ISATE 2013
International Symposium on Advances in Technology Education
25 – 27 September 2013, Nara, JAPAN

Improving computer programming education using Gamification

Tomomichi Nishino[*,a] and Yoshikazu Kobayashi[b]

[a] Department of Applied Chemistry, Akita National College of Technology, Akita City, Japan
[b] Department of Mechanical Engineering, Akita National College of Technology, Akita City, Japan

* nishino@ipc.akita-nct.ac.jp

Abstract

This report clarifies changes in attitudes and educational effects from using the gamification method for computer programming class instruction. Gamification is the process of using game thinking and game mechanics to raise learner's motivation and to stimulate learner's interest. Examples of gamification in programming education are awarding points to students according to their level of achievement and using leader boards. Survey results reveal that student motivation is increased. Improvement activities using gamification methods and their associated effects and problems are examined.

Keywords: *Computer programming education, Gamification, Educational improvement, Survey results, Students' motivation*

Introduction

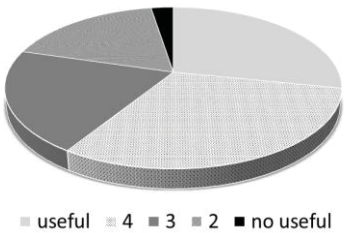
Demand for programming ability is increasing in various fields. A certain degree of ability is necessary for engineers who do not aim to be computer programmer. Generally, it is said that interest in programming is necessary for programming acquisition. Therefore, interest in programming is important for initiative in instruction. To date, some report in the literature have described exercise method using small step tests in the early stage of programming education (S.Yoshida, *et al.* 2012). However, no effective practice method has been established for an introduction to programming education.

Instruction in computer programming has been introduced for first-year students of Applied Chemistry. However, some students are uninterested in computer programming because they believe that computer programming is unrelated with their specialization. Others have the notion that they are not good at programming. A marked difference exists between high and low motivation to learn. The authors have strived to teach computer programming to those students who have little or no motivation. From last year, to improve students' motivation, the authors have sought to teach computer programming using gamification. This report describes the introduction of gamification to classes and some educational effects of using gamification method in computer programming education.

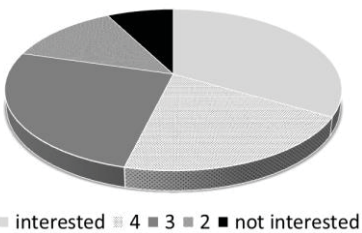
Conventional lecture style

Computer programming courses are mandatory for first year students in the Department of Applied Chemistry at the Akita National College of Technology. Computer programming lectures, which are 100 min long, are held once a week for about 15 times during one semester. In the class, students receive instruction in basic knowledge of C programming such as variables, arithmetic, control statements, looping constructs, arrays, and file input-output in the first half. Subsequently, in the second half, students do programming practice individually using a PC. Finally, the teacher presents a model answer and checks it. The teacher gives a small test related to practice contents for promoting a review of past lessons, and then determine the degree of comprehension. The programming grade is calculated from periodical examination results (80%) and small test results (20%).

The authors conducted an opinion survey of students about computer programming. The survey results shown in question 1 (Do you think that the lesson of



Q.1 Do you think that the lesson of programming is useful in the future ?



Q.2 Is it interested in the lesson of programming ?

Figure 1 Surveys results about programming.

programming is useful in the future?) of Figure 1 reveal that nearly 60% of students recognized that programming is "useful" or "slightly useful". Furthermore nearly 50% of the students felt "interest" or "slight interest" in programming. For question 2 (Is it interested in the lesson of programming?), nearly 20% of students reported "little interest" or "no interest".

Gamification

In 2011, Deterding *et al.* (2011) defined gamification as the use of game design techniques, games thinking, and game mechanics to enhance non-game contexts. Reportedly, important effects that compel a user to wants to use a service can be expected by adding a game feeling to a learning task. Although the concept has been explored primarily in marketing areas, the potential for its application has been extended to other fields such as health, environment, government, and education (Lee and Hammer, 2011). Especially for education, gamification offers great potential to motivate students.

Gamification is typically achieved using achievement points, with leader boards for the visualization of states, growth with competition and/or cooperation, and so on. A gamification loop is portrayed in Figure 2.

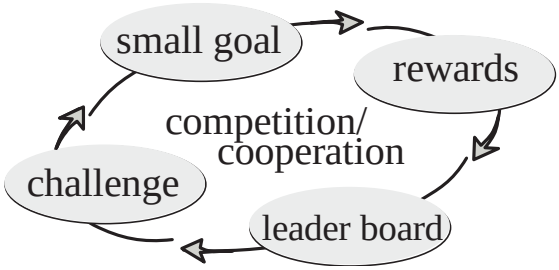
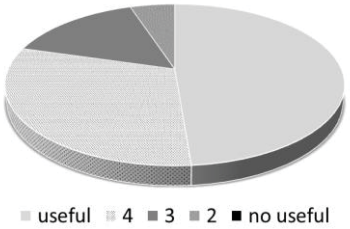


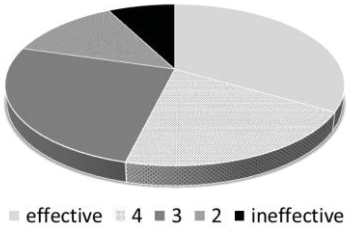
Figure 2 Gamification loop.

Introduction of proficiency tests to programming classes

The programming proficiency test was first introduced to measure students' knowledge and to stimulate the students' motivation. A proficiency test administered on the last week of programming assessed the first half of the class. The authors scored the tests and returned them. Principal differences between the proficiency test and a conventional small test included the formulation of pass/fail criteria. Passing grades accounted for 60% of the grades given for this test. Students who passed were granted a rating at that time. However, it is necessary for student with unacceptable performance to take a re-examination before the next class. Because the pass line was low, many students felt little concern about it. Moreover, some students seemed to attend class only rarely without understanding it. Questionnaires were filled out anonymously at the end of the course. Survey results revealed that nearly 80% of the students recognized that the programming proficiency test was "useful" or "slightly useful", as shown in responses to question 3 (Do you think that the



Q.3 Do you think that the programming proficiency test is useful for review or understanding?



Q.4 Do you think that the system of proficiency test is effective for programming ?

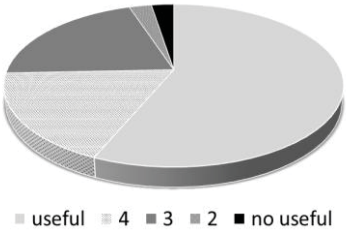
Figure 3 Surveys result about proficiency test.

programming proficiency is useful for review or understanding?) in Figure 3. Furthermore, survey results related to the borderline between pass and failure in the test show that nearly 55% of the students felt that it was effectual. However, responses to question 4 (Do you think that the system of proficiency test is effective for programming?) show that nearly 20% of the students felt it to be ineffectual. Proficiency tests were insufficient to make students more interested in programming, but they served as a study incentive for low-scoring students.

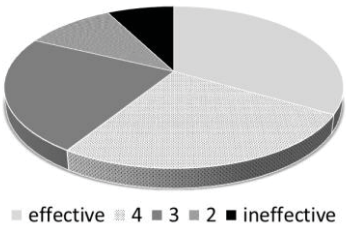
Introduction of gamification to programming classes

After instruction, the students performed programming practice individually using the PC. Figure 4 shows the climate of the classes. Students solved the given subjects and sent a file to the teacher by e-mail when the programming task was completed. The authors checked whether the programming file worked. Then they awarded points to students according to the order in which a number of files arrived. Subsequently the points obtained through programming tasks were

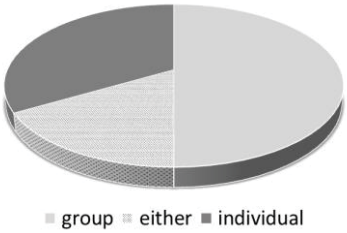
Figure 4 the students performed programming practice individually using the PC.



Q.5 Do you think that this programming practice is useful?



Q.6 Do you think that the point system and point visualization system are effective for learning programming?



Q.7 Which do you think is more effective for you to make points visible to others?

Figure 5 Survey results about the seminar course.

opened to all class members. By exposing the point totals to students, it became possible to check on the students themselves. However, we do not use points for students' records at this time. Those approaches led to a high number of students showing good effects, but no effect was visible for students with poor achievement. Therefore, we formulated a mechanism to introduce a team system for class cooperation. Class members were divided into six groups. The numbers of points were released for each group. Then, cooperative action in each group became apparent. Especially, actions were observed by which well-performing students taught poorly performing students and pointed out their errors.

Survey results are presented in Figure 5. Results for question 5 (Do you think that this programming practice is useful?) shown in Figure 5 reveal that nearly 75% of the students recognized that programming practice is "useful" or "slightly useful". Survey results revealing opinions about awarding points and leader boards in the practice show that nearly 60% of the students felt it to be effectual. However, nearly 20% of the students felt it to be ineffectual, as shown in responses to question 6 (Do you think that the point system and point visualization system are effective for learning programming?). Apparently, it is more effective to open scores to everyone: 46% of the students said "group" and 30% of students said "individual" in question 7 (Which do you think is more effective for you to make points visible to others?). Students were concentrating on programming within a time limit. Furthermore, students consulted one another more than they ever had before.

Conclusions

The authors introduced a system of gamification into computer programming education, with challenges related to programming exercises, small goals that were subdivided and evaluated, a reward system that assigned points, and a leader board showing totals and points for each group. From introduction of the system, we demonstrated that mutual communication was encouraged. In this trial stage, changes were observed in student motivation. Gamification is regarded as offering additional possibilities for the improvement of learning outcomes.

References

Deterding, S., Dan Dixon, Rilla Khaled, and Lennart Nacke (2011). From Game Design Elements to Gamefulness: Defining "Gamification". Proceedings of MindTrek. ACM.

Lee, J. J. and J. Hammer (2011). Gamification in Education: What, How, Why Bother? Academic Exchange Quarterly, 15(2). Retrieved from http://www.gamifyingeducation.org/files/Lee-Hammer-AEQ-2011.pdf

S.Yoshida, *et al.* (2012). Exercise Method Using Small Step Test for Introduction to Computer Programming Education (in Japanese). *J of JSEE*, 60(6), 170-175.