

C++ Programming Games (CPG) as Supplementary Method to Learn C++ Syntax

R. M. Hanifa, S. Mohamad, I. A. Bahrudin, S. H. K. Alan and S. R. Sinnasamy

Abstract—Programming is a fundamental skill that all computer science and information technology students are required to learn. At Universiti Tun Hussein Onn Malaysia (UTHM), C++ programming is one of the compulsory courses offered to first year students taking Diploma in Information Technology (DIT). The course is taught during their first semester and as a result, students have difficulties to pass the course due to their lack of knowledge on programming as freshmen. Teaching fundamental programming to freshmen is challenging and academics have attempted in finding suitable methods to enhance students' interest to do further exploration and self-experimentation. Realizing the importance of programming courses especially C++, a C++ Programming Game, also known as CPG, has been designed as a complementary learning aid to attract students' attention with the use of hangman concept in the game. Hangman is a fun and simple game. The score will be published at the end of every session to rate students' understanding. A thorough explanation is given if the answer is wrong to help them reflect on their mistake. The questions are generated randomly for every session. A group of about 56 students in the first year of DIT at UTHM were offered to test out the game. The results show that more than 60% of the students found the games to be useful as an additional method to help them better understand the syntax of C++.

Index Terms—C++ Programming, Educational Games, Spiral Model, Teaching Aid

1 INTRODUCTION

TECHNOLOGY has created and transformed the learning and teaching processes, which in turn brings new opportunities to the educational system. According to Olapiriyakul and Scher [1], technology has become crucial in educational development and for the revolution in learning system. Programming is an art and it requires the individuals' ability to interpret challenges into solutions [2]. Knowledge of programming tools and languages, problem-solving skills, and effective strategies for program design and implementation are needed in the art of programming [3]. C++ programming language is one of the compulsory courses offered to first year student taking Diploma in Information Technology (DIT) at Universiti Tun Hussein Onn Malaysia (UTHM). Most of the students have difficulties to get good results due to their lack of knowledge on programming as freshmen. Besides, the course is taught during their first semester and this is very challenging for the academics to teach programming course due to time pressure and syllabus

constraints.

Students often give up as they find programming course to be too difficult to understand [4]. A study conducted by Garner [5] found that looking on lengthy codes, caused students to lose interest in programming language or they may carry on without comprehending the basic. According to Robin et al [6], it takes about 10 years of experience to turn a novice into an expert programmer. This indicates how challenging it is to teach programming course and usually the option taken by most academics is to force students to learn by heart or to lower the passing mark [7]. Teaching and learning programming have become more complex and they require the development of a teaching aid that integrates virtual and concrete experiences such as technology assisted learning.

2 RELATED WORK

Programming courses are found to be difficult and often have the highest dropout rates [7]. Learning theory alone will expose students to learning content they are likely to retain information and therefore improve their learning and understanding. But, through games, the students will be more motivated and at the same time have fun. For example, Graven and MacKinnon [8] have developed a prototype of game-based environments for learning C++ programming which has been tested to a group of first year students learning C++ programming. Questionnaires have been distributed to C++ students in Scotland and Norway. The results show that Scottish students have an average knowledge on C++; they enjoy programming and find computer games more enjoyable. However, they have limited experience with computer games designed

-
- R. M. Hanifa is with the Centre of Diploma Studies, Universiti Tun Hussein Onn Malaysia, Malaysia, Johore 86400. E-mail: rafizah@uthm.edu.my
 - S. Mohamad is with the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Malaysia, Johore 86400. E-mail: shamsulm@uthm.edu.my
 - I. A. Bahrudin is with the Centre of Diploma Studies, Universiti Tun Hussein Onn Malaysia, Malaysia, Johore 86400. E-mail: ariyani@uthm.edu.my
 - S. H. K. Alan is with the Centre of Diploma Studies, Universiti Tun Hussein Onn Malaysia, Malaysia, Johore 86400. E-mail: Masahiro.alan@gmail.com
 - S.R. Sinnasamy is with the Centre of Diploma Studies, Universiti Tun Hussein Onn Malaysia, Malaysia, Johore 86400. E-mail: surain.wira@gmail.com

for learning. Norwegian students, on the other hand, have below average knowledge on C++ and they do not enjoy programming but they have experience with computer games.

According to Bjoerner and Hansen [9], the problems in designing games include the lack of learning contents or the lack of game contents. Besides, educational games do not provide a continual balance between the challenge and the students' skill to solve the given problem. In their research, they suggested three different perspectives to be communicated across in order to design a useful educational game: teacher, students and game designers. They used a holistic perspective when designing the games which involved integration of thinking, feelings, perceiving, behaving, culture and context from game designers, teachers and students. Their findings show that different perspectives offer different things on different levels and can be used to set some design principles.

The research conducted by Depradine [10] is part of an ongoing effort to improve teaching of fundamental object-oriented programming that use computer games as a mechanism in helping the performance of students in both coursework and examinations, when compared to previous years. A preliminary investigation on the use of gaming as an active learning strategy to improve student performance was carried out over the 2010 to 2011 academic year. The study focused on the actual gaming activities and how they were used in the delivery of the course. The overall results for the academic year 2010 to 2011 exceeded all prior years with significant improvements in the number of grades A and B. Corresponding reductions in grades D and F were also achieved. The improved results in the final exam, especially in the low number of failure reveal that students were able to perform well under exam conditions, demonstrating that the material learned during the course was successfully applied.

Sarpong et al [2] in his paper stated that there are many factors that influence the high rate of failure of students in computer programming courses. Their paper focused on teaching methodologies and strategies implemented in teaching programming courses. A survey was conducted to investigate teaching methods adopted by teachers of computer programming in Ghana. They tried to identify the best teaching practices that would enhance the teaching of programming courses as well as improve the teaching environment in order to create a conducive atmosphere for learning. The results show that the main problems identified as contributing to high students' failure in programming courses are the teaching methods and strategies. Hence, they proposed teaching methods and strategies which are in line with the findings of Miliszewska and Tan [11] which is to select an appropriate teaching method or blend of methods to help in delivering knowledge to students.

Wallas [12] suggested five characteristics of a good interface when designing an application: responsiveness,

familiarity, clarity, concision and forgiveness. Having a good interface is very important since it presents what is intended to be conveyed to the user. Realizing the importance of technology in education, an innovative game has been designed to captivate students' interest and enhance their understanding on the syntax of C++ programming to make the process of learning and teaching to be more exciting. Thus, this paper will adopt technology as a guideline for designing CPG as a teaching and learning aid for students taking C++ programming.

3 METHODOLOGY

The development of CPG is based on the utilization of the Spiral Model which combines the idea of iterative development with the systematic, controlled aspects of the waterfall model [13]. Iteration means the steps are performed over and over again. The development of the game is conceived, designed, prototyped, tested and the test results are put into the next iteration of the same steps [14]. Basically there are five phases in the Spiral Model: define, design, prototype, playtest and feedback.

During the first phase, the definition of educational game named CPG is defined. The design phase is where the storyboards for CPG are prepared. Rules and procedures for the games are also written during this phase and the basics for a workable game are made. Next, a deliverable or prototype is produced which is closer to the final product. Prototype is a working model that is built and starts to get the look and feel of the physical features of the game. Playtest is where the game is tested to get feedback from the user so that improvements can be made before the actual game is released.

3.1 Development of CPG

CPG implemented the concept of hangman in the game to make it more exciting. Besides, the hero icon used in the game is self-drawn to give a sense of excitement in learning. There is also background music in the game. Music is played while the game progresses. Music loops are essential elements in the creation of games. It is almost as important as the game itself. Figure 1 shows the main interface of the game.



Fig. 1. Main interface of CPG

Clicking the Start button, the student will be given 3 option buttons as can be seen in figure 2. The three buttons are City B for beginners, City I for intermediate and City A for advance. Once the option is opted, the question interface will appear (see figure 3). Questions asked are based on the level of difficulties and the questions are generated randomly during each usage. The type of questions prepared are multiple choice questions, true and false questions, matching questions, spot the error questions and fill in the blank questions.



Fig. 2. Option buttons to be chosen



Fig. 3. Sample of questions

For each option selected, students are allowed to make 5 errors only. Each time the error is made, the hero icon which is divided into five main sections will be displayed: head, body, leg, chair and the removal of the chair. The example can be seen in figure 4. For each incorrect answer, the detailed explanation is given. The game is over when the chair is being removed which means the hero is totally hung (refer to figure 5). Towards the end of every session, the score board will show the score as in figure 6. This can be used to rate the student's understanding on C++ programming.



Fig. 4. Head is drawn upon incorrect answer



Fig. 5. The game is over



Fig. 6. The scoreboard showing the results

3.2 Trials

A group of about 56 students in the first year of Diploma in Information Technology at UTHM were offered to test out the game to evaluate the game developed. The students are given 3 minutes of introduction to the game and procedures on how to navigate the game. They are also given a questionnaire with 10 simple questions that took 2 to 3 minutes to fill in. Feedback from the first year student was sought to evaluate their perceptions on the

usefulness of the game in helping them to understand C++ programming syntax.

3.3 Results and Discussion

The questionnaire collected from the students are analyzed and the results of the analysis are shown in Table 1. The table shows that basically, the overall average of students who strongly agreed with the CPG being effective is 31.43% and those who agreed is 56.43%. This percentage is quite high as most of the students are very excited with the game innovation produced since there is no other courses that provide similar teaching aid. The average for students who are not sure is just 8.04%, while 2.14% disagree and only 1.96% strongly disagree. This clearly shows that the majority of students are satisfied with the game developed and agreed that CPG could help them to understand better the syntax of C++ programming (98.21%).

TABLE 1. PERCENTAGE OF STUDENT VIEWS ON CPG

Question	Strongly Agreed (%)	Agreed (%)	Not Sure (%)	Not Agreed (%)	Strongly Not Agreed (%)
1	21.43	66.07	8.93	1.79	1.79
2	17.86	73.21	5.36	3.57	0.00
3	23.21	71.43	1.79	0.00	3.57
4	37.50	35.71	17.86	8.93	0.00
5	53.57	35.71	10.71	0.00	0.00
6	25.00	69.64	0.00	0.00	5.36
7	32.14	62.50	5.36	0.00	0.00
8	33.93	53.57	3.57	3.57	5.36
9	19.64	48.21	25.00	3.57	3.57
10	50.00	48.21	1.79	0.00	0.00
Average	31.43	56.43	8.04	2.14	1.96

4 CONCLUSION

The development of CPG is an effort to diversify the teaching and learning materials in the Department of Information Technology, Center for Diploma Studies, UTHM. In addition, through the study and discussion based on students' feedback through questionnaire survey clearly showed this CPG is acceptable because it is able to attract students' attention, enhance their understanding as well as motivate them to learn the course better. Besides the positive responses from the respondents, the findings suggest that students' enthusiasm to learn programming can be enhanced by using innovative ways in teaching such as using games rather than the standard or traditional teaching approach.

ACKNOWLEDGEMENT

The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) for funding this research.

REFERENCES

- [1] Olapiriyakul, K. and Scher, J. M. (2006) A guide to establishing hybrid learning courses: employing information technology to create a new learning experience, and a case study. *The Internet and Higher Education*. Vol. 9, Issue 4, pp. 287-301.
- [2] Sarpong, K. A-M., Arthur, J. K. and Owusu, P. Y. (2013) Causes of Failure of Students in Computer Programming Courses: The Teacher – Learner Perspective. *International Journal of Computer Applications*. Vol. 77, pp. 27-32.
- [3] Ala-Mutka, K. (n.d) Problems in learning and teaching programming. A literature study for developing visualizations in the Codewitz-Minerva project. pp. 1-13.
- [4] Boulay, B. D. (1986) Some difficulties of Learning to Program. *Journal of Educational Computing Research*. Vol. 2, pp. 57-73.
- [5] Garner, S. (2003) Learning Resources and Tools to Aid Novices Learn Programming. *Informing Science*.
- [6] Robins, A., Rountree, J. and Rountree, N. (2003) Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*. Vol. 13, pp. 137-172.
- [7] Esteves, M. et al. (2010) Improving teaching and learning of computer programming through the use of the Second Life virtual world. *British Journal of Educational Technology*.
- [8] Graven, O. H. and MacKinnon, L. (2009) Prototyping Games-Based Environments for learning C++ Programming. *HCI Educators*.
- [9] Bjoerner, T. and Hansen, C. B. S. (2011) Designing an Educational Game: Design Principles from a Holistic Perspective. *The International Journal of Learning*. Vol. 17, pp. 279-289.
- [10] Depradine, C. A. (2011) Using Gaming to Improve Advanced Programming Skills. *Caribbean Teaching Scholar*. Vol. 2, No. 2, pp. 93-113.
- [11] Miliszewska I. and Tan G. (2003) Befriending Computer Programming: A Proposed Approach to Teaching Introductory Programming Course. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, pp. 132-136.
- [12] Wallas, P. (2014) Five Characteristics of a Great User Interface. Retrieved July, 24, 2014 from <http://paulwallas.com/five-characteristics-of-a-great-user-interface/>
- [13] SDLC Spiral Model. Retrieved July, 30, 2014 from http://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm
- [14] Game Creation Guidelines/Spiral Model of Game Development. Retrieved July, 30, 2014 from http://en.wikiiversity.org/wiki/Game_Creation_Guidelines/Spiral_Model_of_Game_Development



R. M. Hanifa obtained her bachelor degree in Computer Science from Universiti Sains Malaysia (USM) in 1999, followed by master degree of Information Technology, Faculty of Information Technology, Universiti Utara Malaysia (UUM), Sintok, Malaysia in 2001. Currently, she is a lecturer at Center for Diploma Studies, Universiti Tun Hussein Onn Malaysia (UTHM), Johor. Her research interests include artificial intelligent, information system and augmented reality.



S. Mohamad received his BSc in Computer Science from Universiti Teknologi Malaysia in 1999 and MSc in Computer Science from Universiti Sains Malaysia in 2004. Now, he is PhD student at Faculty of Computing, Universiti Teknologi Malaysia focusing on Crowd Simulation research.



I.A. Bahrudin obtained her bachelor degree in Electrical Engineering from Universiti Tun Hussein Onn Malaysia in 2003. Currently, she is an Instructor (J41) at Center for Diploma Studies, Universiti Tun Hussein Onn Malaysia. She is a member of Board of Engineers Malaysia (BEM) and has Amateur Radio Operator Certificate (Class B).

S.H.K. Alan and S.R. Sinnasamy is a final year students in Diploma Information Technology at Universiti Tun Hussein Onn Malaysia.