Sustaining Student Retention of Computer Programming with mLearning Apps

Dr. EM Nalaka EDIRISINGHE
Lecturer, Temasek Polytechnic, Singapore
nalaka@tp.edu.sg

Abstract: The learning of computer programming fundamentals remains a challenge for novice learners across the world. Many different approaches have been attempted from more engaging languages to media computation. One area that has not been addressed yet is how to help students sustain their retention of this knowledge and apply it successfully in their current and future modules and careers. In addition, with the proliferation of smart-phones today's learners are demanding more access to immediate information to help them in their tasks. In this paper, we describe an approach leveraging upon mLearning strategies to help students sustain and apply their knowledge of fundamental programming concepts in the foundation modules. This paper shares the processes and lessons learned from a pilot project of developing customized, native mLearning applications on multiple platforms to address this gap. The paper describes the strategy and experiences of developing the mLearning apps, which included a multiple programming language reference and customized self-tests, and adopting them in the teaching of programming subjects. It discusses the students' and faculty's experiences and their implications.

Keywords: Mobile learning, technology, education, computer programming.

Introduction

Learning fundamental programming for novice learners during their first year of studies at higher education institutions remains a challenge. The attrition rate in such subjects and programs is quite high. Many efforts over the years have been made at tackling this problem through the use of a variety of tools and techniques with different levels of success. However, an area not often studied relates to how students successfully retain this initial burst of knowledge after their first fundamental programming module. The retention of this knowledge throughout their subsequent years of study remains a critical challenge. It is common to find students unable to recollect fundamental concepts learnt in their foundation years, particularly in the final year projects and subsequent careers. At the same time, today's higher education landscape has seen dramatic shifts in terms of teaching and learning, including the easy access to information, the increasing availability of mobile devices amongst learners, and the increasing demand for anywhere, anytime learning. As such, mobile learning and applications provide one platform upon which to tackle this issue of sustaining student retention of fundamental programming concepts. Thus, this paper intends to explore, on a pilot basis, the adoption of mLearning apps during the first year of studies to sustain retention of programming knowledge.

1. Literature Review

1.1 Learning Programming

Students learning programming for the first time face multiple challenges. Resnick (2009) indicated that novice learners faced challenges, such as mastering the syntax of the language and understanding the complexities of programming environments in which they must learn to write programming solutions. Other challenges include the lack of motivation and the inherent difficulty faculty face in engaging student through text-based programming languages (Kelleher, C. & Pausch, R., 2005). Many solutions have been adopted in the past through storytelling and tools, such as Alice, game-based learning and media computation (Kelleher, C. & Pausch, R. 2007). However, there has been inadequate work done into exploring how to facilitate student retention of their knowledge of programming using a specific mobile platform and strategy.

1.2 Mobile Learning

The proliferation of mobile technologies into our everyday lives continues at a rapid pace. Now more than ever learners at a younger age are using smart-phones that provide them with access to a world beyond the confines of their classroom. A recent study of students in higher education in China identified that 85.7% of the pool of mobile users actually access universities via their smart-phones (CNNIC 2010). As such, Rajasingham (2011) highlights that we, as educators, need to take into account the changing landscape and design for the needs of the new mobile savvy audience. One way the educational community has been meeting the needs of such an audience is through mobile learning. Mobile learning is defined as any form of learning that is conducted while the learner is on the go (Sharples et la, 2005). Conole (2008) stated that mobile technologies continue to be used in developing innovative pedagogical scenarios for learners. In the context of learning fundamental programming, different studies have been done to motivate students by allowing them to build mobile applications (Mahmoud, 2008, Spertus et al, 2010). Further, studies have shown it is useful to incorporate mobile learning as a form of blended learning (Shen, 2008). However, insufficient studies have been done on actually deploying customized mobile learning applications in a blended setting to support the student retention of fundamental programming concepts.

2. Background

Temasek Polytechnic is a tertiary institution with over 21 years of experience in conducting information technology related diplomas. All students who enroll in these diplomas every year take up a fundamental programming module. The objective of this module is to teach novice learners programming fundamentals. After this introductory programming module, students continue to take more advanced modules in the areas of object-oriented programming, algorithms and web development, all of which require a strong retention of the fundamental programming knowledge.

3. Subject

3.1 Rationale

Historically teaching the fundamental programming module at our institution has faced many challenges in terms of student motivation and engagement. In the past this gap had

been addressed through the use of different tools such as Alice and flow-charting. However, in recent years faculty members teaching advanced modules had highlighted that student retention of their fundamental programming concepts, such as writing iterative constructs or declaring arrays, was very poor. Faculty continued to face students who had succeeded in their early programming modules but could not sustain their mastery of the knowledge in advanced modules. As such, the teaching team decided to develop native mobile applications on the iOS and Android platforms to help students sustain their retention of fundamental programming constructs.

3.2 Mobile Strategy

In developing the native mobile applications the teaching team adopted a five principle mobile strategy. First, the mobile applications would be focused on providing students with "just-in-time" instruction and access to information. The rationale for selection of the "just-in-time" approach was that the learners would not only be using these applications in their first year of studies but during their subsequent years and careers. Second, the mobile applications would not be a substitute for existing courseware and e-learning content available on the institution's learning management system. Third, the mobile applications would follow an outcomes-based approach to learning. As such, the mobile applications would not simply be a repository of course content or PowerPoint notes. Instead, the mobile applications would give information to achieve certain learning outcomes. This was done in alignment to previous research that indicated that effective learning requires a student-centered, outcome-focused approach (Field, 2005). Fourth, the mobile applications would provide a simple user interface with minimal clicks to access the relevant content. Fifth, each mobile application included differentiated instructional self-tests, which were differentiated by content. Based on these five principles, two mobile applications were developed for the iPhone and Android markets. The mobile applications consisted of two modules: a reference module in multiple programming languages and a self-test module as shown in Figure 1.

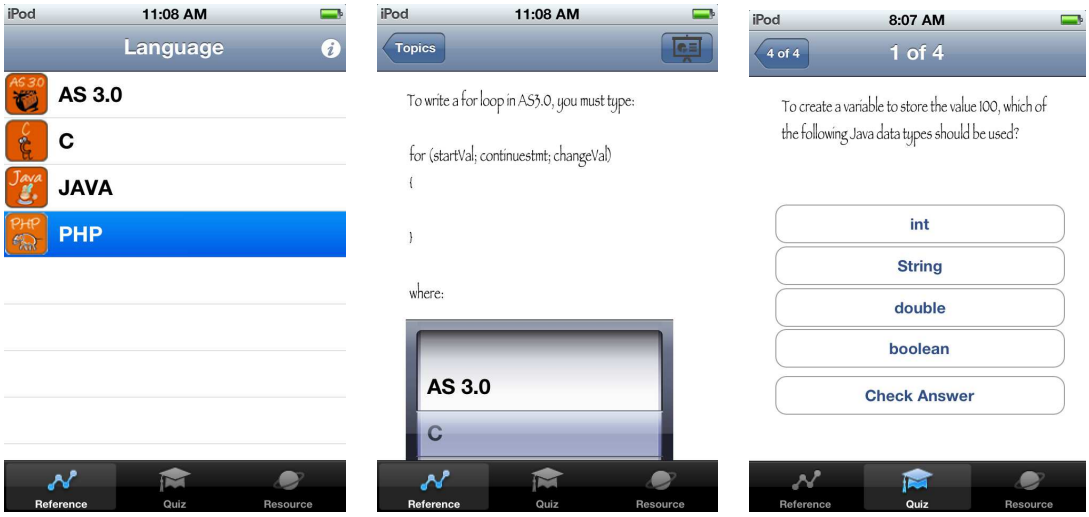

Figure 1: Example Screenshots of the iPhone mLearning Apps

The reference module listed key outcomes students needed to remember, such as declaring variables, writing loops, declaring arrays, identifying even and odd numbers and searching an array. To help students the reference module provided necessary key content for multiple programming languages, such as Java, C, PHP and ActionScript, that students would encounter in their advanced programming modules. The self-test module was

customized to help learners with different abilities remember the key content. Simple learning analytics was integrated as part of each self-test. The self-test module provided differentiated content for two levels of users. The differentiated content was determined through a pre-quiz assessment. Based on the scores achieved in the pre-quiz learners with different abilities would be sent to one of two possible paths. Each path would be customized with the necessary scaffolding and feedback to assist learners to master the content. As such the multiple paths provided learners with a degree of customization of their learning content.

4. Methodology

Since the project was a pilot study the methodology used was qualitative in nature. This pilot focused on the faculty and student populations. The faculty population consisted of those teaching advanced modules and the student population consisted of those taking advanced programming-related modules. The sample size consisted of two (2) faculty members and twenty (20) students. The faculty members and students were selected using simple random sampling. The instruments used in the study included the faculty members' journals, interviews with the students and interviews with the faculty members. The mobile applications were deployed over a period of 3 months and all journal data and interviews were collected and conducted at the end of this phase. The data analysis consisted of content analysis of the textual data and went through 4 distinct stages. In the first stage, the textual data was separated between the student and the faculty members. In the second stage, a partial set of the textual data was grouped together to identify common themes amongst each of the two groups in the sample population. In the third stage, coding was performed to derive inductive categories. In the fourth stage, the inductive categories were ranked based on the number of occurrences falling into each category from all the interview and journal data. As such, these inductive categories formed the results described in the next section.

5. Results

The results of this study can be categorized into two areas: the student perspective and the faculty perspectives.

5.1 Student Perspective

In terms of student perspectives, five key results were noted. First, the students found the mobile applications very helpful in reminding them of key content that they had just learnt when they needed it (i.e. just-in-time). This result was in alignment with previous studies that showed that mobile learning was a step towards making the educational process "just-in-time, just-enough, and just for me" (Peters, 2007). Second, the students liked the convenience with which to find the content without the need to research on-line or lookup their old lecture notes or textbooks. This implied that the mobile learning applications provided students with a time-saving element, thereby increasing their efficiency and productivity, which was aligned to previous studies related to offering students the flexibility to determine the conditions of their own learning (Field, 2005). Third, the students found the quantity of the content just right for their needs. This implied that the amount of content provided for each topic was a potential factor in students using their mobile learning applications. Fourth, the students found the multiple programming languages very useful in helping them navigate through new programming languages in the

future. This implied that students may potentially continue using these apps beyond their first year as they provided a wide range of programming languages that may be relevant in their subsequent years. Fifth, the students found the customized self-tests helped aid them in managing the pace of their own learning. This implied that learners do view personalized learning experiences as a potential factor to their buy-in of mobile learning applications.

5.2 Faculty Perspectives

In terms of faculty perspectives, three key results were noted. First, the teaching staff highlighted a very high adoption rate by students. It was observed that students were quick to use the mobile applications even inside the classroom when they needed to refer to something. Second, the teaching staff indicated that students were very comfortable with the mobile applications as it was considered just another app in their smart-phone. Third, the teaching staff indicated that the outcomes-based approach was very suitable for mobile learning applications as it helped promote the process of "just-in-time" learning. This implied that the development of content needed to be carefully done without simply transferring existing courseware or e-learning content onto a mobile learning application.

6. Conclusion

In conclusion, this pilot study potentially suggests that mobile applications for learning based on a specific strategy customized to the learners' needs can help in student retention of fundamental programming knowledge at the initial stage. However, further studies need to be done to measure student adoption over subsequent years and their effectiveness.

References

[1] Conole, G., de Laat, M. Dillon, T. & Darby, J. (2008). Disruptive technologies, pedagogical innovations : What's new ? Findings from an in-depth study of students' use and perception of technology. *Computers & Education*, 50, 511-523.

[2] CNNIC (2010). 26th Statistics Report for Internet Development in China.

[3] Field R. (2005). Favourable conditions for effective and efficient learning in a blended face-to-face / online method. Proceedings of ASCILITE 2005. Retrieved January 12, 2006, from http://www.ascilite.org.au/conferences/brisbane05/proceedings.shtml.

[4] Kelleher, C. & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computer Surveys*, 37(2), 83-137.

[5] Kelleher, C. & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, 50(7), 64.

[6] Mahmoud, Q. (2008). Integrating mobile devices into the computer science curriculum. Paper presented at the Frontiers in Education Conference, 2008. FIE 2008. 38th Annual.

[7] Peters, K. (2007). m-Learning: Positioning educators for a mobile, connected future. *International Journal Of Research in Open and Distance Learning*, 8(2), 1-17.

[8] Rajasingham, L. (2011). Will Mobile Learning Bring a Paradigm Shift in Higher Education. *Education Research International*, Volume 2011.

[9] Resnick, M., Maloney, J. Monroy-Hernandez, A. Rusk, N. Eastmond, E. Brennan K., et al. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.

[10] Sharples, M. Taylor, J. & Vavoula, G. (200). Towards a Theory of Mobile Learning. Paper presented at the mLearn 2005, Cape Town, South Africa.

[11] Shen, R., Wang, M. & Pan. X (2008). Increasing interactivity in blended classrooms through a cutting-edge mobile learning system. *British Journal of Educational Technology*, 39(6), 1073-1086.

[12] Spertus, E., Chang, M.L., Gestwicki, P. & Wolber, D. (2010). Novel approaches to CS 0 with app inventor for android. Paper presented at the Proceedings of the 41st ACM technical symposium on Computer Science Education.