

Increasing student motivation in computer programming with gamification

José Figueiredo
Research Unit for Inland Development
Polytechnic of Guarda
Portugal
jfig@ipg.pt

Francisco José García-Peñalvo
Computer Science Department Research Institute for
Educational Sciences GRIAL research group
University of Salamanca
Spain
fgarcia@usal.es

Abstract—Games have important motivational power. They take advantage of a set of tools to encourage people to engage with them just for the joy of playing and the possibility to win. While gamification is gaining ground in a lot of areas in our society, its application in education is still an emerging trend. In recent years, gamification has attracted the attention of researchers from different areas such as teaching and learning computer programming. Ever since the first programming languages emerged, the problems inherent to programming teaching and learning have been studied and investigated. The theme is very serious, not only for the important concepts underlying computer science courses but also for reducing the lack of motivation, failure, and abandonment that result from student frustration. In most of these studies and research one factor prevails, lack of student motivation or how to motivate students to learn programming. One way to combat this problem is to use gamification. Using game design elements in non-game contexts is one of the good ways to motivate and encourage students to learn programming. To assess how gamification impacted the learning experience, we compared data from one gamified and non-gamified year. In general, the results show significant improvements in terms of attendance to class, participation, and proactivity. They also suggest that our approach can reduce the high rate of failure grade among students. In conclusion, this case study, we show how the use of concepts related to gamification can improve motivation, passion, beauty, joy, awe, e naturally the succeed in programming.

Keywords— *Gamification, learning programming, teaching programming, programming problems.*

I. INTRODUCTION

Today's students need to be able to adapt to a dynamic environment surrounded by new technologies. Basic computer literacy is not enough to stay competitive in the current workforce and have the necessary competencies for the 21st century [1]. It has become essential that students develop a deeper understanding about computing and adequately apply computing skills. The developing of skills like creativity, problem-solving, persistence, collaboration, communication, and critical thinking ability, are crucial skills to a student's future success in the face of constantly evolving technology regardless of their area of study [2], [3]. The programming courses are usually considered to be effective in fostering and improves these skills. Many university students are studying basic programming, but unfortunately, they have difficulties.

Programming is hard. In [4], Dijkstra mention that the programming is much more an intellectual challenge the art of programming is the art of organizing complexity. In the last decades, there has been an intense research activity in

studying the difficulties in teaching and learning introductory programming. The substantiation of this situation is well documented in the work developed by Bennedsen in [5], where several works are presented on the subject of difficulties introductory programming courses. In the research work, it is clearly identified that it is well known in the Computer Science Education community that students have difficulty with programming courses. Teaching programming to novices has been considered a big challenge in computing education. Several studies reflect on the factors that negatively influence the teaching / learning of programming, as the examples presented in the works described in [2], [6]–[9]. In [10], factors are described as: the abstract concepts that programming implies; the competencies that are necessary for the resolution of problems; the mental abilities necessary for the decomposition of problems; for many students it is the first contact with computational thinking and programming; the students have to know the syntaxes, the semantics and the structure of a new non-natural language, in a short space of time. As a consequence of that, initial programming courses are often characterized by a great deal of learning frustration, a fairly significant student drop rate, and students with difficulties lose interest and motivation [5]. One way to improve motivation, passion, beauty, joy, awe and naturally the succeed in programming is to use gamification.

The organization of the article is as follows. Section 2 contains an overview of the related work. The methodology work will be introduced in Section 3 and the analysis and discussion of the results in Section 4. Section 5 contains the conclusions.

II. RELATED WORK

Games attract millions of people all over the world, they spend many times performing often tasks just for fun [11]. This is an enormous power and gamification reflects how to use this power. The use of game mechanics in non-gaming contexts, meaning of gamification [12], appears to be an emerging trend in many sectors, and especially in education. In the last decade, several works focus on gamification as a tool to motivate students and increase their engagement in programming courses [13]–[19].

In paper [13], it presents a survey gamification tools used to teach computer programming. This paper also presents some important considerations about the use of gamification in teaching as the meaning of two main types of gamification - structural gamification and content gamification and, the concept of two types of academic motivation - intrinsic and extrinsic motivation. It also describes the meaning of the use of some of the mechanics of games like leaderboard, levels,

points, badges, awards, missions and others. Several works focus on present a proposal for a gamification framework for online distance courses to learn how to program [15], [16]. Moreover, there are also studies, that evaluated the learning effectiveness and engagement of a gamified learning activity targeted at the learning of C programming language, with successful results [14].

Paper [20], presents a follow-up of a systematic mapping study of research on applying gamification to education. The study also confirms that the penetration of gamification in education is still fast-growing and the practice has outpaced researchers' understanding of its mechanisms. The authors further state that the rise of the number of studies with inconclusive or negative results suggests that gamification has passed its early phase and is considered in a more critical and analytical.

III. METHODOLOGY WORK

The main motivation, in the accomplishment of this work, is to reduce the high rate of failure and lack of motivation of students in courses of introductory programming. This work follows the concept used in many videogames, where a player character has a score that summarizes his/her skills, and where these skills can be improved through problem-solving, collect objects and training. Similarly, the student will have a score that summarizes their programming skills and can train and improve their skills as needed.

In order to carry out this research, the identification of the gamification elements that are used in an educational context was defined as a goal, especially in introductory programming courses. Consequently, we need to know what elements/techniques of gamification working better in an educational context. Based on these principles we will test and experiment with a set of gamification techniques with the specific goal of improving and motivating students to learn to program. In subsection A we describe some of the gamification techniques to use, in subsection B we present how we will collect the data.

A. Building the student profiles with gamification techniques

The idea of building a profile with the student's programming competencies is based on the same concept of current video games such as FIFA 19 or Assassin's Creed. The characters are invited to build and improve their characteristics and skills in specific areas to complete their tasks or change their level. For example, a FIFA player may train a penalty, dribble, free kicks and corner kicks practice and other actions to improve his/her abilities during the game. Likewise, we want each student to be able to improve and deepen their programming skills by performing a set of appropriate and worked exercises for each student and situation [21].

From the teacher's point of view, it is important to know each student's score and, have a detailed view of each student's skills. As a consequence, put in place a system of action plans in the areas of weakness of the student. Over time, the teacher will be able to know the minimum score required at each stage of the learning process for success.

The score of each student, or points for skills, are collected throughout the programming classes, by

continuously monitoring and evaluating student activities. Some examples of activities for data collection are: participation in classes, curiosity and initiative, in doing new activities, try to write programs in the best way, results from activities related to the detection of cognitive reasoning abilities and spatial visualization, capacity of give and following instructions, capacity of using conditional structures, capacity of using iterative structures and making C programs. However, the set of activities to be performed uses some gamification techniques, such as the surprise box and elimination tournament. Surprise box is a set of activities of which one is drawn for the student to solve. The activities will be rated with points according to the degree of difficulty. Points will be awarded to the student after correct resolution. In an elimination tournament two students compete in solving an exercise. The student who loses is eliminated from the tournament. The winning student proceeds to the next stage. Points will be awarded for each phase reached in the tournament.

B. Data analysis

This study involved a group of 67 students, from the academic year 2019/2020, and a group of 87 students, from the academic year 2018/2019, of an introductory programming course (Introduction to Programming), lectured to the first year, first semester, students of the Computer Science (CS) course at the Polytechnic of Guarda (IPG), Portugal – an institution of higher education. In this course the C language is used to teach the basic programming concepts. The average IPG admission grade for 2018 was 12.0 ± 0.7 points. In the academic year of 2019, the average IPG admission grade was 12.3 ± 1.3 points, for computer science course. It should be noted that gamification techniques were only applied in the current academic year (2019/2020). The gamification techniques described in the previous subsection was applied in practical classes.

It is also important to note that the current academic year is not yet over, so it will not be possible to compare the final result of the two academic years. Thus, some of the activities developed throughout the course will be analyzed, as well as the attendance to classes and report the point of view of the teacher and students about the use of gamification in the classroom. The activities for evaluation and control, for both academic years, consisted of solving C programs, applying basic concepts such as integer division and remainder division, flow control and translate algorithms for C language.

Finally, students were asked to express their opinion on the use of gamification techniques such as leaderboard, points, levels, badges, immediate feedback, surprise box and elimination tournament. The questionnaires are rating scale questions type, with a scale between 0 and 10 (strongly unmotivating and strongly motivating). In the same questionnaire, for each question, an optional open-ended question is out for the student to make a comment. These questionnaires were answered by 33 students, which corresponds to approximately 50% of the students.

IV. RESULTS AND DISCUSSION

The main motivation is to reduce the high rate of failure and lack of motivation of students in courses of introductory programming. Unfortunately, it is not yet possible to evaluate the failure rate. However, it is possible to state that

students are more motivated for the learning process. To confirm this statement, we present Figure 1 where we can see that the student attendance rate in the current academic year is about 18.1% higher than the previous year. For the academic year 2018/2019, the average number of students per class is 35 ± 8.6 (40.6%) students, for the year 2019/2020 it is 39 ± 6.3 (58.7%) students.

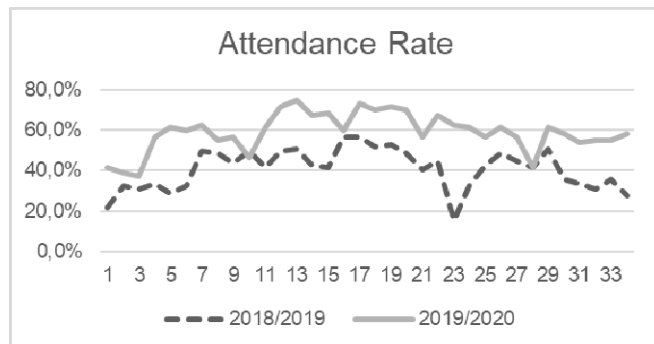


Fig. 1. Attendance rate for the academic year 2018/2019 and 2019/2020.

Table 1 presents the results of the practical activities performed in the classroom. As can be seen the results obtained in the academic year 2019/2020 are better. The average of the results obtained in 2018/2019 was 11.09 ± 5.6 and 10.12 ± 4.3 , in 2019/2020 were 11.10 ± 7.1 and 12.99 ± 5.6 values, respectively in activity 1 and activity 2. Although, the results of the activity 1 are not significantly better, the activity 2 results improved considerably. The weak result in activity 1, may be due to the fact that it was the first activity carried out and the few gamification activities carried out so far.

TABLE I. RESULTS OF ACTIVITIES

Interval	Academic year			
	2018/2019		2019/2020	
	Atv1	Atv2	Atv1	Atv2
]0, 10[37.7%	54.0%	43.1%	32.6%
[10, 14]	42.6%	30.0%	19.6%	23.3%
]14, 20]	19.7%	16.0%	37.3%	44.2%
Average	11.09 ± 5.6	10.12 ± 4.3	11.10 ± 7.1	12.99 ± 5.6

Students' opinions on the use of gamification techniques in the classroom such as leaderboard, points, badges present in Table II, and immediate feedback, surprise box, and elimination tournament are presented in Table III. As we can see, the students' opinion is very good. More than 80% of students rate their experience and the use of gamification techniques in the classroom between 5 and 10.

TABLE II. RESULTS LEADERBOARD, POINTS, AND BADGES

Interval	Gamification techniques					
	Leaderboard		Points		Badges	
[0,4]	1	3,0%	4	12,1%	2	6,1%
[5,7]	13	39,4%	15	45,5%	16	48,5%
[8,10]	19	57,6%	14	42,4%	15	45,5%

TABLE III. RESULTS IMMEDIATE FEEDBACK, SURPRISE BOX, AND ELIMINATION TOURNAMENT

Interval	Gamification techniques					
	Immediate Feedback		Surprise Box		Elimination tournament	
[0,4]	0	0,0%	0	0,0%	0	0,0%
[5,7]	11	33,3%	15	45,5%	10	30,3%
[8,10]	22	66,7%	18	54,5%	23	69,7%

In addition to this basic quantitative analysis, qualitative analysis of students' comments on their experience and perception of gamification is important. The following we described the most relevant of the student comments on the use of classroom gamification techniques. Regarding the use of the leaderboard, we highlight the following comments:

- "It is a great motivation to study programming".
- "It's a good strategy to get students to work harder and get more motivated".
- "It is very motivating for the first but very demotivating for the last of the table".
- "Students at the bottom of the list eventually give up".
- "It puts a lot of stress on students at the bottom of the list because they can't get out and are clearly identified".
- "We must be care do not to cause divisions between students, because they occupy a very low place on the leaderboard".

About using points:

- "A student having a grade is something that the student gets very nervous about, and with this method, the students 'have fun' and do things better without pressure".
- "Points, or grade, whatever because both for me have the same goal".
- "In my point of view, the grade is more attractive".

Comments on the use of badges.

- "Badges, unlike the other options, do not have such a quantitative connotation as compared to the traditional assessment".
- "As in the games, 'trophy hunting' is very appealing".
- "Having to keep knowledge in order not to lose badges will lead students to strive to keep knowledge up to date".
- "Badge system, I do not consider to be motivating because not everyone has this desire to win".

Comments about surprise box.

- "Good idea, like this and always something new".
- "It may be interesting as long as the level of the exercises is balanced".
- "This system is simple and greatly helps less motivated students to earn points".

- “Allowing the student to change activity if unable to resolve and be penalized for it”.

Students' opinions about the elimination tournament.

- “This kind of activity gives more desire to do the exercises and gives us another motivation and another goal”.
- “The idea of a tournament is quite interesting but not everyone wants to compete because they are afraid to compete against the ‘sharks’”.
- “It is very motivating”.
- “For students eliminated from the tournament, do another tournament, like Champions League and Europa League”.
- “Compete according to leaderboard ranking”.

Finally, some overall comments made by the students.

- “You have to keep in mind that a lot of people play for fun so by making this a ‘game’ many people can stop taking it seriously which can bring undesirable results”.
- “Gamification in the classroom has everything to make teaching more appealing and to make students happy to participate in class”.

Overall, the comments received in response to this invitation confirmed that the use of gamification techniques are very important and motivating. It should be noted the negative effect, stated by the students, that the use of a leaderboard can demotivate and stress the students at the end of the table. And more attention should be given to choosing the pairs of competitors in each stage of the elimination tournament. Very interesting are the general comments at the end of the questionnaires that the use of classroom gamification techniques in the programming course can be very interesting and motivating. But it may also lead students to think this is just kidding, and do not take classes as a serious matter.

V. CONCLUSIONS

The results are encouraging compared to previous years. We observed that students are more involved and motivated in the learning process, actively participating in all the challenges and activities requested. The number of students who successfully completed the evaluation activities increased significantly. On the other hand, the teacher has a great knowledge of each student's weaknesses and strengths in all topics of the programming learning process.

ACKNOWLEDGMENT

We thank our students from the academic year 2019/2020, from the Introduction to Programming course, of the Computer Science (CS) course at the Polytechnic of Guarda (IPG), Portugal, for their availability and participation in all activities.

REFERENCES

- [1] S. Dekhane, X. Xu, and M. Y. Tsoi, “Mobile app development to increase student engagement and problem solving skills,” *J. Inf. Syst.*

- [2] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, “A study of the difficulties of novice programmers,” *ACM SIGCSE Bull.*, vol. 37, no. 3, p. 14, Sep. 2005.
- [3] M. McCracken *et al.*, “A multi-national, multi-institutional study of assessment of programming skills of first-year CS students,” *ACM SIGCSE Bull.*, vol. 33, no. 4, p. 125, Dec. 2001.
- [4] E. W. Dijkstra, “Notes on structured programming,” 1969. [Online]. Available: <http://www.cs.utexas.edu/users/EWD/ewd02xx/EWD249.PDF>. [Accessed: 28-Nov-2019].
- [5] J. Bennesden, “Teaching and learning introductory programming: a model-based approach,” University of Oslo, 2008.
- [6] S. Shuhidan, M. Hamilton, and D. D’Souza, “A taxonomic study of novice programming summative assessment,” *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 95, pp. 147–156, 2009.
- [7] A. de J. Gomes, “Dificuldades de aprendizagem de programação de computadores: contributos para a sua compreensão e resolução,” *Dificuldades Aprendiz. programação Comput. Contrib. para a sua compreensão e resolução*, 2010.
- [8] A. Robins, J. Rountree, N. Rountree, A. Robins, J. Rountree, and N. Rountree, “Learning and Teaching Programming: A Review and Discussion,” vol. 3408, no. April, pp. 37–41, 2016.
- [9] S. Fincher *et al.*, “Computer Science at Kent programming courses,” no. 1, 2005.
- [10] T. Jenkins, “On the Difficulty of Learning to Program,” *Language (Baltim.)*, vol. 4, pp. 53–58, 2002.
- [11] S. Deterding, “Gamification: designing for motivation,” *Interactions*, vol. 19, no. 4, p. 14, 2012.
- [12] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, “From game design elements to gamefulness: Defining ‘gamification,’” *Proc. 15th Int. Acad. MindTrek Conf. Envisioning Futur. Media Environ. MindTrek 2011*, pp. 9–15, 2011.
- [13] R. Elshiekh and L. Butgerit, “Using Gamification to Teach Students Programming Concepts,” *OALib*, vol. 04, no. 08, pp. 1–7, 2017.
- [14] M. B. Ibanez, A. Di-Serio, and C. Delgado-Kloos, “Gamification for engaging computer science students in learning activities: A case study,” *IEEE Trans. Learn. Technol.*, vol. 7, no. 3, pp. 291–301, 2014.
- [15] M. Piteira, C. J. Costa, and M. Aparicio, “Computer Programming Learning: How to Apply Gamification on Online Courses?,” *J. Inf. Syst. Eng. Manag.*, vol. 3, no. 2, 2018.
- [16] R. Queirós, “PROud-A gamification framework based on programming exercises usage data,” *Inf.*, vol. 10, no. 2, pp. 1–14, 2019.
- [17] S. Deterding, K. O’Hara, M. Sicart, D. Dixon, and L. Nacke, “Gamification: Using game design elements in non-gaming contexts,” *Conf. Hum. Factors Comput. Syst. - Proc.*, pp. 2425–2428, 2011.
- [18] B. Kumar, “Gamification in Education - Learn Computer Games with Fun,” no. 2, pp. 46–53, 2012.
- [19] L. E. Nacke, K. Harrigan, and N. Randall, “Gamification 2013: Proceedings of the First International Conference on Gameful Design, Research, and Applications: October 2-4, 2013, Stratford, Ontario, Canada,” *ACM Int. Conf. Proceeding Ser.*, 2013.
- [20] D. Dicheva and C. Dichev, “Gamification in Education: Where Are We in 2015?,” 2015.
- [21] J. Figueiredo, N. Lopes, and F. J. García-Peñalvo, “Predicting Student Failure in an Introductory Programming Course with Multiple Back-Propagation,” in *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM’19*, 2019, pp. 44–49.