Code It! A Gamified Learning Environment for Iterative Programming

Jeffrey GARCIA, Jo Rupert COPIACO, John Philips NUFABLE, Francis AMORANTO and Judith AZCARRAGA*
De La Salle University, Philippines
judith.azcarraga@dlsu.edu.ph*

Abstract: Learning computer programming concepts has always been a challenge to most students. An innovative learning environment for teaching iterative programming concepts is presented here. The system includes gamification elements such as point system, leaderboards and achievements. Thirty (30) students were asked to use both non-gamified and gamified learning systems. The level of engagement of the students with the system and improvement on their programming performance were observed. Results show that students are more engaged and have more preference over the gamified version than the non-gamified one. Moreover, 57% of the users have improved programming scores after using both systems. Whether students are able to retain better the programming concepts have not yet been assessed.

Keywords: Gamification, Learning Environment, Iterative Programming

1. Introduction

Gamification has been used in recent years to invoke a certain behavior among users by applying game elements into non-game context (Deterding et al., 2011). Many gamification systems such as Lumosity (http://www.lumosity.com), and Codecademy (http://www.codecademy.com) had been created for educational use with a certain amount of success (Loewen, 2012). To that end, gamification has been applied to numerous learning contexts such as programming but its merits and disadvantages have not been properly researched.

In this study, a gamified learning environment system for iterative programming called Code It! was developed and tested among its target users. The goal was not only to build a system that would help students in learning iterative programming concepts but also to evaluate how effective and attractive to students are the gamified system would be compared to a non-gamified one. This study also aims to compare student perception on their performance with their actual performance.

The system was tested among thirty (30) undergraduate students as they answer several iterative programming questions in varying difficulty levels. Two versions of the system were deployed - gamified and non-gamified versions. With the gamified system, gamification elements such as points, personal leaderboards and achievements were included.

2. Gamified Learning Environment Systems

Games are defined as physical or mental activity designed for amusement and/or pleasure. Learning systems with game features have yielded some positive results in terms of engagement and comprehension of their users.

Systems with gamification elements include leaderboards, assessment reports in the form of progress bars or graphs, points or *experience* upon the completion of a task, achievements or *badges* upon the completion of a requirement, a means to use gathered points for added customizability and a means of publicizing the current progress to others.

For the *leaderboards* element, the progress of a user as well as information on who has used the system the most, are shown. Another gaming element is the display of *points* which is a numerical value that represents how much progress is made upon answering a question correctly. The total points obtained would then be shown through a progress bar that would give a visual representation of how far the student has gone over a period of time they have been using the system. These points give meaning to an action undertaken and that each progress done, can be rated based on how much is obtained upon accomplishing the task correctly.

Such elements can be seen in the gamified and collaborative works of Coursera (https://www.coursera.org), Sokikom of the Sokikom Research Study with Scales Technology Academy, Kaplan Gamification System (Shane, 2013), Lumosity (http://www.lumosity.com) and Duolingo (Vesselinov and Grego, 2012).

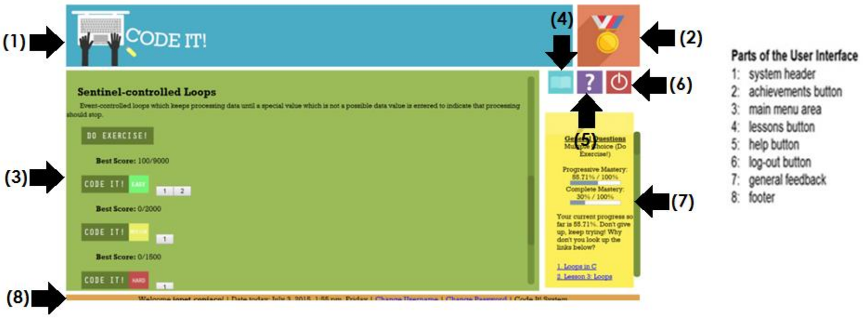
Coursera provides a wide variety of courses with video lectures, exams and exercises. Coursera provides feedback and progress can be measured based on the assignments and quizzes the user partakes in. Sokikom, a mass multiplayer online math game, combines the capability of online games to have users communicate with one another and explore a virtual world together. The Kaplan Gamification System was used in online classrooms in Kaplan University. Recent achievements or badges, a progress bar, to see how many points the user needs to advance to the next level and additional learning resources are visible to the users (Shane, 2013). Lumosity, also known as Lumos Labs, uses gamification in the form of brain training activities which would require speed, memory, attention, flexibility, problem solving or a combination of them. Duolingo helps students learn different languages such as Latin American Spanish, French, German, Portuguese, Italian, Dutch, etc. Users gain skill points or *coins* as they learn a particular language, such as when they complete a lesson. (Vesselinov and Grego, 2012).

Over time, a variety of attractive features and new twists to the application of game mechanics had been applied. Rewards such as badges and points in Kaplan and Duolingo systems do not only motivate users in maintaining their attention span but could also depict the current grasp of the user on a particular subject and their ease or difficulty they experience when advancing to new set of exercises. Depending on the subject or task being gamified along with the target audience, there exists numerous ways that would make learning and solving problems tasks engaging and entertaining.

3. Code It! Gamified Learning Environment System

The Code It! system is a learning environment with gamification elements for teaching iterative concepts of the C programming language. This was developed as a support tool for in-class learning. The system is web-based which can run in most web browsers and even in mobile devices. Tools and languages such as PHP, JavaScript, HTML, CSS XAMPP (Apache server, PHP and MySQL) and Codeblocks were used in developing the entire system.

The system focuses only on the PBLs (Points, Badges, Leaderboards) game elements in the form of point system, leaderboards and achievements as shown Figure 1.



Sentinel-controlled Loops

Event-controlled loops which keeps processing data until a special value which is not a possible data value is entered to indicate that processing should stop.

Best Score: 100/9000

Best Score: 0/2000

Best Score: 0/1500

General Questions
Multiple Choice (Do Exercise!)

Progressive Mastery: 55.71% / 100%

Complete Mastery: 30% / 100%

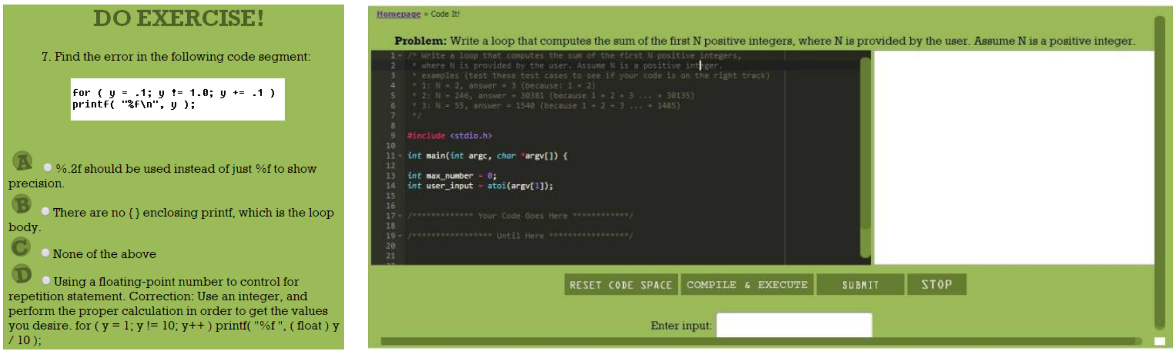
Your current progress so far is 55.71%. Don't give up, keep trying! Why don't you look up the links below?

1. Loops in C
2. Lesson 3: Loops

Welcome [illegible]! | Date today: July 3, 2015, 1:55 pm, Friday | Change Username | Change Password | Code It! System

Figure 1. Code It! system user interface

The system includes topics such as count-controlled, sentinel-controlled and flag-controlled loops. Questions are in the form of multiple choice and programming type as shown in Figure 2.



DO EXERCISE!

7. Find the error in the following code segment:

```
for ( y = .1; y != 1.0; y += .1 )
printf( "%f\n", y );
```

A ○ %.2f should be used instead of just %f to show precision.

B ○ There are no { } enclosing printf, which is the loop body.

C ○ None of the above

D ○ Using a floating-point number to control for repetition statement. Correction: Use an integer, and perform the proper calculation in order to get the values you desire. for (y = 1; y != 10; y++) printf("%f ", (float) y / 10);

Homepage » Code It!

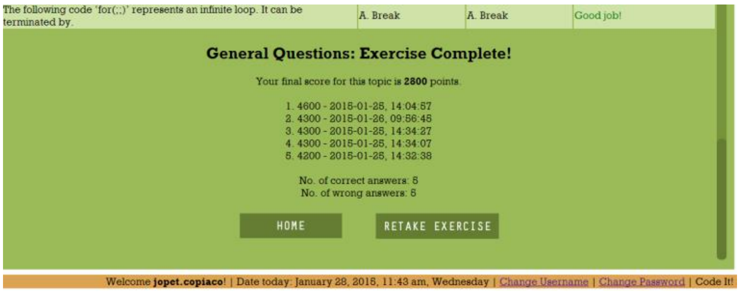
Problem: Write a loop that computes the sum of the first N positive integers, where N is provided by the user. Assume N is a positive integer.

```
/* Write a loop that computes the sum of the first N positive integers,
 * where N is provided by the user. Assume N is a positive integer.
 * examples (test these test cases to see if your code is on the right track)
 * 1: N = 2, answer = 3 (because: 1 + 2)
 * 2: N = 246, answer = 30381 (because 1 + 2 + 3 ... + 30135)
 * 3: N = 55, answer = 1540 (because 1 + 2 + 3 ... + 1485)
 */

#include <stdio.h>

int main(int argc, char *argv[]) {

int max_number = 0;
int user_input = atoi(argv[1]);


/************* Your Code Goes Here ************/

/***************** Until Here *****************/


```

Figure 2. Sample multiple choice question and programming problem

Points are given out upon correctly answering an exercise. The amount of points given for a multiple choice type of question varies according to how many times it has been answered correctly in a consecutive fashion. For instance, should the given points for the first take of a question be 100 points, answering the same question again would gain 200 points and up to a maximum of 300 points. On the other hand, if the user fails to answer the question once, the points given for that question will be reset to 0 and will be reflected in the personal leader board as well as in the progress bar which depicts the current level of mastery of the user. Moreover, for programming-type questions, a fixed amount of points is given upon completion unlike in the multiple choice type questions.

The system, apart from measuring the progress through points, also provides *personal leaderboard* gamification element. Such element only presents the performance of a particular user but not of other users. Figure 3 shows a sample personal leaderboard in which the performance of the user for the past five takes of a particular exercise are presented.



The following code 'for(;;)' represents an infinite loop. It can be terminated by.	A. Break	A. Break	Good job!

General Questions: Exercise Complete!

Your final score for this topic is **2800** points.

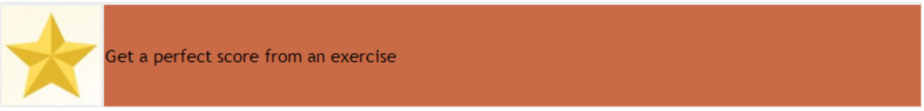
1. 4600 - 2015-01-25, 14:04:57
2. 4300 - 2015-01-26, 09:56:45
3. 4300 - 2015-01-25, 14:34:27
4. 4300 - 2015-01-25, 14:34:07
5. 4200 - 2015-01-25, 14:32:38

No. of correct answers: 5
No. of wrong answers: 5

Welcome **jopet.copiaco**! | Date today: January 28, 2015, 11:43 am, Wednesday | Change Username | Change Password | Code It!

Figure 3. Sample personal leaderboard

The other game element that is adapted in the system is the *achievements* as shown in Figure 4. Achievements are defined by the system administrator. All achievements are visible to the users.

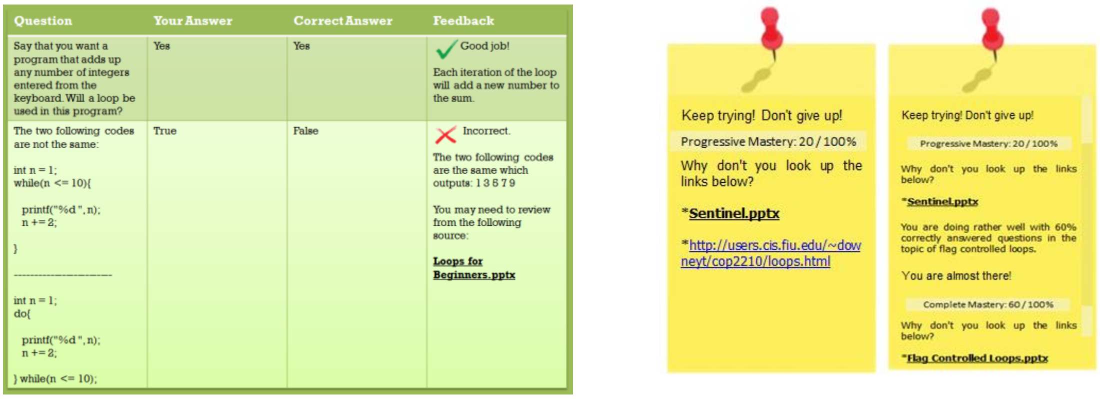


Get a perfect score from an exercise

Figure 4. Sample achievement

Apart from the exercises within the system, a user may also access relevant sources of information or references to a particular topic. Such may be in the form of slides or documents that are uploaded by a professor, or online links to other useful resources. These resources are shown through a general feedback. Such resources are offered specifically to students who are not performing well with the topic for further review.

Upon completion of an exercise, a specific feedback will be given to the user for every question that was answered, as shown in Figure 5a. On the other hand, a general feedback is generated based on student performance on the exercises. This can also provide some resources in the system that students can use to review certain topics. Such are available if the student has not achieved complete mastery of the topic. A sample general feedback is shown in Figure 5b.



Question	Your Answer	Correct Answer	Feedback
Say that you want a program that adds up any number of integers entered from the keyboard. Will a loop be used in this program?	Yes	Yes	✓ Good job! Each iteration of the loop will add a new number to the sum.

The two following codes are not the same:
int n = 1;
while(n <= 10){
printf("%d ", n);
n += 2;
}
---------------------- 	
int n = 1;
do{
printf("%d ", n);
n += 2;
} while(n <= 10);
	True	False	✗ Incorrect.
			The two following codes are the same which outputs: 1 3 5 7 9
			You may need to review from the following source:
			Loops for Beginners.pptx

Keep trying! Don't give up!

Progressive Mastery: 20 / 100%

Why don't you look up the links below?

***Sentinel.pptx**

*http://users.cis.fiu.edu/~downeyt/cop2210/loops.html

Keep trying! Don't give up!

Progressive Mastery: 20 / 100%

Why don't you look up the links below?

***Sentinel.pptx**

You are doing rather well with 60% correctly answered questions in the topic of flag controlled loops.

You are almost there!

Complete Mastery: 60 / 100%

Why don't you look up the links below?

***Flag Controlled Loops.pptx**

a) b)

Figure 5. Sample a) specific and b) general feedback

4. Assessing the Effectiveness of Code It!

The Code It! system with and without gamification elements was tested among thirty (30) undergraduate students for a period of about six (6) weeks. All the participants are familiar with the C programming language. Some of them are currently taking a C programming course while the others have taken the course for some time.

The participants were given pre- and post-tests. The purpose of the tests is to gauge their current programming skill particularly on iterative programming concepts. Both tests include 15 multiple choice and 2 programming questions. The pre-test is given prior to the use of the system. This includes 5 questions on three sub-topics with varying difficulty levels. After the test, the user was asked to answer an entry survey questions. These are used to profile the participants and to assess their perception on programming in general and their experience in using the programming language.

The users were divided into two (2) equal groups (i.e. 15 students per group). For Group 1, the gamified version of the system was used until the half way point of the deployment period (a maximum of three weeks), or if the user has answered all questions at least once. After which, the non-gamified system was used for the same period as the gamified. The users from the other group, Group 2, used the non-gamified version first followed by the use of the gamified version, with the same period as the first group. The type and difficulty level of the questions for both groups are identical. The set of questions is generated only once and is not adjusted according to the student score.

Upon using both systems, a user is given a post-test and an exit survey. The exit survey is composed of questions that ask about the perception of the user regarding their usage of the system.

5. Results and Discussions

To determine the effectiveness of the Code It! system and the effect of various gamification elements in teaching iterative programming, the system was deployed to its target users. The pre- and post-tests were very useful in the assessment of student performance after using the system. Figure 6 presents the scores of the students before and after using the system. Also shown are the consolidated results of all 30 users who used both gamified and non-gamified versions of the system. Out of the 30 users, 17 of them had shown some improvement in their overall scores (a post-test result greater than their pre-test result). Moreover, 3 users had obtained identical results (a post-test result equal to the pre-test result). All the remaining students had a score lower than their pre-test result (see Table 1) in which 2 of them are currently taking a programming course while the rest are either not familiar with iterative concepts or familiar but have taken a programming course for several months ago.

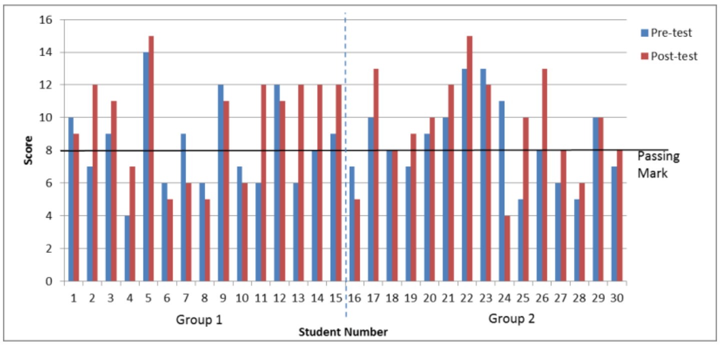


Figure 6. Pre-test and Post-test results.

Table 1 presents a summary of student performance in programming exercises. Students who had answered programming exercises are most likely to have better performance than those who did not. Programming exercises have often been effective in enhancing programming skills since they involve application and understanding of concepts they had learned from various sources of information. Moreover, there is an overall improvement in performance based on the mean difference of post-test and pre-test scores for both groups. Mean difference for Group 1is 1.40 while Group 2 is 0.93. Overall mean difference among all the students is 1.17.

Information such as the number of times the system was used, the number of actions per day and the type of questions answered were all captured. Such actions are the log-in time, downloading of a resource, accessing a multiple choice question exercise, and accessing a programming type question exercise.

Table 1: Programming Exercise Usage and Student Performance

Performance from Pre-test to Post-test	**Improvement of Score**	**No improvement**	**Equal Scores**	**Total**
No. of Students who answered programming exercises	13	3	2	18
No. of Students who did not bother to answer programming exercises	4	7	1	12

Figure 7 shows that more student actions were recorded in the gamified system than the non-gamified one. It was consistent for both groups that the gamified system was used more often than the other. All students in Group 1 have used the gamified more than the non-gamified while students in Group 2 have exhibited the same behavior except for 3 students. Based on the results, students are more inclined to use the gamified system more often than the non-gamified one.

On another note, out of the thirty (30) users, twenty-seven (27) or 90% of them believed that the use of gamified system has helped them understand the iterative concept regardless of their scores in the post-test.

6. Conclusion and Future Study

Learning environments that include gamification elements have shown to improve the scores of their users. One of the factors that may have contributed to the improvement in performance is the increased engagement of the users with a gamified system. This study confirms that, indeed, the scores of students have improved and their engagement with the learning environment has increased when a gamified system is used.

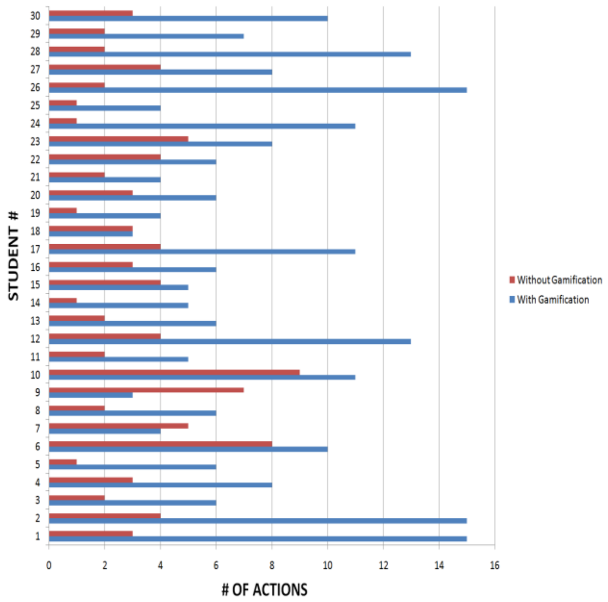


Figure 7. Number of actions performed by students on both gamified and non-gamified systems

Thirty (30) undergraduate students have participated in the testing of a learning environment for teaching iterative concepts of the C programming language, called Code It!. Two versions of the system were tested among the students: gamified and non-gamified. For the gamified version, three gamification elements such as points, leaderboards and achievements were included in the system, whereas for the non-gamified version, such gamification elements were removed from the system.

Each student was able to use both versions. Pre- and post-tests were given to students for the assessment of their knowledge in iterative programming before and after using the two systems, respectively. Survey questions were also provided for the evaluation of the systems. Other useful information such as the number of times an exercise was taken and the number of resources that were used or downloaded were recorded.

Results have shown that most students preferred to use the gamified system more than the non-gamified version. Students tend to be more engaged with the former rather than the latter, i.e. more activities were logged in the gamified than the non-gamified system. Moreover, students who often used the gamified system had shown some improvement in their programming scores, based on their pre-test and post-test scores. Even students who have no improvement in scores had claimed that the system, indeed, had helped them in learning the iterative programming concept. This would suggest that students had a more positive view of their results with the use of the system.

Gaming elements such as points, achievements and personal leaderboards have shown to engage students and tend to develop some positive effects on student learning perception. Other gaming elements may be explored in the design of future learning environment systems for teaching basic programming concepts.

References

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: defining "gamification". From game design elements to gamefulness. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. (pp.9-15). ACM.

Ibanez, M., Di-Serio, A., & Delgado-Kloos, C. (2014, June 6). IEEE Xplore Abstract - Gamification for Engaging Computer Science Students in Learning Activities: A Case Study.

Loewen, S. (2012) Does Lumosity Work? Retrieved August 30, 2014, from http://www.healthguidance.org/entry/16057/1/Does-Lumosity-Work.html

Shane, K. (2013). Kaplan’s Gamification System Shows 155% More Student Engagement | Gamification Co. Retrieved February 2, 2014, from http://www.gamification.co/2013/07/18/kaplan-gamification-system-shows-more-studentsengagement/

Vesselinov, R. & Grego, J. (2012). Duolingo Effectiveness Study. Retrieved February 11, 2015 from http://static.duolingo.com/s3/DuolingoReport_Final.pdf