Yang, J. C. et al. (Eds.) (2018). Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education

Identifying Meaningful Gamification-Based Elements Beneficial to Novice Programmers

Jenilyn AGAPITO[ab]* & Ma. Mercedes RODRIGO[a]
[a]*Ateneo de Manila University, Philippines*
[b]*Ateneo de Naga University, Philippines*
*jen.agapito@gmail.com

Abstract: Gamification is defined as the use of game design elements in non-game contexts to encourage certain behaviors. It is becoming a popular intervention used in computer science learning environments, including CS1. However, prior works have mostly implemented reward-based game elements which have resulted in varying behaviors among students. Meaningful gamification, described as the use of game design elements to encourage intrinsic motivation, is said to be a more student-centric approach. It is based on the concept of the Self-Determination Theory which states that there are three components associated with intrinsic motivation: mastery, autonomy, and relatedness. In this paper, we explore what gamification elements were beneficial to novice programmers. It looked into students' use of elements implemented into a system that allow them to take assessment activities typical of CS1. The elements are: feedback cycles, freedom to fail, and progress to support mastery; control to enable autonomy; and collaboration for relatedness. Control, freedom to fail, and feedback were the elements students used to their advantage since these allowed them to work towards improving their scores in the activities.

Keywords: gamification, novice programmers, CS1

1. Introduction

Gamification is defined as adding game-like elements and mechanics to a learning process (Deterding et al., 2011). It intends to provide users with a more gameful experience and to encourage certain desired behaviors (Deterding et al., 2013) by adding motivational affordances in an environment. In the academic setting, an area of interest for learning instructors and professionals is the search for more engaging designs of classroom instruction. Among the models which have been utilized in providing learning frameworks is that of games (Kapp, 2012). Hence, gamification has been evident in education and classroom design. Student retention and attrition has been a concern of computer science educators and researchers (Hoda & Andreae, 2014). The application of gamification in computer science instruction, particularly in programming courses, has been recognized as a prospective method that could lead towards a positive influence on learning.

1.1 Gamification in Education

Gamification has become evident in education and classroom design. Computer Science subjects, including CS1, have been the focus of several studies on gamification (Gibbons, 2013; Ibáñez et al., 2014; Neve et al., 2014; Pirker et al., 2014). They experiment with incorporating game design elements such as reward systems (Harrington, 2016; Sprint & Cook, 2015), leaderboards (Neve et al., 2014; Pirker et al., 2014;), point systems and leveling (Ibáñez et al., 2014), and microworlds (Neve et al., 2014) into aspects of the learning environment. Their results show how students express preference of a gamified strategy over the traditional because it is able to address their need for fun.

Learning environments are commonly gamified by using badges, levels/leaderboards, achievements, and points because they are relatively easy to implement. This is referred to as reward-based gamification (Nicholson, 2015). However, this technique has been shown to be effective in contexts that can supply the rewards continually. In the classroom, studies report on disparate responses from students. Some took the method enthusiastically and performed well; others felt demotivated and disengaged.

1.2 Meaningful Gamification

Nicholson defines gamification as the use of game design elements to help a user build intrinsic motivation to encourage engagement in a specific context (Nicholson, 2015). This is referred to as meaningful gamification. The theory behind is the Self-Determination Theory (SDT) (Deci & Ryan, 2002) which states that there are three components associated with intrinsic motivation: mastery, autonomy, and relatedness. Mastery is when one learns to the point of competence; autonomy means having a choice; and relatedness is about one's social engagement. This paper presents the results of an exploration on what gamification elements may be beneficial to students. It looked into how students used five game design elements implemented into an activity management system that allows them to take activities, such as quizzes and programming exercises, typical of CS1. The results describe which of the elements they have used to their advantage.

2. Research Objective

The main objective of the research was to experiment on the use of gamification in an introductory programming class. It sought to explore how the learning experience may be influenced by the presence of certain game design elements in the learning environment. This paper particularly focuses on the goal to determine which game design elements were beneficial to students when they accomplish assessment activities such as quizzes and programming exercises typically given in CS1.

3. Methods

This paper is part of a bigger study on the exploration of the impact of gamification on novice programmers' achievement and learning experience. A web-based platform that allows teachers to manage assessment activities typical of CS1 was developed. Detailed discussions regarding the system and how the game design elements were implemented are discussed in an earlier paper (Agapito & Rodrigo, 2017). A summary of the game design elements implemented is found in Table 1. They were purposely mapped to the components of SDT.

Table 1

Game Design Elements Implemented into the System

Component	Element	Description
Autonomy	Control	Students can choose from a set of questions to answer
Mastery	Feedback Cycles	Scores/ marks per item may be displayed
	Freedom to Fail	Students can re-attempt an activity.
	Progress	Progress for various skills represented as a radar chart.
Relatedness	Collaboration	Facility to award tutor points to other students.

This section presents details regarding the user testing conducted with CS1 students. Two (2) iterations of testing were conducted on different sets of students. The methods were the same for both iterations. Prior to the second round, the gamified system was modified based on how the first group interacted with the game design elements.

3.1 Participants

The first iteration was conducted on students enrolled in a CS1 class at a private university (School A) in the Philippines. They were using Python. Two sections (A and B) of students aged between 16 and 23 participated. Section A had 35 students while section B had 33. However, there were only 16 students from each class who completed all the exercises given to them during the experiment. The second round was done in a state university (School B), also in the Philippines. They were taught computing fundamentals using C++. Two sections (1A and 1B) of students aged 18–33 participated.

Section 1A had 35 students while section 1B had 25. Thirty-two (32) students from 1A completed all the exercises given during the experiment. However, only seven (7) from 1B were able to completely finish.

3.2 Testing Methods

Each of the two classes per iteration was randomly assigned as either the control (A – School A, 1A – School B) or experimental (B – School A, 1B – School B) group. The control group used a non-gamified version of the system, that is, the game design elements described previously were not available in their version. On the other hand, the experimental group had the features listed in Table 2. As mentioned, the features of the gamified version were modified based on the interactions in the first round. The number of extra items was increased to better provide data on the order in which the items are answered. When a student skips items, it is taken as an indication that he/she picked items he/she is comfortable answering rather than just going through the activity sequentially.

Table 2

Features of the System Used by the Experimental Group

Element	Iteration 1	Iteration 2
Control	extra question in quiz; extra problem in programming exercise	5 extra questions in quiz; 3 extra questions in programming exercise
Feedback Cycles	score displayed after submitting an activity; marks if items are correct/incorrect can be shown by clicking the “Show Item Feedback” button	score may be displayed by clicking on “Display Score”; marks if items are correct/incorrect can be shown by clicking “Show Item Feedback”
Freedom to Fail	unlimited attempts for an activity	3 attempts for an activity
Progress	profile which shows radar chart	profile which shows radar chart; a “Display Skill” option in the profile which makes their chosen skill visible to their classmates
Collaboration	facility to commend a classmate	facility to commend a classmate; a message emphasizing the feature

A “Display Score” button was added to track if students like to see their scores. The number of attempts was decreased from “unlimited” to three. This was based on the average number of attempts in the first round which was two (2). An extra allowable attempt was just added. As an added metric for progress, a “Display Skill” option was included in their profiles. It allows them to select one of the five skills tracked, that will be visible to their classmates. As for collaboration, feedback from the first experiment suggested the participants did not know there was such a feature. To make it more visible, a message that would emphasize it was added. The experiment went on for three sessions; each ran for two hours. Students were given 2 short quizzes, 2 programming exercises, and a long quiz.

3.2.1 Variables for Analysis

Control was characterized by students’ unequal question chains and question sequences. A question chain is a string representation of the sequence of items in an activity answered by a student. On the other hand, the question sequence represents the order in which the items were shown. If a question chain is different from the question sequence, it suggests the student’s tendency to answer items in the order he/she chooses to. Clicks on the “Show Item Feedback” and “Display Score” buttons allow us to capture their use of the feedback mechanisms. The number of times they re-attempt an activity depicts one’s exercise of freedom to fail. Progress is measured by the number of times students view their profiles. A metric added during the second round was the mechanism to select a skill they would want to “share” to their peers. The use of this feature and their profile views can be telling they value the information tracked by the radar chart. Finally, for collaboration, the number of tutor points awarded and received was considered to see whether the feature was utilized.

4. Results and Discussion

The focus of this paper is on the exploration of which of the five elements were beneficial to the students. To address this, students from the experimental groups were grouped into strong and weak students based on their long quiz scores. Students who got at least 60% were grouped as strong students; otherwise, they were grouped among the weak students. Then, their use of the different game design elements was examined. The results are discussed below.

4.1 School A - Experimental

All sixteen students from the experimental group who completed the exercises got passing scores in the long quiz. Since this was the case, we closely looked into the interactions with the game elements of those who got the lowest and highest scores. Student 89 had the lowest score – 15 out of 25–the passing mark. On the other hand, Student 114 was the only student who got a perfect score.

Students 89 and 114 both showed behaviors of choosing items to answer in their activities (control). This can be seen in their number of unequal question chains and sequences in their attempts. Looking into Student 89's question chains, we observed he would sequentially answer items up to the number required. Then, when he sees the last item and decides he would want to answer it, he'd clear out one those previously answered so he could submit an answer for the last one. Student 114, on the other hand, skips not just the last items in an activity, but also those in between. He also has the inclination to go back to a previously skipped item. Nonetheless, it can be seen from their data how they used their abilities to choose items to submit answers for.

Both students clicked on the "Show Item Feedback" button in 4 out of the 6 activities. Student 114 did not click the button for SQ2 and LQpt2–activities in which he got perfect scores. Since he already got all items correctly, this form of feedback was not anymore necessary.

As for freedom to fail, Student 89 did not re-attempt any of the activities, regardless of getting a zero in the programming activities. By contrast, Student 114 seemed to re-attempt until he got a perfect score. This shows how the re-attempt mechanism helped him improve his scores.

Thirteen (13) students viewed their profiles at least once. Both Students 89 and 114 had logged two profile views – the average number recorded for this group. The commend facility was the least utilized. Among the 16 students, only 2 used it and it seemed as though they "exchanged" points because they awarded respective points to one another. Students 89 and 114 did not use this feature.

4.2 School B - Experimental

Unlike School A, only 2 of the seven 7 students who completed the activities in School B got passing marks in the long quiz. In the succeeding discussions, the students who passed will be referred to as the strong students; otherwise, the weak students.

To further substantiate the skipping behaviors exhibited by School A, the number of extra items in the second experiment was increased to provide students more choices. The high percentages of attempts of the strong and weak students with unequal question chains suggests they exercised their ability to choose items if it were permitted.

In this experiment, the score was intentionally hidden. But, a "Display Score" button was available if they want to know how they fared. The "Show Item Feedback" was still in place in their version. All 7 students clicked "Display Score" for all activities signifying that the score is significant to them. The strong students did not click on "Show Item Feedback" for the programming activities. Two of the weak students did not display per item marks for all activities.

Both the strong and weak students utilized the re-attempt feature of the system. They appear to re-attempt to get passing marks or to get better scores. However, two scenarios may be observed regarding the weak students' attempts and scores: (1) scores did not change; and (2) the scores fluctuate. Some students re-attempted the activities but ended up not getting better or worse scores.

Others had attempts with lower scores than their previous takes. The way the strong and weak students used the re-attempt facility of the system shows it works to both their advantage. Consistent with School A's data, it allows strong students to improve their performance. At the same time, weaker students who are not able to receive favorable scores the first round are provided with opportunities to recover.

As for progress, all 7 students from School B viewed their profiles at least once. The strong students viewed their profiles 12 times–the maximum number of views recorded. This may indicate that the students found the information depicted in their radar charts and summaries to be helpful. The radar chart represented what areas they are good at and those that they can improve in terms of the "skills" defined in the system. Self-monitoring as a way to better understand one's strengths and weaknesses has been shown as a characteristic of high-performing students (Wagster et al., 2007). An additional metric for progress was a feature to display one's skill so it is viewable by their classmates. Three students utilized this feature–one of them was one of the strong students. Allowing them to "share" their chosen skill, ideally their strongest, may have further encouraged him to view his profile.

The commend system turned out to be the least used. O the 7 students, 4 awarded commends to their peers. One of the strong students awarded all his available commends and received one.

4.3 Summary

The preceding sections discuss how the strong and weak students used the game design elements to help us gain insights which are beneficial to them. The results imply how freedom to fail coupled with feedback worked to the advantage of the students. Allowing students to re-attempt, immediately giving scores, and allowing them to see which items they correctly/incorrectly answered encouraged them to improve their scores. This was especially favorable to students who do not pass the first time. Control, through providing them with extra items to choose from, was utilized by the students. In both experiments, the number of unequal question chains and sequences showed students' tendency to decide which items to answer or the order in which to answer them. Progress and collaboration turned out to be the least used. However, strong students from School B had the maximum number of profile views among all students. This leads us to presume they were monitoring their "skills" and such behavior may have helped them in performing better. Unfortunately, collaboration through peer mentoring was a feature not very visible to the students. Even when it was highlighted, the number of students who used it in the second experiment could not be deemed significant.

5. Conclusion and Future Works

Majority of prior works in gamification implemented reward-based techniques that encourage extrinsic motivation. Scott Nicholson introduced meaningful gamification, the use of game design elements to encourage desired behaviors by tapping into students' internal motivation. It leverages on the Self-Determination Theory which states that there are three components associated with intrinsic motivation–autonomy, mastery, and relatedness.

This paper focused on the exploration of the effects of meaningful gamification on novice programmers' learning experience. It sought to identify which of the elements implemented were beneficial. Generally, students used control, feedback, and freedom to fail to their advantage. They answered items non-sequentially. This implies they make decisions relative to items they answer or the order in which to answer them. The score was found to be vital as evidenced by all students from School B clicking on the "Display Score" button for all activities. The per item marks kept them apprised of mistakes they can correct in next attempts. The re-attempt mechanism gave students the chance to fail and recover. Strong students re-attempted to get perfect scores; the weaker ones aim for the passing. Regardless of the goal, it gave them the opportunity to improve. The radar chart appeared to be significant to School B's strong students who had the most number of views recorded. They may have regarded it as a way of self-monitoring to aid them in assessing areas they can improve.

These results are promising. However, due to the sample size and duration of the experiments, these may not be deemed generalizable. Nonetheless, the study contributes observations of behaviors that may be valuable to consider in continuing gamification research in the educational setting. In terms of moving the project forward, further testing with more students in a longer period of deployment may be valuable in collecting more data. More data collected may better provide behavioral patterns that can help validate the results of the research. Further analyses such as looking into time students spend on an attempt and type of questions frequently skipped may provide other characterizations not covered in this study.

Acknowledgements

We would like to thank the student participants who signified their participation in the study. We also like to thank the CS1 instructors for permitting us to collect data in their schools. Much gratitude to the Department of Science and Technology–Engineering Research and Development for Technology (DOST-ERDT) for the scholarship awarded to the student researcher. Lastly, we thank the Ateneo Laboratory for the Learning Sciences (ALLS), Ateneo de Manila University for supporting this study.

References

Agapito, J. & Rodrigo, M. M. (2017). Designing an Intervention for Novice Programmers Based on Meaningful Gamificaiton: an Expert Evaluation. In *Proceedings of the 25th International Conference on Computers in Education*. New Zealand: Asia Pacific Society for Computers in Education.

Deci, E. L., & Ryan, R. M. (2002). Handbook of self-determination research. University Rochester Press.

Deterding, S., Björk, S. L., Nacke, L. E., Dixon, D., & Lawley, E. (2013, April). Designing gamification: creating gameful and playful experiences. *In CHI'13 Extended Abstracts on Human Factors in Computing Systems* (pp. 3263–3266). ACM.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011, September). From game design elements to gamefulness: defining gamification. *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments* (pp. 9–15). ACM.

Gibbons, T. E. (2013, October). COR: A new course framework based on elements of game design. In *Proceedings of the 14th annual ACM SIGITE conference on Information technology education* (pp. 77–82). ACM.

Harrington, B. (2016, May). TrAcademic: Experiences With Gamified Practical Sessions for a CS1 Course. In *Proceedings of the 21st Western Canadian Conference on Computing Education* (p. 25). ACM.

Hoda, R., & Andreae, P. (2014, January). It's not them, it's us! Why computer science fails to impress many first years. *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148* (pp. 159–162). Australian Computer Society, Inc.

Ibáñez, M. B., Di-Serio, A., & Delgado-Kloos, C. (2014). Gamification for engaging computer science students in learning activities: A case study. *IEEE Transactions on Learning Technologies*, 7(3), 291–301.

Kapp, K. M. (2012). Games, gamification, and the quest for learner engagement. *T+ D*, 66(6), 64–68.

Neve, P., Livingstone, D., Hunter, G., Edwards, N., & Alsop, G. (2014). More than just a game: Improving students' experience of learning programming through gamification. A paper from the *STEM Annual Conference* 2014.

Nicholson, S. (2015). A recipe for meaningful gamification. In *Gamification in education and business* (pp. 1–20). Springer International Publishing.

Pirker, J., Riffnaller-Schiefer, M., & Gütl, C. (2014, June). Motivational active learning: engaging university students in computer science education. In *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 297–302). ACM.

Sprint, G., & Cook, D. (2015, March). Enhancing the CS1 student experience with gamification. In *Integrated STEM Education Conference (ISEC), 2015 IEEE* (pp. 94–99). IEEE.

Wagster, J., Tan, J., Wu, Y., Biwas, G., & Schwartz, D. (2007, January). Do learning by teaching environments with metacognitive support help students develop better learning behaviors? In *Proceedings of the Annual Meeting of the Cognitive Science Society*. 29(29).