

Problem Based Learning using SCRATCH: Pattern Orientated Instruction

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

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

ABSTRACT

This poster examines the resources that had to be developed to allow the programming language SCRATCH to be used to teach Introductory Programming to students enrolled in both the Diploma in Information and Communications Technology (DICT) and the Bachelor of Information and Communications Technologies (BICT) courses. Challenges met focused on the fact that none of the games programs conformed to any of the programming standards taught by the department and no satisfactory business programming examples appeared to exist.

KEYWORDS

Game Problems, Introductory Programming, SCRATCH, pattern types, solution design, desk checks.

INTRODUCTION

Entry level programming in the Diploma in Information and Communications Technology (DICT) consists of a mixture of game problems (forty per cent) and business problems (sixty per cent). All programming theory revolves around a set of standards. More specifically the structure and logic is built around a set of design patterns. The conversion to SCRATCH as the Introductory Programming language necessitated a complete redevelopment of the business program models and a development of gaming models. The game models were based on the standard SCRATCH [1] games supplied with the SCRATCH installation.

METHODOLOGY

All problem solutions and algorithm designs revolve around the ideas of 'Pattern Matching'. Learning and teaching revolve around the philosophy of 'use this design and the solution will work'. Standard Program Development Life Cycle steps are used in all algorithm design. Students are taught that if they can understand the program requirements then they should be able to

match that requirement with an appropriate pattern. The pattern gives a macro view of the logic steps. In addition to this all solutions consist of three major subprograms and overarching control logic.

CONTENT

Computer Science methodologies are taught by a mixture of theory and the development of working models based on common problems found in the business and game environment. Coding standards are demonstrated and developed in combined lecture and laboratory sessions. These standards have been developed by the department over a number of years. Procedural design concepts are developed and applied to the Object Oriented drag and drop SCRATCH model. Game examples are used to introduce the student to fundamental concepts. At this stage we take some of the standard SCRATCH game code and refactor it to our standard structure and design patterns. Later business problems are introduced.

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
- Group work

REFERENCES

[1] <http://scratch.mit.edu/> for the SCRATCH game '2 ideas'.

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).