Gamification to Develop Coding Skills

Ciro Martins

ciro.martins@ua.pt
Águeda School of Technology and Management – University of Aveiro (PORTUGAL)
Institute of Electronics and Informatics Engineering of Aveiro (IEETA) (PORTUGAL)

Fábio Marques

fabio@ua.pt
Águeda School of Technology and Management – University of Aveiro (PORTUGAL)
Institute of Electronics and Informatics Engineering of Aveiro (IEETA) (PORTUGAL)

Ana Balula

balula@ua.pt
Águeda School of Technology and Management – University of Aveiro (PORTUGAL)
Research Centre "Didactics and Technology in Education of Trainers" (PORTUGAL)

Abstract – Digital technologies are increasingly assuming a crucial role in daily routines and in our professional lives, and, thereof, they are more and more relevant to provide solid answers to emerging societal challenges. Nonetheless, in initial programming curricular units, it is not always easy to manage the students' different knowledge level, especially due to their diverse learning backgrounds. In this scenario, a teaching/learning strategy was designed to minimize the students' heterogeneity (as well as the constraints that spur from it) and try to guarantee learning equity among them. Thus, the strategy was based on a flipped classroom approach, which implies the students' autonomous work prior to face-to-face (in-class) work. For that purpose, two platforms were chosen with different purposes: i) *Code Avengers* was selected to create controlled, practical learning and teaching experiences in a gamified environment, and ii) *Mooshak* for the teacher for automatic assessment, to monitor and give feedback to students in online environment. The global perception as to the strategy implementation is that it has great potential, i.e. it had a positive impact on the students' interest in developing coding skills, in particular due to its gamification facet and the instant feedback provided to students.

Key-words: programming skills, learning motivation, flipped classroom, gamification

Introduction

Nowadays, almost everything has a digital facet or tends to be interconnected through technology. Consequently, knowing how to code is more valuable than ever and coding skills are treasured in a growing range of diversified areas, namely: medicine, science, engineering, marketing, finance...

Particularly in Higher Education settings, students should already be fairly independent, pursuing learning opportunities, and centring the learning process on their theoretical and practical educational needs. Nonetheless, the students tend to lack autonomy and self-directness, and their manifold schooling backgrounds translate into very heterogeneous classes (with well- and ill-prepared students). This raises some in-class problems, namely: i) less proficient students have to put a greater effort into following the topics taught in class, ii) students with greater difficulties quickly lose interest and have a higher dropout rate, and iii) teachers find it difficult to manage the students' heterogeneity.

In order to address this problem, i.e. the students' different expertise levels in programming languages – a.k.a. coding skills –, a teaching/learning strategy was designed and tested, which implied the use of *Code Avengers* (Code Avengers, 2018) and *Mooshak* (Leal & Silva, 2003) as tools to develop coding proficiency resorting to game-based activities and to solve online exercises (and getting automatic feedback), respectively. The use of these platforms also allowed for the teacher's monitoring and assessing of the students' individual performance

To sum up, the main purpose of this work is to depict a teaching/learning strategy used in an introductory programming curricular unit, as presented in the following sections.

Context and main goal

In initial programming curricular units, it is very common for the students' knowledge level to be very different. Thereof and because this inequality (of knowledge) is also not always easy to manage inside the classroom, the main goal was to create and implement a teaching and learning strategy to be used in introductory programming curricular units. In this first experience, the strategy was tested in the course "Introduction to Programming" of a Higher Professional Technical Course, i.e. Network and Computer Systems.

The strategy was designed ought to boost the students' learning autonomy by assigning them programming tasks to be developed prior to in-class lessons, in which more in-depth, complex learning was to be developed (individually and collaboratively). This teaching approach can be defined as:

> "[the] pedagogical approach in which direct instruction moves from the group learning space to the individual learning space, and the resulting group space is transformed into a dynamic, interactive learning environment where the educator guides students as they apply concepts and engage creatively in the subject matter." (Flipped Learning Network, 2018)

According to Jenkins et al. (2017, 8), "with the growing access to vast information through the internet, the traditional model of teacher as the sole steward of knowledge has become obsolete". In flipped classrooms, floor is given to students by transferring responsibility to them for their learning, encouraging interaction and collaboration with peers, promoting their interest in the topics, as well as the development of critical thinking, redirecting the teaching process to the needs that stem from the students' learning processes (Bergmann & Sams, 2012) – and these were some of the most relevant assumptions underlying the strategy design as presented below.

The teaching/learning strategy

Several platforms have recently emerged, allowing users – beginners, as well as more experienced – to develop or deepen their coding skills in well-structured consecutive modules, with increasing complexity levels but that can be used at a self-paced rhythm. In the purposed strategy two different platforms were used, viz. *Code Avengers* and *Mooshak*. As to the former, it provides active and entertaining settings to develop coding skills, using technology to create controlled, practical learning and teaching experiences (Mateous & Aleman, 2009), as exemplified in Fig. 1.

Regarding pedagogical features, Code Avengers is structured sequential modules, entails different complexity levels and instructions are globally easy to understand. Moreover, it puts forth interesting and relevant examples of apps, games and websites in a gamified learning environment, and allows for students to learn at a self-paced rhythm and for teachers to remotely monitor their progress, as well as to provide (instant) feedback.

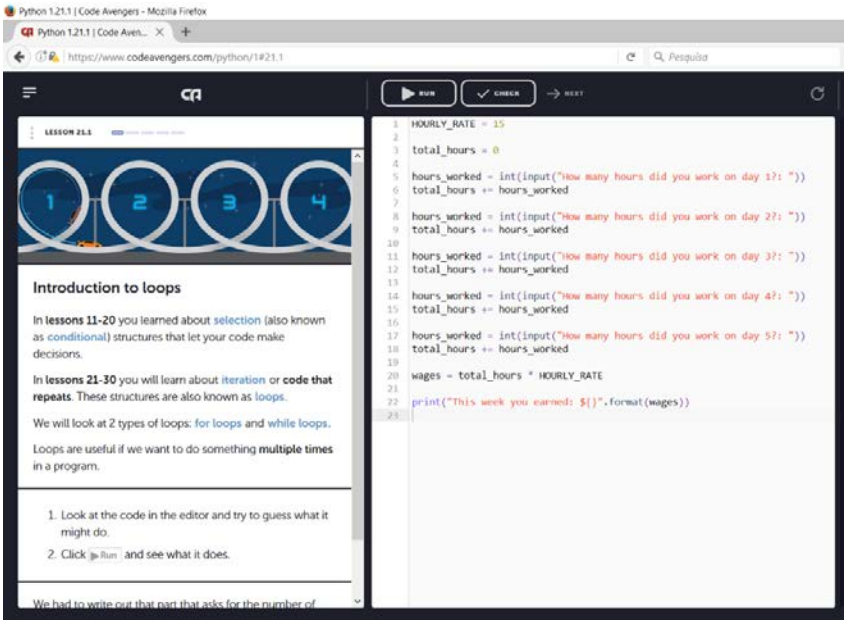

Figure 1. *Code Avengers* – online platform to develop coding skills

In this platform, students learn by embracing thought-provoking challenges. Within each module, students can also answer some quizzes – created to consolidate new knowledge from a more

theoretical perspective – and points, badges and leader boards are used to reward their good results. Moreover, the teachers' dashboard allows them to identify the topics students are struggling with and, thanks to live updates of each student's progress, students that need support can be more easily identified and help is much more effective.

Moreover, a tool for the automatic validation of the student's exercises and assessment outcomes was also used, i.e. Mooshak[8].This is an online tool currently used to manage programming contests (Rodrigues, Marques & Martins, 2014) and from the several features it provides, the most relevant for the strategy were: automatic assessment of submitted solutions and the possibility of providing feedback and clarifying doubts online.

In terms of operationalizations the students' performance in every task proposed was considered in their final grades (see Table 1), once it was of utmost importance to consider the learning process for assessment purposes. Thus, five different tasks were considered as presented in Table 1.

Elements	Task	Time	Final grade
1	7 sets of exercises (individual work in Code Avengers)	1 set before each face-to-face session (automatic grading)	15%
2	5 sets of exercises (individual work in Mooshak)	1 set during and after each face-to-face session (automatic grading)	15%
3	Project work (group work)	Project work	15%
4	Individual test (Mooshak)	4 practical exercises	20%
5	Individual test (Mooshak)	2 practical exercises	35%

Table 1. *Teaching/learning Strategy* – tasks

Final remarks

In terms of future work, data collected in both platforms will be triangulated and analysed to fully understand the impact of the strategy. The data set encompasses, for instance, the students partial performance and final results, the number of accomplished tasks, the time taken to solve the tasks (Code Avengers), when the tasks were finished, how many times students tried to solve a task before succeeding (Mooshak), etc.

Nonetheless, the teacher's perceptions regarding the strategy implemented is quite positive, once it resulted in a higher student engagement, which translated into a more homogeneous knowledge level in the classroom, generating in-depth and complex learning generated in class. It opened the possibility for students to learn at their own pace, always relying on the teachers monitoring and feedback when struggling with specific tasks. Besides, the inclusion of a group work also promoted

[8] See http://mooshak.dcc.fc.up.pt/.

the students' interest, because it was grounded on a game-based activity and implied the interaction and discussion of solutions with colleagues.

Finally, as to improvements, it became clear that the use of two platforms that were not interconnected posed some problems. Consequently, it is crucial to work on technological solutions that allow for the automatic aggregation of assessment results per student, updating results in real time. Furthermore, it might also be important to consider the possibility to display the class results (per student) in online social networks (those closer to the students most used online environments). This would probably, not only reinforce the students' sense of self-achievement, but also promote completion and competition with their peers – a very important feature in gamification.

References

Bergmann. J., & Sams, A. (2012). Flip your Classroom: reach every student in every class every day. Washington, DC: ISTE; and Alexandria, VA: ASCD.

Code Avengers (2018). Code Avengers. https://www.codeavengers.com/ (Accessible in 2 of February 2018).

Flipped Learning Network (FLN) (2014). Definition of Flipped Learning - Flipped Learning Network Hub. http://flippedlearning.org/definition-of-flipped-learning/ (Accessible in 2 of February 2018).

Jenkins, M., Bokosmaty, R., Brown, M., Browne, C., Gao, Q., Hanson, J., & Kupatadze, K. (2017). Enhancing the design and analysis of flipped learning strategies. Teaching & Learning Inquiry, 5(1). http://dx.doi.org/10.20343/teachlearninqu.5.1.6

Leal, J.P., & Silva, F. (2003). Mooshak: A Web-based multi-site programming contest system. Software: Practice and Experience, 33(6), 567-581.

Mateous, G., & Aleman, F. (2009). Make Learning Fun with Programming Contests. Transactions on Edutainment II, 246-257.

Rodrigues, M., Marques, F., & Martins, C. (2014). Teaching Advanced Computer Programming Topics using an Automatic Validation Tool: Challenges and Solutions. Proceedings of EDULEARN14 Conference (pp. 1541–1547).