagues talking to students in their modules. There was evidence of a community of learners being developed, and this generated a positive feedback mechanism within the group of interested students (Thoman, Sansone, & Pasupathi, 2007).

Some students admitted they had spent more time on this to the detriment of other work. It was also clear from the resulting games that many of the students focussed too much on elements that were not assessed, for example the aesthetics, which were not an assessable outcome of this module.

- Is it possible to make course work too interesting or fun for students that they get distracted from what they need to learn?
- Should a non-game exercise be used to limit this?

Difficult combining self-learning with taught material

The Java module is a second year module, and therefore all of the students have followed the first year programming module (equivalent to CS0 in the US). For many

ISSN: 2044-3315

this was their first experience of self-study, and some had difficulty linking the theory taught in previous lectures, both in Java and other programming modules, with the self-study to achieve the necessary skills to complete the coursework. Seemingly the expected prior knowledge built up before this learning experience was not well established for the individual student, which according to (Biggs, 2003) means that the student is unlikely to use a deep approach to their learning.

- Is it reasonable to expect that second year undergraduates students can combine self-learning and taught theory within a short time scale?

Timing issues

Because the framework for the final game was given within the online eLearning material it was decided to have a short deadline on the coursework (some twenty days after the last lab session), which some students had difficulty in meeting. Many said that the transition from developing everything themselves to using somebody else's code as a basis had been more difficult than they anticipated. This could also be observed within the lower scoring student's work, as it was clear that they had to stop working before finalising their work.

- Would it help students to give them longer time, or would they procrastinate?

Conclusion

The use of Android and the online eLearning material within the Java Programming module was a success from the aspect of students' enthusiasm and participation. There were issues with doing it, especially for the poorer students who perhaps struggle with combining self-study and taught material. Additionally some students focussed on non-assessed aesthetics of the game to the detriment of assessed aspects. These aspects contributed to an unexpected distribution of marks. The timing of deadlines might also have contributed to this, and the lecturing team will try and improve on this, and emphasise the connection between online material and lecturing content in next year's module. It is being discussed whether the coursework should become a non-game exercise to help the students focus on the real learning outcomes, however it is feared that some of the positive enthusiasm might be lost from the module.

ISSN: 2044-3315

Acknowledgements

Prof. Shirley Williams for supporting the initial mobile phone workshop within the School.

Dr Giuseppe Di Fatta for agreeing to use the online material within the Java programming module.

Mr Pat Parslow for support and being a critical friend

References

Abelson, H., Chang, M., Friedman, M., Lomas, C., & Wolber, D. (2010). Workshop - Google App *Inventor for Android: Creating Mobile Applications as a First Computing Experience*. 40th ASEE / IEEE Frontiers in Education Conference (pp. W1C-1). Washington, DC: IEEE.

Appbrain.com. (2012, July 9). *Number of available Android applications*. Retrieved July 9, 2012, from AppBrain: http://www.appbrain.com/stats/number-of-android-apps

Apple. (2012, July 9). *iPhone App Store*. Retrieved July 9, 2012, from Apple: http://www.apple.com/iphone/built-in-apps/app-store.html

Biggs, J. (2003). Teaching for Quality Learning at University. In *The 3P Model of learning and teaching* (2nd ed., pp. 18-19). Buckingham: The Society for Research into Higher Education & the Open University Press.

Hsu, Y.-C., Rice, K., & Dawley, L. (2012). *Empowering educators with Google's Android App Inventor: An online workshop in mobile app design*. British Journal of Educational Technology, 43(1), E1-E5.

IDC. (2012). *Android- and iOS-Powered Smartphones Expand Their Share of the Market in the First Quarter.* FRAMINGHAM: IDC.

Lundqvist, K. (2011, June 14). *Android Game Development Tutorial*. Retrieved July 9, 2012, from Karsten's Space: http://oster-lundqvist.com/karsten/?p=4886

Thoman, D. B., Sansone, C., & Pasupathi, M. (2007). *Talking about interest: exploring the role of social interaction for regulating motivation and interest experience.* Journal of Happiness Studies, 8(3), 335-370.

ISSN: 2044-3315