

Learning Computer Programming: A Gamified Approach

Mário Pinto  

uniMAD – ESMAD, Polytechnic of Porto, Portugal

Teresa Terroso  

uniMAD – ESMAD, Polytechnic of Porto, Portugal

Abstract

Learning computer programming is a difficult task for most students who start learning in this field. In fact, many students refer that learning computer programming is an arduous and difficult task, presenting some fear in addressing these issues. However, the main challenge for beginners is not in the language or syntax, but in devising a solution to solve the proposed problem. On the other hand, the younger audience is used to clicking on an icon and seeing an application with an appealing interface! Thus, students are often discouraged when, in a classroom, they have to implement an algorithm to classify numbers or sequences of characters and print them, sometimes in unappealing development environments. The lack of immediate feedback on the solution proposed by the student is another aspect that promotes some demotivation, as students often have no real idea of where they went wrong and how they can improve the solution they present.

This paper describes the introduction of gamification elements in an introductory programming course, based on a conceptual framework proposed by Piteira. The main objective is to motivate and involve students in the learning process, through the introduction of strategies based on gamification (such as challenges, progression and levels), as well as strategies such as problem-based learning, seeking to make teaching programming more attractive and appealing to students. The paper describes a case study carried out in the course of algorithms and data structures in School of Media Arts and Design, at Polytechnic of Porto, in Portugal.

2012 ACM Subject Classification Computing methodologies; Applied computing → Computer-managed instruction

Keywords and phrases programming learning, teaching methodologies, teaching-learning, higher education, gamification

Digital Object Identifier 10.4230/OASICS.ICPEC.2022.11

1 Introduction

Learning to program is often a difficult process to which traditional teaching approaches have not been able to respond effectively [15, 5, 7, 10]. Introduction to programming courses or disciplines have, with some frequency, failure rates above 50%, according to several studies. This reality also occurs in higher education institutions, in courses and areas of computer science and Web development. Numerous studies have been carried out on this reality, and point to some causes, such as [8, 12, 7]:

- Difficulties in interpreting and understanding the proposed problems, often leading students to start solving a problem without fully understanding it;
- Difficulties in solving logical and concrete problems, in the form of algorithms;
- Difficulties in capturing students' attention and interest in learning the fundamentals of programming, with learning often oriented towards solving more or less abstract problems and using unappealing interfaces.

Teaching programming can also be difficult, because learning programming requires much more than acquiring technical knowledge and skills. Students should learn about programming structures, logic, and syntax of the language used. But they must also quickly



© Mário Pinto and Teresa Terroso;
licensed under Creative Commons License CC-BY 4.0

Third International Computer Programming Education Conference (ICPEC 2022).

Editors: Alberto Simões and João Carlos Silva; Article No. 11; pp.11:1–11:8

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

begin to build strategies to combine the knowledge acquired in solving problems through algorithms and their computer coding. The task is not yet complete because the code must be tested, debugged, and often reworked and optimized. In this context, some programming learning approaches and strategies tend to be adopted, in order to relieve the difficulties felt by students, such as [2, 6]:

- The introduction of dynamic elements in learning strategies, such as gamification, which uses game design elements in educational contexts. The introduction of techniques and strategies that include elements such as missions or challenges, embedded in a learning narrative, progression mechanisms such as points or levels, relationships that promote cooperation in the development of a solution and peer evaluation, among others, can help to increase student involvement and motivation [14];
- Automatic code evaluation systems, based on intelligent tutors. These systems allow students to progress in their learning outside the classroom environment, having automatic and immediate feedback on the solutions they submit. They also make it possible to increase students' degree of autonomy, often providing personalized learning paths depending on the profile and skills of each student [1];
- Systems that use visual representations, animations and simulation of algorithms, seeking to make learning more dynamic, visual and interactive. These are animation systems with the purpose of appealing to the potential of the human visual system, contributing to a better understanding of inherently dynamic concepts, when compared to the textual format.

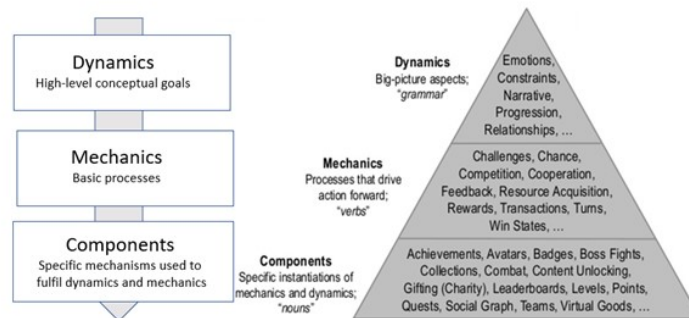
These approaches do not, by themselves, solve the difficulties felt in the initial learning of programming. But when combined with appropriate pedagogical strategies, they can help to relieve these same difficulties, converting learning into a more accessible and motivating process for students.

Thus, this paper begins by contextualizing the concept of gamification elements, distinguishing them into components, mechanics and dynamics, followed by a brief summary about gamification design frameworks, which can guide us in the process of implementing a gamified strategy. Section 3 describes the methodology adopted, which is based on a simplified approach to the 6D framework [16], applied to the teaching context. The following section presents the results obtained in the curricular unit under study, followed by its discussion and brief conclusions.

2 Gamification in programming learning

Gamification is the use of typical game elements (such as experience points, badges, progress indicators, levels or achievements) in other different contexts than games, in order to involve and motivate users in achieving a specific goal. The gamification of teaching and learning environments aims to involve and motivate students through the inclusion of elements and mechanics present in game design. This inclusion makes it possible to reward students for successful daily tasks, provide instant and personalized feedback, present their progress in the course, and foster healthy cooperation and competition among students. In this context, gamification also seeks to involve students in learning activities, making them more fun and motivating. One of the main challenges in introductory programming education is the need to capture students' attention and motivation. Especially the younger audience, is accustomed to clicking on an icon and see an application with appealing interface popup! Thus, they become unmotivated when, in a classroom, they are asked to implement an algorithm to sort numbers or strings, and print them in the console, for instance. Gamification seeks

precisely to address this problem through the implementation of new strategies for the development of knowledge, often also based on pedagogical approaches such as problem-based learning. According to some authors, developing a learning project based on gamification mechanisms, implies the implementation of three conceptual levels [16, 7]: i) components (such as achievements, badges, levels, points, etc.); ii) mechanisms (such as challenges, competition, cooperation, rewards, etc.); iii) dynamics (such as emoticons, relationships, progression status, etc.). Figure 1 shows an overview of game elements addressed for a gamification project implementation.



■ **Figure 1** Game elements pyramid for gamification [16].

Dynamics encompasses the big picture aspects of a gamified system. At the top of the pyramid, they are the most high-level conceptual elements in a game or gamified system. These are factors that absolutely must be considered, even if they don't enter directly into the game itself. Some examples of dynamics elements: constraints, emotions, narrative, progression, and relationships. The second group of elements is the mechanics. These are the basic processes that drive users to engage with the content and continue to drive the action forward, such as: challenges, chance, competition, cooperation, feedback, resource acquisition, rewards, transactions, turns, and win states. Components make up the largest group of game elements. In many ways the components are a more specific form of either dynamics and mechanics. These elements are less abstract than the first two categories and lead to actual tools that can be employed to begin to incorporate gamification in the environment of interest [10]: points, levels, badges, bonuses, notifications or peer evaluation of students are some of the elements that we can use at this level of abstraction [4, 9]. Based on the game elements pyramid for gamification (Figure 1), the gamified strategy adopted in this project sought to encourage students to progress in their teaching/learning process, in a more engaging and autonomous way. In this context, a mission was proposed to the students, based on 3 challenges. The challenges were associated with gamification elements such as points, content unlocking (according to the progress in challenges) and badges. Peer evaluation (by students) was also included, with the aim of rewarding those who were most active in cooperating with their colleagues. Challenges included various mechanisms for obtaining feedback from student work, such as:

- Points: based on user scenarios, points can contribute to other game mechanics, such as levels or progression. Points can also confer a bonus on the student's final evaluation. Points can be earned by completing the proposed challenges. We expected that students would get engaged to the acquisition of points and on re-taking their study.

11:4 Learning Computer Programming: A Gamified Approach

- Levels: in an educational context, levels allow progression and sequence through contents and activities. Each module activity was designed to a level. Inside the module, students have sub-levels. Students needed to complete the activities to reach the next. You may also use levels to unlock content – new problems to solve!
- Badges: which are medals that reward users for specific behaviours, are some of the most visible elements of gamification because they confer status. Badges are most useful for students who rank high in external motivation.
- Peer evaluation of students: The goal is also to introduce some ethics in the process of student's assessment, rewarding those who are somehow recognized by their peers with better performance (be it more skills, greater availability to help colleagues, ability to cooperate, etc.). It often seeks to encourage cooperation, collaboration and recognition by your peers.

However, some authors [10, 11] emphasizes that deploying the appropriate mechanics and components actually comes at the very end of the design process. Several authors propose different frameworks that help to systematize the programming teaching/learning process, many of them using gamification elements. This paper does not seek to carry out an in-depth literature review on the processes of designing a gamified teaching-oriented strategy. However, in recent years there has been an effort to formalize the process of designing a gamification strategy, and several gamification design frameworks have emerged, such as MDA (Mechanics, Dynamics and Aesthetics), Octalysis, GAME (Gather, Act, Measure, Enrich) or the 6D framework [3, 13]. Piteira [8] in a recent research, proposes a conceptual framework to implement gamification on courses of computer programming learning. The proposed framework includes several progression steps, such as: i) knowing the target audience; ii) Learning goals and learning outcomes; iii) Structure of gamified learning; iv) Identify and organize the study contents; v) Apply gamification elements appropriate to the context, and considering the aspects mentioned in the above steps [16, 11].

3 Methodology Adopted

Recognizing the difficulties that students normally present in learning computer programming, we tried to define a new approach to improve the success rate of students in this field. Based on the concepts of the increasingly popular gamification, we tried to introduce some mechanisms of game design, in the curricular unit of algorithms and data structures.

This curricular unit takes place in 15 weeks, during the 1st semester of the 1st year, of the degree in technologies and information systems for the web, at Escola Superior de Media Artes e Design, Polytechnic of Porto. The course was attended by 56 students, and it was designed to cover the fundamental concepts of programming introduction using the Python programming language. The course was organized in 5 main modules: i) fundamentals of programming: data types, variables, operators, simple data structures, basic control structures (if, while, for, . . .); ii) more complex data structures such as arrays, lists, queues and stacks; iii) data persistence (files); iv) introduction to GUI programming: basic components (such as labels, text boxes, buttons, radio buttons, images, timers, etc.), objects properties and events; v) development of a final project, applying the previous concepts in the development of an application.

From the first week, the teaching/learning process was oriented towards the exposition of small theoretical-practical contents, followed by carrying out some practical exercises of application. However, from the beginning, there were many difficulties in understanding on the part of the students, as well as some lack of motivation and interest in carrying out

the proposed practical activities. Once the first knowledge assessment test was carried out, scheduled for week 7 of the course, it was found that only 30 of the 56 students took the test (about 54% of the students). And among these, only 11 obtained a positive evaluation (representing about 37% of those who attend the evaluation moment, and only 20% of the enrolled students). So, considering:

- The results obtained, truly discouraging;
- The students' difficulty in assimilating basic programming concepts and applying them in practice;
- The lack of motivation of most students;
- The inability to understand and solve the proposed problems.

We tried to introduce new strategies, to respond to the difficulties felt by the students in this introductory course. The objective was to promote a new approach, based on the paradigm of gamification learning, as well as strategies such as problem-based learning, seeking to make teaching more attractive and appealing to students.

In this case study, the framework presented by Piteira et al [8, 9] was adopted, since it is oriented towards programming learning courses. Furthermore, the framework was applied with remarkable success by its authors, thus providing an opportunity to confirm the results obtained in previous studies. According to the process of implementing a gamified strategy, described in the previous section, a new plan was drawn, organized in the following steps (Table 1):

At the end of the gamified activity, a questionnaire survey was carried out, in order to understand the students' degree satisfaction with this initiative. The main objectives of the questionnaire focused on understanding the students' feedback on the gamified activity carried out, as well as if it was useful to extend this initiative to other modules of the course. Before the questionnaire was applied, we conducted a pre-test applied to a group of ten people, with similar characteristics to the final sample, to identify omissions and the level of understanding of the questions addressed in the questionnaire. The data collection took place at the end of the activity, between January 3 and 10, 2022. We received 21 responses, which corresponds to more than 90% of the students who joined this gamified initiative.

4 Results and Discussion

The questionnaire had four questions, with answers based on a likert scale, between 1 and 5, where 1 meant very little and 5 very much. Table 2 summarizes the answers obtained in these four questions.

Analyzing the results obtained, and considering the responses classified as levels 4 and 5 on the Likert scale (as much or very much), we can see that most students were satisfied or very satisfied with the strategy adopted. Around 100% of the respondents considered that gamification helped them to improve their levels of motivation and involvement in the teaching/learning process, as well as allowing them to develop skills in programming more easily.

It should be noted that only 11 students had obtained a positive evaluation in the first moment of evaluation (representing about 37% of those who attend the evaluation moment, and only 20% of the enrolled students), which demonstrates the enormous difficulties felt.

It should also be noted that most students found it useful to replicate this teaching and learning strategy, which incorporates some gamification elements, in other modules of the discipline.

■ **Table 1** Process of implementing a gamified strategy.

Process (step) of implementing a gamified strategy	Description
Target Audience	Students enrolled in the discipline of Algorithms and Data Structures. The joining to this gamified strategy was optional, consisting of a complementary way to the teaching and learning process previously defined.
Learning objectives	i) Consolidate knowledge about simple data structures, lists and arrays, as well as data persistence (the first three modules of the course, described above); ii) Develop skills related to problem-solving capability; iii) Develop student autonomy and self-expression.
Structure of gamified learning	The gamified strategy involved conceiving a mission, proposed to students, which is composed by 3 challenges. These challenges were organized into levels (unlockable, once students successfully complete the previous level), with successively higher degrees of difficulty. Once students have completed a level, they earned points, which are later converted into supplementary points in the final assessment of the course (cumulative points with all other assessment elements initially planned). They also earned a badge for completing each level. In addition, students are also encouraged to collaborate and cooperate with each other, through participation in internal discussion forums. At the end of the mission, the students would evaluate their peers, choosing the 3 students who stood out the most in helping and tutoring their colleagues.
Identify and organize the study	As mentioned, the mission was organized around 3 challenges: i) the first consisted of implementing a small rooster game (tic tac toe), a traditional game. It should consist on 2 players, who playing alternately; ii) After completing this first level, the second challenge consisted of adapting the same game, where one of the players is the computer. In this case, the algorithm should simulate the computer's movements, based on principle of playing against the computer; iii) once completed and submitted this challenge, the third level was unlocked. This level consisted on adapting the same game, but now the validation criteria for the assessment of the submitted solution was now focused on the efficiency with which the game simulated the computer movements. At this third level, efficiency and code quality were decisive key factors in awarding solutions. It is important to note that this third level awarded only the 3 best solutions submitted by students.
Apply gamification elements	This initiative is available on moodle between December 20, 2021 and January 3, 2022. The proposed mission should be completed in this period, for all those who joined this initiative (remember that it was not mandatory). One of the goals was to keep students motivated and involved in the discipline, even during the Christmas break.

5 Conclusion and Future Work

The experience presented in this paper describes the introduction of some gamification elements in the programming teaching/learning process, namely in an introductory course, based on a conceptual framework presented by Piteira. The results obtained are clearly in line with those presented in the study carried out by Piteira [9], confirming that this approach helped to achieve the desired results, with an increase in student motivation and involvement. Greater satisfaction, motivation and interest are clearly conclusions that we can draw from the questionnaires carried out.

■ **Table 2** summary of responses obtained to the questionnaire.

Questionnaire questions	1	2	3	4	5
How satisfied are you with the gamified strategy adopted?			1	4	16
Does gamification help you to improve your levels of motivation and engagement with the learning process?				3	18
Did the challenges proposed helped you to develop your skills in programming?			1	5	15
Do you think gamification should be applied to other course modules?				4	17

However, further studies would be needed to allow us to compare the results obtained with those that would have been obtained if we continued the more expository teaching/learning process, followed by exercises, which was adopted in the first weeks of the course (and until the first moment of evaluation).

On the other hand, more substantial conclusions also imply a comparison with the results obtained in previous years, in the same course, as well as an analysis of results over a longer period of time. That is, it will be necessary to replicate this model in the following academic years, in order to obtain a data set that allows to perform a trend analysis of results and thus more informed conclusions.

References

- 1 Kirsti Ala-Mutka. A survey of automated assessment approaches for programming assignments. *Computer Science Education*, 15:83–102, June 2005. doi:10.1080/08993400500150747.
- 2 Yoram Bosse and Marco Aurelio Gerosa. Why is programming so difficult to learn?: Patterns of difficulties related to programming learning mid-stage. *ACM SIGSOFT Software Engineering Notes*, 41:1–6, January 2017. doi:10.1145/3011286.3011301.
- 3 Patrick Buckley, Seamus Noonan, Conor Geary, Thomas Mackessy, and Eoghan Nagle. An empirical study of gamification frameworks. *Journal of Organizational and End User Computing*, 31:22–38, January 2019. doi:10.4018/JOEUC.2019010102.
- 4 Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. Gamification in education: A systematic mapping study. *Educational Technology & Society*, 18:75–88, July 2015.
- 5 Anabela Gomes, Cristiana Areias, Joana Henriques, and Antonio Mendes. Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. *Revista Portuguesa de Pedagogia*, 42:161–179, July 2008. doi:10.14195/1647-8614_42-2_9.
- 6 José Paiva, José Leal, and Ricardo Queiros. *Authoring Game-Based Programming Challenges to Improve Students' Motivation*, pages 602–613. *Advances in Intelligent Systems and Computing*, Springer, January 2020. doi:10.1007/978-3-030-11932-4_57.
- 7 M. Pinto. Learning computer programming: The role of gamification. In *EDULEARN18 Proceedings*, 10th International Conference on Education and New Learning Technologies, pages 9492–9497. IATED, 2-4 July, 2018 2018. doi:10.21125/edulearn.2018.2263.
- 8 Martinha Piteira and Carlos Costa. Computer programming and novice programmers. In *Proceedings of the Workshop on Information Systems and Design of Communication*, pages 51–53, June 2012. doi:10.1145/2311917.2311927.

- 9 Martinha Piteira, Carlos Costa, and Manuela Aparicio. A conceptual framework to implement gamification on online courses of computer programming learning: Implementation. In *Proceedings CISTI 2017*, pages 7022–7031, November 2017. doi:10.21125/iceri.2017.1865.
- 10 Ricardo Queirós, Mário Pinto, and Teresa Terroso. Computer Programming Education in Portuguese Universities. In Ricardo Queirós, Filipe Portela, Mário Pinto, and Alberto Simões, editors, *First International Computer Programming Education Conference (ICPEC 2020)*, volume 81 of *OpenAccess Series in Informatics (OASIs)*, pages 21:1–21:11, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/OASIs.ICPEC.2020.21.
- 11 Ricardo Queiros, Mário Pinto, Alberto Simões, and Filipe Portela. *A Primer on Gamification Standardization*, pages 1–13. IGI Global, January 2022. doi:10.4018/978-1-7998-8089-9.ch001.
- 12 Jože Rugelj and Maria Lapina. Game design based learning of programming. In *proceeding of International Scientific Conference Innovative Approaches to the Application of Digital Technologies in Education and Research (SLET-2019)*, May 2019.
- 13 Alberto Simões. *Using Game Frameworks to Teach Computer Programming*, chapter 10, pages 221–236. IGI Global, January 2017. doi:10.4018/978-1-5225-1034-5.ch010.
- 14 Jakub Swacha and Pawel Baszuro. Gamification-based e-learning platform for computer programming education. In *X World Conference on Computers in Education*, June 2013.
- 15 Christopher Watson and Fred Li. Failure rates in introductory programming revisited. *ITICSE 2014 - Proceedings of the 2014 Innovation and Technology in Computer Science Education Conference*, June 2014. doi:10.1145/2591708.2591749.
- 16 Kevin Werbach and Dan Hunter. *For the Win: How Game Thinking can Revolutionize your Business*. Wharton Digital Press, January 2012.