Teaching Introductory Programming using SCRATCH: Learning By Game Building

Amitrajit Sarkar
Christchurch Polytechnic
Institute of Technology
Madras Street, Christchurch
Amitrajit.Sarkar@cpit.ac.nz

Rob Oliver
Christchurch Polytechnic
Institute of Technology
Madras Street, Christchurch
Rob.Oliver@cpit.ac.nz

Dr Michael Lance
Christchurch Polytechnic
Institute of Technology
Madras Street, Christchurch
Michael.Lance@cpit.ac.nz

ABSTRACT
This poster examines the benefits of building computer games for students studying in Introductory Programming courses and using these games as a vehicle to teach computer science concepts.

Keywords
Game Building, Introductory Programming, SCRATCH, visual programming, participation, engagement.

1. INTRODUCTION
Scratch is a visual programming environment that was developed by the Lifelong Kindergarten group at the MIT Media Laboratory. Scratch is widely used by young people around the world and it is commonly used in schools at all levels. Lecturers at universities and tertiary institutes have taken to using Scratch in Introductory Programming courses before progressing into programming in commercial languages.

Our entry-level programming students grew up playing games from infancy and these 'digital natives' were raised in a digital environment surrounded by highly interactive games. They expect technology everywhere, including the classrooms. Students often complain that they cannot see the relevance between tertiary assignments and real life. At CPIT we are using SCRATCH in various introductory programming courses.

2. METHODOLOGY
We are trying to teach Scratch differently to how the constructivists have done it in past. Academics have started questioning the constructivist way of teaching introductory programming and challenging the basic premise, that putting introductory students in the position of discovering information for themselves is a bad idea. (Kirschner, Sweller, & Clark, 2006)

In our degree program, the Bachelor of Information and Communication Technologies (BICT) we start our Introductory Programming paper with 5 weeks of SCRATCH and target the following basic learning objectives:

1. Introduce students to the concept of iteration and loops in Software Engineering.
2. Introduce students to the concept of conditional statements in Software Engineering.
3. 3. Introduce students to the concept of Objects - Sprites have state (such as position) and behaviors.
4. 4. Allow students to experiment with advanced iterations, loops and conditional statements by creating the Theseus and Minotaur game.

3. CONTENT
The Theseus and Minotaur game rules were discussed and web sites visited that explained the history behind the game. The students visited the Java version of the game website and played the game, initially at level 1 only. The game logic was discussed and the students designed the solution with pen and paper first. Understanding the algorithm and designing the solution is way too important to ignore and is part of the course learning outcomes. Students who have gone beyond level 3 have acknowledged the importance of design. Once the design activity is completed the students then used Scratch to implement their design.

Examples of in-line commenting, decomposition and refactoring were identified. Worked examples were discussed and the Level 1 solution of the game created. This became a starting point for creating other levels of the game.

4. CONCLUSION
Our preliminary observation confirms that games are powerful medium for learning programming. We observed that Game-Based learning has impacted positively on the following learning attribute:

- Participation
- Engagement
- Relevance
- Confidence
- Satisfaction/Fun
- Game Designing
- Meeting the Learning Objectives

Initially we asked the students to do the first three levels of the game but gave additional challenge and rewarded them if they had completed all the 10 levels.

We observed exceptional levels of motivation to participate in discussions and attempt to write the code.

We also observed a great deal of code reusability.

5. REFERENCES
Kirschner, P., Sweller, J., & Clark, R. (2006). Why minimal guidance during instruction does not work:an analysis of the failure of constructivist, discovery,problem-based,experimental, and inquiry-based teaching. *Educational Psychologist*, 75-86.

This poster paper appeared at the 3rd annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2012) incorporating the *25th Annual Conference of the National Advisory Committee on Computing Qualifications*, Christchurch, New Zealand, October 8-10, 2012. Mike Lopez and Michael Verhaart, (Eds).