TEACHING COMPUTER PROGRAMMING USING GAMIFICATION

Siew Peng Shorn

School of Engineering, Nanyang Polytechnic, Singapore

ABSTRACT

There are many challenges in teaching computer programming: the diversity in students' ability and aptitude levels; the time-consuming nature of programming; and the difficulty in motivating students to learn computer programming. Gamification refers to the application of gaming elements to non-game context, such as education, with the goal of increasing the engagement of students and inspiring them to continue learning. This paper presents the methodology of incorporating gamification elements in the teaching of computer programming and investigates the effects of gamification on students' learning gains and interest in learning computer programming at the School of Engineering in Nanyang Polytechnic, Singapore. Key findings on the extent gamification supports students' learning gains and interest in learning computer programming will be shared. Finally, the challenges faced in planning and designing appropriate educational games to teach computer programming will also be highlighted.

KEYWORDS

Gamification, Learning Gains, Interest in Learning, Computer Programming, Higher education, Tertiary education, Standards: 8

Note – In the context of Nanyang Polytechnic, the term 'course' refers to a 'program' while the term 'module' refers to a 'course'. For example, *Diploma in Electronic Systems* is a course; *Computer Programming* is a module.

INTRODUCTION

Generation Z youth is technology-savvy. They have digital technology and internet technology readily available to them at very young age and they are exposed to games or gamified activities that are available on their mobile phones and computers. The Generation Z youth also engages and maintains various social media platforms such as Facebook, Twitter and Instagram. They like to have fun and prefer non-verbal communications using digital technology than verbal communications. However, they tend to have short attention spans. The very nature of the Generation Z youth posts challenges in motivating them to learn computer programming, which is often perceived as a boring, time-consuming and difficult module.

Gamification, the use of game design elements in non-game contexts (Deterding, Dixon, Khaled, & Nacke, 2011), might overcome the challenges faced in the teaching of computer programming. As gamification uses "game-based mechanics, aesthetics and game thinking to engage people, motivate action, promote learning, and solve problems" (Kapp, 2012), it provides an avenue for capturing the attention spans of people. The goal of gamification is to maximize enjoyment and engagement through capturing the interest of learners and inspiring them to continue learning (Hwang, Hong, Cheng, Peng, & Wu, 2013).

Active learning increases student performance in science, engineering and mathematics (Freeman, et al., 2014). Gamification, a form of active learning, is gaining ground in education. Some research showed that, if gamification was properly applied, it could improve attendance (Fotaris, Mastoras, Leinfellner, & Rosunally, 2016) and engagement (Leong, Koh, & Razeen, 2011) , enhance understanding and consequently enhance performance (Mekler, Brühlmann, Opwis, & Tuch, 2013) . A study using an online game layer in teaching introductory programming found that gamification could significantly improve student engagement (Leong et al., 2011). Another study found that a gamified learning approach using a combination of “Kahoot!”, “Who wants to be a millionaire” game and “Codecademy” for computer programming class was motivating and enriching for both students and instructors (Fotaris et al., 2016).

Learning to program is difficult, especially for novice programmers (Piteira & Costa, 2013). Gamification may offer opportunities in solving these issues. Most studies focused on the engagement and motivation factors of gamification. This paper studied the effects of the gamified learning on students’ situational interest and learning gains in computer programming in a pilot study at the tertiary engineering education.

METHODOLOGY

The study focuses on the effectiveness of using gamification in teaching computer programming module, a module offered in the first year of two engineering courses at Nanyang Polytechnic, Singapore. The module was taught using C programming language. Two groups of students with similar profile, the experimental group and the control group, were identified. Gamification was applied in lectures and tutorials for teaching the experimental group while only traditional methods were used in teaching the control group. At the end of the semester, a common test and a survey were conducted for both groups to examine students’ academic performance and learning gains. For the experimental group, a survey on situational interest was also conducted to examine students’ interest in computer programming after experiencing gamified learning approach. Results were analyzed using quantitative methods.

Participants

Two groups of first year engineering students who were registered for the Computer Programming modules were invited to participate in the study conducted in academic year 2017 semester 1. The experimental group was from the Aerospace/Electrical/Electronics Programme (AEEP) and the control group was from the Diploma in Electronic Systems (DES). The module was taught over a period of 15 weeks. There were 86 AEEP students and 105 DES students in total in academic year 2017. These two groups have comparable academic results in the previous academic year 2016 semester 1 as shown in Table 1.

Table 1. Academic Results in Academic Year 2016 (before Action Research)

Group	Number of students	Mean Score	Standard Deviation
AEEP	92	68.8	12.7
DES	145	69.1	13.2

Control Group Class Setup

The control class had a weekly 2-hour practical session in a computer laboratory and a weekly 2-hour lecture & tutorial session in a lecture room. At the end of the semester, the students were required to complete a mini-project over a 3-week period.

Experimental Group Class Setup

Similar to the control group, the experimental group also had a weekly 2-hour practical session in a computer laboratory and a weekly 2-hour lecture & tutorial session in a lecture room. At the end of the semester, the students were also required to complete a mini-project over a 3-week period.

For the experimental group, gamified activities were introduced during the 2-hour lecture & tutorial session. Both team-based competitive games and individual competitive activities were conducted for the experimental group. All the games are directly linked to the module learning outcomes and used to replace tutorial questions with the same learning outcomes.

A total of 5 team-based games were conducted on a fortnightly basis. Students were asked to group themselves in teams of 3 to 4 members per team. Throughout the entire semester, students remained in their own teams. In each game, teams were ranked according to their order of completing the game correctly. Immediate feedback was given to students for errors in completing the game so that students would learn from their mistakes or misconceptions.

The first team-based game is a matching game. Each team was given a mixture of 14 different pseudocodes and 14 different flowcharts. Among the given pseudocodes and flowcharts, there were some containing errors and some that do not match. Students were required to compete among themselves in teams to identify two pairs of matching pseudocodes and flowcharts.

In the second game, a box containing many C identifiers as shown in Figure 1. Students were required to pour out all the given identifiers and identify the correct C identifiers. There was no information given on the total number of correct C identifiers present in the box.

Figure 1. Competitive Game to identify the correct C identifiers

The third game focused on arithmetic and logical operations in C programming language through the use of magic square puzzle (see Figure 2). Each team in the experimental group was given a 3x3 magic square to solve. A 3x3 magic square was an arrangement of the numbers from 1 to 9 in an 3 by 3 matrix, with each number occurring exactly once, and such that the sum of the entries of any row, any column, or any main diagonal was the same. The magic square that was given to students contained some cells already initialized with numbers. Students needed to solve the magic square puzzle by filling up the remaining cells with integers 1 to 9 without repeating any of the numbers. They were required further to fill up each cell with the arithmetic or logical expression that evaluated a value equal to the number in each cell. While students enjoyed solving the magic square puzzle, they also applied the arithmetic and logical operations in C programming language.

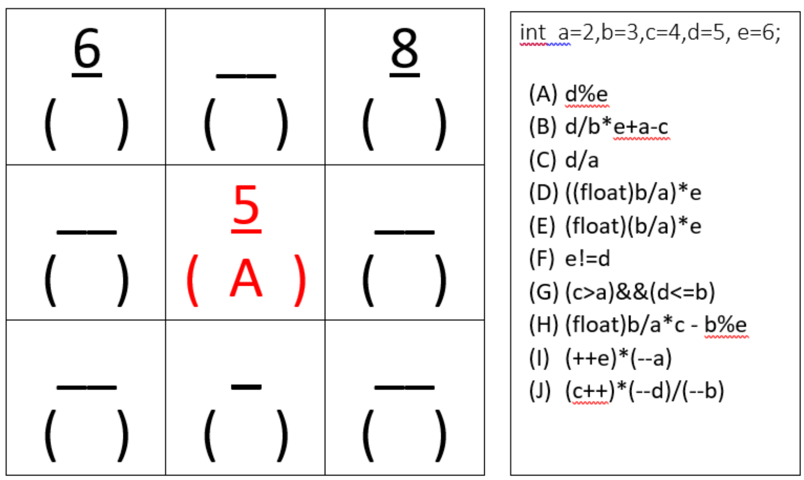

Figure 2. Magic Square Puzzle

In the fourth game, each team was given a short program written using arrays and a set of 20 resistors of different values. By understanding the program, each team had to identify the correct resistor from the given set. Students learnt about arrays and string functions through this game.

The fifth and last team-based game attempted to make a personal connection to students' enjoyable moments during the freshman orientation. During the orientation week before the start of the academic semester, students participated in a challenge that required them to construct a paper plane in teams. Students had a lot of fun and laughter during the orientation challenge. . In this last team-based game, teams were required to write program codes using loops to display a phrase such as "I enjoyed making and flying paper airplanes during orientation."

Besides the five team-based games, short individual quiz-like games were also conducted during the 2-hour lecture-tutorial sessions. In these short quizzes, students were required compete among themselves to identify the errors in a given program displayed onto the screen. There were multiple errors in each program. Each student was limited to one maximum correct attempt per session in order to give opportunities to more students. Figure 3 shows an example of an individual quiz-like game.

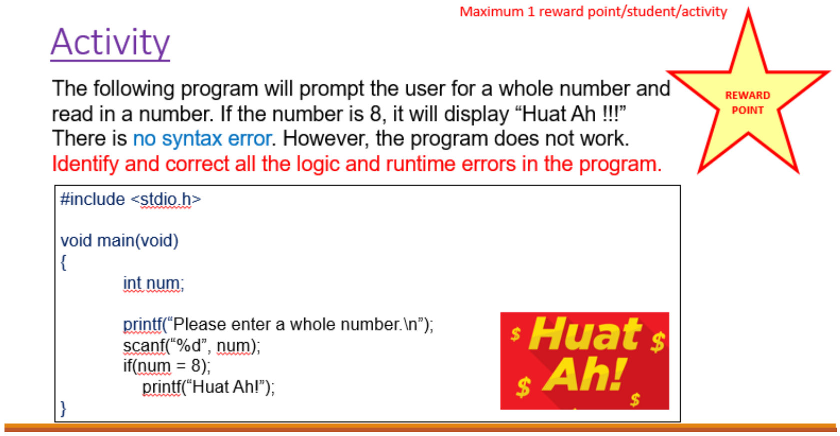

Figure 3. An example of an individual quiz-like game

An animation game from an online gamified programming learning website (www.codingame.com) to teach C programming loops was also used in the module. The animation game chosen was called "The Descent". A screenshot of the game is shown in Figure 4. In this animation game, a default non-working program was given. The animation story was about a spaceship called "Enterprise" that was going to land on the surface on a planet. There were 8 mountains of random heights on the surface. While the spaceship was landing, it must destroy the mountains in descending order of their heights, failing which the spaceship would crash into one of the mountains. Students learned about programming loops while playing the animation game.

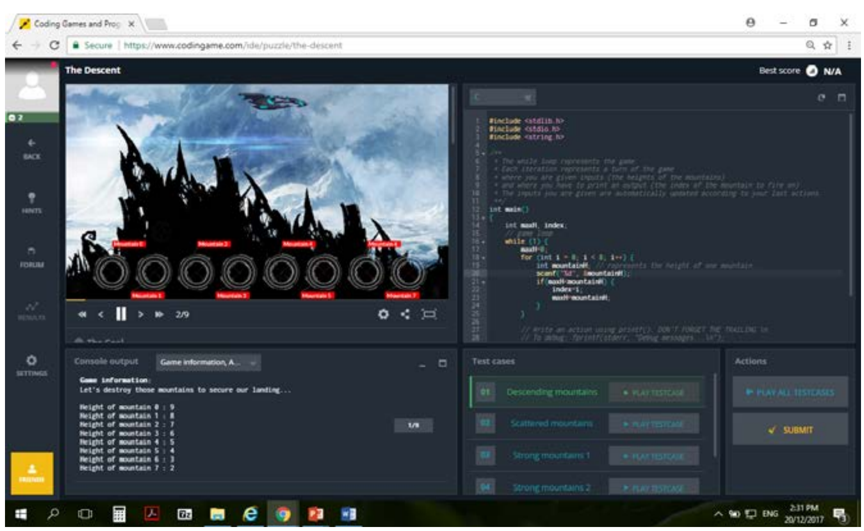

Figure 4. Screenshot of the animated Game "The Descent"

Points, Rewards & Leaderboard

Games have leaderboards and rewards (Glover, 2013). In this study, reward points and participation points were given. Reward points were given out for both team-based games and short individual games. For the team-based games, all the teams were ranked according to the order of completing each game correctly. Top three teams were awarded one reward point each in each game. In the individual games, one reward point was awarded to each correct answer, and each student was limited to one correct attempt per individual game. Participation points were also rewarded to students who participated in the team-based games.

A leaderboard was created and updated in Blackboard Learning Management System weekly. The purpose was to reflect the rankings of each student based on the reward points he/she had captured in class in these competitive activities. Their rankings were also displayed during the weekly lectures. The leaderboard with weekly rankings aimed to motivate students to move up in the leaderboard. Leaderboards served as a source of motivation for students because they saw their work publicly and instantly recognized, and because they could compare their progress with other classmates (Dominguez et al., 2013).

At the end of the semester, the total reward score for each student was computed. The maximum reward score for each student was capped at eight points. Top individuals with the highest reward score were awarded with individual prizes, and the best team with the highest reward score was awarded with team prizes. Both reward points and participation points were recorded and formed part of the overall score of the module.

CHALLENGES

In designing gamification elements into the module, one needs to take into consideration the diversity in the student population. Often, a class consists of academically strong and weak students, motivated and disinterested students, social and solitary students. To address this diversity, a mixture of individual games and team-based games were developed in this study. Total reward points were capped for each student. Reward points for individual games were also capped for each student for each individual game to provide opportunities to other students.

In forming the teams, students are allowed to form their own teams but there is a need to ensure each team has a good mix of academically strong and weak students. This allows the weaker students to learn from better students.

Different types of team-based games were created in this study and they were not repeated for different topics for novelty purposes. The process of having to create different types of games and thinking of the types of games that suit the topics is challenging and may take several days for each game. However, once the games were designed and created, re-using them is easy and does not require much time and effort. Unless automated, regular updates of reward points and leaderboard requires time and effort.

RESULTS AND DISCUSSION

A common test and a survey were conducted for both experimental and control groups at the end of the semester.

Students' Performance in Common Test

A common E-Quiz test was conducted for all the students in both the experimental and control groups at the end of the semester to evaluate the effectiveness of gamification in students' learning gains. The test consisted of 35 multiple-choice questions and covered all the topics in the module. It is used as a proxy for academic performance in this study. The results (normalized to a base of 100 marks) are shown in Table 2.

Table 2. Results of Common E-Quiz Test

	Sample Size	Mean Score	Median Score	Standard Deviation
Experimental Group	86	67	69	17
Control Group	105	54	54	17

Hypothesis testing is carried out to evaluate the effectiveness of gamification in enhancing students' academic performance measured using the common test results. Based on these data, it is concluded that it is statistically significant ($p < 0.01$) that an average student with gamified learning activities scored better than an average student without gamified learning activities. The quantitative results provide strong direct support for the hypothesis that gamification is effective in enhancing students' academic performance.

Cohen's d, the effect size of the mean score for measuring the magnitude of difference in mean between experimental group and control group is computed to be 1.06. Cohen provided rules of thumb for interpreting effect sizes, suggesting that |.1| represents a 'small' effect size, |.3| represents a 'medium' effect size and |.5| represents a 'large' effect size. Our results clearly indicates a large effect on the use of gamification in the improvement of mean score.

Post-survey

A common online post-survey on student assessment of learning gains was conducted for all the students in both the experimental and control groups at the end of the semester. The survey was voluntary. The survey items were adopted from Student Assessment of Learning Gains, a framework for measuring student learning gains and engagement (Lim, Hosack, & Vogt, 2012). In the survey, participants were asked about their understanding level on the concepts, applications, interest and confidence level. A 5 points likert-scale ranging from "Not at all", "Just a little", "Somewhat", "A lot" to "A great deal" was used. Among a total of 86 students in the experimental group, 50 students responded to the post-survey, representing a response rate of 58.1%. For the control group, 49 students out of a total of 105 students responded, with a response rate of 46.7%. Results of the post-survey are shown in Table 3.

The survey results showed that the experimental group has a slightly higher mean values than the control group across all the categories. It means that the experimental group students perceived themselves with higher understanding of concepts and greater gain in attitude, confidence and applications than students from the control group. However, these results are not statistically significant at significance level of 0.05 and of small effect sizes.

Table 3. Post-survey results

		Experimental Group	Control Group	Statistical Significance	Effect Size
Category	Number of items	Mean	Mean	p-value	Cohen's d
Concepts	8	3.26	3.15	0.26	0.13
Attitude Gain	3	3.20	3.03	0.17	0.19
Confidence Gain	6	3.15	2.99	0.18	0.18
Applications Gain	2	3.21	2.95	0.07	0.30
1-Not at all	*2-Just a little*	*3-Somewhat*	*4- A lot*	*5- A great deal*	

Survey on Interest in Computer Programming

Interest increases learning. Two types of interest have been identified by researchers, namely, individual interest and situational interest. Students tend to be more engaged if what they are learning is related to their individual interests. Situational interest is spontaneous, transitory, and environmentally activated (Krapp, Hidi & Renninger., 1992). Unlike individual interests that are developed over a long period of time, situational interest is temporary and triggered by external environment. Situational interest often precedes and facilitates the development of individual interest (Krapp et al., 1992). Moreover, situational interest is changeable and partially under the control of teachers.

According to the four-phase model of interest development (Hidi & Renninger, 2006), the first phase of interest development is a triggered situational interest. If sustained, this first phase evolves into the second phase, a maintained situational interest. The second phase may lead to the third phase characterized by an emerging individual interest and eventually the final phase, a well-developed individual interest.

A more detailed survey on situational interest in computer programming was conducted for the experimental group. A total of 53 students responded. The survey was conducted at the end of the last team-based game. In the survey, participants were asked questions on triggered situational interest, maintained situational interest feeling and maintained situational interest value based on a three-factor model developed by Linnenbrink-Garcia for academic domains (Linnenbrink-Garcia, et al., 2010). A 5 points likert-scale ranging from “Not at all”, “Just a little”, “Somewhat”, “A lot” to “A great deal” was used. Results of the survey are shown in Table 4.

Table 4. Results of Survey on Interest in Computer Programming

	Number of items	Mean
Triggered Situational Interest (SI)	4	3.06
Maintained Situational Interest (SI) Feeling	4	3.14
Maintained Situational Interest (SI) Value	3	3.30

Triggered-SI measures the level of grabbing students’ attention. Maintained-SI feeling measures whether the games are enjoyable and engaging. Maintained-SI value measures whether the games are viewed as important and valuable (Linnenbrink-Garcia, et al., 2010). The results showed that the gamification is able to achieve maintained situational interest beyond triggered situational interest. Maintained situational interest is important for developing individual interest.

CONCLUSION

This study explored the effect of gamification, an active learning method, in a computer programming module on students’ learning gains and interest in a tertiary institution. The findings suggest that gamified learning approach has positive effect on students’ academic performance and situational interest. Its effect on students’ assessment of learning gains is positive but of small effect size and lack of statistical significance when compared to traditional teaching approach.

This study is limited by several factors that are beyond the control of the author. First, the two groups of students under study may have different background such as previous education, experience and skills, motivation and interest. Secondly, the two groups were taught by different lecturers. Notwithstanding all the lecturers in this study are experienced in both programming and teaching, they have different personalities and different styles of teaching. This factor might be minimal as they were able to build good rapport with students and received very good feedback from the students. Finally, a single cycle is carried out in this study. More cycles of study would need to be conducted in future to verify the results.

Increasingly, educators have looked into gamification as a tool to improve students' learning outcomes. What kind of game principles to adopt and how to contextualize the games to meet the learning outcomes might be a challenge. There is no one-size-fit-all solution. Considerable amount of time is needed for planning, games design and creation, gamification execution, and regular rewards and leaderboard updates for the first run. However, once the games and leaderboard system are developed, re-using them for subsequent runs is easy and requires little effort.

REFERENCES

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining "Gamification". 5th International MindTreck Conference (pp. 9-15)

Domínguez , A., Saenz-de-Navarrete, J., de-Marcos, L., Fernández-Sanz, L., Pagés, C., & MartínezHerráiz, J. (2013). Gamifying learning experiences: Practical implications and outcomes. Computers & Education, 63, 380-392.

Freeman, Eddy, McDonough, Smith, Okoroafor, Jordt, Wenderoth. (2014). Active Learning increases student performance in science, engineering and mathematics. Proceedings of the National Academy of Sciences of the United States of America. vol. 111 no. 23

Fotaris, P., Mastoras, T., Leinfellner, R., & Rosunally, Y. (2016). Climbing Up the Leaderboard: An Empirical Study of Applying Gamification Techniques to a Computer Programming Class. The Electronic Journal of e-Learning, 14(2), 94-110. Retrieved from www.ejel.org

Glover, I. (2013). Play as you learn: gamification as a technique for motivating learners. Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (pp. 1999-2008). Chesapeake: AACE.

Hidi, S., & Renninger, K. A. (2006). The four-phase model of interest development. Educational Psychologist, 41, 111-127.

Huang, W., S. Dilip. (2013). A Practitioner's Guide To Gamification Of Education. PDF, Toronto: Rotman School of Management.

Kapp, K. M. (2012). The gamification of learning and instruction: Game-based methods and strategies for training and education. San Francisco: John Wiley & Sons.

Krapp, A., Hidi, S., and Renninger, K. A. (1992). Interest, learning and development. In Renninger, A., Hidi, S., and Krapp, A. (eds.), The Role of Interest in Learning and Development, Erlbaum, Hillsdale, NJ, pp. 3–25.

Leong, B., Koh, Z., & Razeen, A. (2011, September). Teaching Introductory Programming as an Online Game. Retrieved from Ben Leong, Associate Professor, School of Computing, National University of Singapore: https://www.comp.nus.edu.sg/~bleong/teaching/teaching-online-game.pdf

Lim, B., Hosack, B., & Vogt, p. (2012). A Framework for Measuring Student Learning Gains and Engagement in an Introductory Computing Course: A Preliminary Report of Findings. The Electronic Journal of e-Leanring, 10(4), 428-416. Retrieved from www.ejel.org

Linnenbrink-Garcia, L., Durik, A. M., Conley, A. M., Barron, K. E., Tauer, J. M., Karabenick, S. A., & Harackiewicz, J. M. (2010). Measuring situational interest in academic domains. Educational and Psychological Measurement, 70(4), 647-671.

Mekler, E. D., Brühlmann, F., Opwis, K., & Tuch, A. N. (2013). Do points, levels and leaderboards harm intrinsic motivation? : an empirical analysis of common gamification elements. First International Conference on Gameful Design, Research, and Applications (pp. 66-73). Toronto: ACM.

Piteira, M., & Costa, C. (2013). Learning computer programming: Study of difficulties in learning programming. Proceedings of the 2013 International Conference on Information Systems and Design of Communication (pp. 75-80). Lisboa: ACM.

BIOGRAPHICAL INFORMATION

Siew Peng Shorn, is the Programme Manager for Aerospace/Electrical/Electronics Programme and a Deputy Manager in the Embedded Technology Centre (Device Applications Section) at the School of Engineering, Nanyang Polytechnic. He is in charge of the overall management, curriculum review and development for the Aerospace/Electrical/Electronics Programme. He is also the Module Coordinator of Computer Programming module in the Aerospace/Electrical/Electronics Programme.

Corresponding author

Siew Peng Shorn
School of Engineering
Nanyang Polytechnic
180 Ang Mo Kio Avenue 8
Singapore 569830
65-6550-0585
siew_peng_shorn@nyp.edu.sg