A NEW GAME-BASED APPROACH FOR THE CREATIVE LEARNING OF PROGRAMMING IN THE DIGITAL MEDIA AREA

A. Coelho, L. Reis, J. Monteiro

FEUP / INESC TEC (PORTUGAL)

Abstract

Teaching in the scientific area of Digital Media has been evolving to meet the challenges facing our society, particularly to create technologies that effectively provide added value to the human being, both professionally and in the personal life.

This article presents a new teaching methodology that combines “creative programming” and “creatively learning programming”, intertwining the use of digital games as the metaphor for the programming constructs, and gamification to enhance engagement and regular work. This is backed up on the knowledge that the introductory learning of programming is a process that requires the acquisition of cognitive skills, which fit into what is defined as Computational Thinking, but also that the process of learning programming has a high level of complexity. This causes specific difficulties that have been observed over several years, mainly as the practice requires specific mathematical skills, not so common in Digital Media students.

The teaching methodology is based on a constructionist approach, with PBL, defining learning paths supported on digital games development. The difficulty inherent in constructing a problematic base, coherent with the learning process, led to the development of the Creative Programming Matrix, based on the triadic certification method. This matrix enables the teacher to select, organize and develop the best game examples to support the learning process. This work has been preliminarily evaluated through questionnaires to students and surveys to faculty that teaches programming courses. This method was instantiated in the restructuring of a course on a Master program in Multimedia. A set of gamified activities promoted a natural progression in the learning, essential for a first approach to the programming. From a set of clear objectives, the target behaviours and the profile of the students were defined, the cycles of involvement and the necessary tools were developed, not forgetting that learning should be fun! The results validate a higher efficacy and engagement of the students.

Keywords: Creative Programming, Games-based learning, Gamification, Learning programming, Digital Media.

1 INTRODUCTION

The profile of the professional in the area of digital media is hybrid and diversified, combining skills in the areas of Design, Art, Technology, and Communication. The dimension of Technology in Digital Media training has as the main objective to develop skills for the design and development of multimedia tools. Thus, the need to develop in the students the skills related to Computational Thinking and to provide them with the effective capacity to develop software in medium complexity projects.

Following the constructionism learning theory, we present the concept of "Introduction to Creative Programming", which combines a focus on the development of computational thinking, through a set of active pedagogical dynamics. Through problem-based learning (PBL), it appeals to creativity (games, for example) to foster essential skills for the development of low/medium complexity software. This course is designed through a gamification perspective, creating short iterative cycles that promote regular learning activity, combining a component supervised by the teacher and another component developed by the student in autonomy.

The course “Introduction to Creative Programming” is intended to provide the fundamentals of software development, but also integrates an introduction to interactive systems, particularly the design and development of digital games. The aim is to leverage learning programming, a typical analytical process, with the design and development of digital games, giving way to the creative capacity of students.
The process of learning programming requires the acquisition of cognitive skills, which fit into what is defined as Computational Thinking [1] and has high complexity [2]. This causes specific difficulties that have been observed over several years [3]. The conjugation of these two topics is proposed to overcome this well-identified problem of the lack of motivation, and even anxiety, for the learning of programming by a significant number of students in these programs.

Following the spirit of Bologna, the use of b-learning is fundamental to sustain the autonomy of the student in this process, with self-learning materials. The learning process is centered on the Digital Media student, with heterogeneous profile and motivated for the development of creative projects. The option to integrate the design and development of Digital Games enables a pedagogical approach to PBL, enhancing individual dynamics as well as group dynamics, which are intended to be early adopted. Finally, this entire learning process is designed based on gamification to ensure a regular and autonomous learning process, with high levels of motivation and involvement, but always under the guidance of the teacher. To this end, it is essential to support the activities through a learning management system (LMS).

The structure of this document mirrors this goal. Section 2 presents the basic concepts of Computational Thinking and Gamification, followed by related work that supports the proposed contribution. Section 3 defines the teaching methodology, including the methodological approach, the assessment, the gamification project, and the teaching method. This last subsection presents the main contribution – the Creative Programming Matrix. Section 4 presents a preliminary study and the results obtained. Finally, section 5 presents some conclusions and future work.

2 BACKGROUND AND RELATED WORK

The teaching methodology is based on the concepts of Computational Thinking, as the fundamental competences for success in learning programming, and on gamification for the motivation and emancipation of the learning process.

2.1 Computational Thinking

Computational Thinking was defined by Jeannette Wing in 2006 [1], although the term had already been published previously. In this seminal article, it is considered that the way to solve a computational problem can be used to solve many of the complex problems of daily life. It is no longer a specific computing competence, but an essential competency in the development of an individual.

The key to computational thinking is the concept of "abstraction", that is, how one grasps the essence of the problem and decomposes into subproblems until one finds patterns of solutions that can be easily automated through an algorithm.

Although several approaches have considered different cognitive processes, the four that stand out:

- Abstraction - Focus on the most relevant information by ignoring implementation details.
- Decomposition - Addressing a problem by decomposing it into simpler, easier-to-solve subproblems.
- Algorithm design - develop a detailed solution to a problem, step by step.
- Pattern Recognition - Identifying similarities between problems helps find standardized solutions.

2.2 Gamification

Gamification is defined as the use of techniques and game design elements in non-game contexts [4]. We can consider it as a UX design process that intends to promote specific behaviours on the user to improve her/his experience. The process of gamification presupposes an approach based on the "player's journey", starting in the first phase of "Onboarding", where the user identifies the goals that must achieve, is confronted with the "rules of the game" and motivated to "play the game". In a second phase, "Scaffolding", a learning path is built that promotes the acquisition of skills progressively, until finally reaching a final stage of greater challenge where the user demonstrates to have acquired the competencies and to have mastered the fundamental knowledge, the "Endgame".
Some gamification frameworks can be used to provide a more structured design of the solution. We have followed the Werbach's 6 steps to Gamification framework [5], that consists of a six-step design process:

1. Define business objectives;
2. Outline the target behaviours;
3. Describe the players;
4. Elaborate the activity cycles;
5. Do not forget the fun (motivation and involvement);
6. Implement the appropriate tools.

2.3 Related previous work

This work was leveraged by a set of research projects previously developed. The use of games in learning and in particular the certification of skills acquisition was the subject of a doctoral thesis that led to the development of the triadic certification method [6], which eventually served as the basis for the evaluation model and learning analytics of the H2020 BEACONING project [7]. It has also leveraged by a master's dissertation [8], which explores programming learning based on PBL, using the development of digital games as the motor of the learning process, but ensuring that the set of tutorials to be developed enables the acquisition of desired competencies according to the triadic certification model. In another master's dissertation [9], we developed gamification for the teaching of programming. The game elements and gamification activities were identified, as well as the possibilities of integration in LMS (in this case Moodle). Two studies were also developed, one with students and the other with faculty, which demonstrated the motivation of both groups for the gamification and the positive valorisation that gave to the use of scoring as a continuous evaluation mechanism.

3 TEACHING METHODOLOGY

In the preparation of this teaching methodology, special attention was paid to the creation of mechanisms to ensure students' motivation and regular work, which are essential for the success in learning programming. For this purpose, a gamified activity plan was created that allows the student to acquire skills in the field of computational thinking and the ability to develop software projects for medium complexity interactive systems.

3.1 Methodological Approach

The teaching methodology is based on a constructivist approach, with PBL, defining learning paths based on problems based on digital games. The difficulty inherent in constructing a problematic base coherent with the learning process led to the adoption of the triadic certification method [6]. The triadic certification method serves as a guideline for the preparation of the tutorials, ensuring a comprehensive approach to the contents of the program and its availability in the most appropriate sequence. These tutorials are developed by the student in the classroom, under the supervision of the teacher and, at the end of the lesson, programming challenges are introduced based on the games developed, to be solved by the student in autonomy. The reading of the main bibliography is done before the class, so that it can be reviewed and discussed by the teacher, in class, before the tutorial is done. Gamification is also used to motivate students to adopt a regular learning pace to acquire the necessary skills at each stage of the semester.

3.2 Assessment

We consider a heterogeneous assessment, combining a distributed component, with weekly challenges and practical examinations on a computer, with a final exam, where the student demonstrates the competencies acquired more comprehensively. The distributed component has a weight of 70% and the final exam 30%.

The distributed assessment consists of two components. Practical examination on a computer (4 times in the semester with a duration of 60 to 90 min each) that allow evaluating the ability of the student to be able to put into operation algorithms to solve specific low/medium complexity problems.
And a continuous assessment based on a gamified project that, through a scoring system, evaluates the student's attitude throughout the entire learning process.

3.3 Gamification Project

The gamification of the learning process took into account the framework Werbach's 6 steps for Gamification [5] and is described in this section.

3.3.1 Definition of objectives

The main goal is to develop a motivating and engaging learning environment that encourages students to study programming regularly, maintaining short-term (weekly) goals. In this way, it is possible to leverage the acquisition of computational thinking skills and the ability to develop in autonomy low/medium complexity programs, fundamental to progress along the course.

As an additional objective, it is intended to incorporate the creative skills of Multimedia students to take an introductory approach to the development of digital games, also serving as a motivational component for learning programming.

3.3.2 Outlining target behaviours

The second step focuses on the definition of behaviors that students must adopt (or correct) in order to achieve the objectives defined in the first step:

- Read the modules of the study book before the class
  For the class to be more active and to promote greater dynamism and discussion, it is important that the students read before the class the modules assigned in the previous week.
- Solve programming assignments on a regular basis
  It is essential that students regularly practice (weekly), in autonomy, programming assignments and that they undergo a formative self-assessment to detect by themselves their main gaps to be able to overcome it later on with the teacher in class.
- Promote the individualization of the learning process
  Students have previous heterogeneous knowledge of, and the pace of the learning processes thus becomes asymmetrical among students. So it is essential to promote the differentiation of the learning method according to their competences, avoiding too much pressure on students with greater difficulty and avoiding demotivation in students with higher competencies.
- Class attendance and punctuality
  It is essential for students to attend classes and be assiduous, maximizing both the pace of face-to-face classes and the guidance from the teachers.

3.3.3 Describe the players

Students in Digital Media have a heterogeneous background, both at the level of the basic training and the sociodemographic level, from recent graduates to professionals with a great experience.

3.3.4 Develop activity cycles

According to the target behaviors identified, two cycles of activities were defined: a cycle throughout the semester, which intends to map the “player’s journey” and another weekly cycle that outlines a scoring system that encourages the regular activity of students. The semester cycle (Fig. 1) was developed for 14 weeks of classes, with 2 weeks being used for Onboarding, 10 for Scaffolding and the final 2 weeks for Endgame.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Onboarding</td>
<td>Scaffolding</td>
<td>Endgame</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 1 - Cycle of activities for the semester*
In the first week of classes, the course is presented, describing the learning objectives, the early concepts and its methodology, namely the gamification process: the "rules of the game". The first two weeks are focused on Computational Thinking with a brief introduction to the programming language. We adopted a programming learning environment (PLE). In this way, we introduce the fundamental concepts of computational thinking and the lexicon and syntax of the programming language through the game mechanics.

From the 3rd week of classes, and over 10 weeks, the student enters the "Scaffolding" phase where learning will be promoted in weekly cycles, developing challenges of increasing difficulty. The student progresses through three levels called "Apprentice", "Pro" and "Expert":

- Apprentice (0 points): The initial level.
- Pro (500 points): Allows to eliminate the grade of the worst practical exam.
- Expert (1000 points): It also allows you to eliminate the worst practical examination. But the student can still replace another practical examination for a project.

Finally, in the "Endgame" phase, students are subject to a final assessment, the big challenge, consisting of two practical exams covering all topics taught, or a project for students who have reached the level of "Expert".

The progression between levels is performed through a scoring system where the student earns points for each challenge overcome. Levels and score allow students to measure their progress throughout the learning process and provide very relevant rewards such as eliminating the worst grade of the practical exams. These rewards are important to allow the student to recover from specific difficulties in the learning process and to individualize their path. Being the objective of the student to score 1000 points, several challenges are assigned throughout a weekly cycle with 10 iterations where the student can win up to 1110 points. This additional 10 % gives some tolerance to prevent the student from losing an opportunity because of a personal problem or unpredictable event.

The weekly cycle is shown in the following figure:

![Weekly activity cycle](image)

The first activity of the weekly cycle is to read the supporting text. To validate that the student has read the module, a multiplayer quiz game is used at the beginning of class (Kahoot). This game allows the teacher to summarize the most relevant topics in the reading of the supporting text and to validate if the students understood them. And also encourages students’ attendance at the beginning of the class, since points are scored for participation and answer correctly to a certain number of questions. The class continues with a brief synthesis and discussion of the programmatic contents of the readings through a set of examples, developed interactively with the students. In the second part of the class, an assignment (tutorial) is presented to be developed in autonomy by the students, with the supervision of the teacher. Participation in classes is registered and scored and, at the end of the class, challenges based on the games are presented to be developed at home. Finally, at the end of the week, students have to undergo formative self-assessment, with support of an automatic assessment system.

### 3.3.5 Do not forget the fun (motivation and engagement)

The topic of digital games provides the fun component, allowing the assignment developed in class to be then played by the student. Also, it is also possible to vary the mechanics and dynamics of the game, in the sense that the student can create her/his version, fomenting creative abilities. In a way, it seeks to obtain what the psychologists call "IKEA effect" [10] that promotes a higher value perception for the goods that were produced by the self.
Although this component will be further worked out in the future, the focus here is on students' intrinsic motivation, ensuring that students are aware of their progression and that immediate feedback is provided. In the case of the project assessment component, students are free to develop a project of their liking.

3.3.6 Implement the appropriate tools

A gamification project requires the use of a set of tools for its operationalization. In this case a scoring system, an automatic assessment system and a set of tutorials based on game development.

The scoring system allows the students to monitor their progress and outline their method of work according to the level they intend to achieve ("Apprentice", "Pro" and "Expert") – see Table 1.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>score</th>
<th>n. of challenges</th>
<th>total</th>
<th>duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading + quiz</td>
<td>15</td>
<td>10</td>
<td>150</td>
<td>120 min</td>
</tr>
<tr>
<td>Class</td>
<td>10</td>
<td>12</td>
<td>120</td>
<td>180 min</td>
</tr>
<tr>
<td>Homework</td>
<td>30</td>
<td>10</td>
<td>300</td>
<td>120 min</td>
</tr>
<tr>
<td>Self-evaluation</td>
<td>20</td>
<td>7</td>
<td>140</td>
<td>60 min</td>
</tr>
<tr>
<td>PE1, PE2</td>
<td>200</td>
<td>2</td>
<td>400</td>
<td>60 min</td>
</tr>
</tbody>
</table>

The activities of the weekly cycle add up to 710 points which makes it possible for a student to reach the level of "Pro" (500 points), allowing to eliminate the worst grade obtained in the practical examinations, but not "Expert". This level requires that the student reach 1000 points and get a minimum of 75% (300 points) in both initial practical examinations (PE1 and PE2). That is, filtering is done for the students who demonstrate the capacity to develop the optional project in autonomy. Out-of-class activities have an average weekly duration of 5 hours during these 10 weeks. It is important to consider that these points are also mapped in the continuous assessment, allowing the student to rate the effort spent in these activities, assigning a tangible value.

Fundamental to this project is the use of an automatic assessment tool for programming assignments developed at home. Produced entirely from scratch, it works seamlessly with the university mail server. In this way students can carry out their self-assessment by sending an e-mail to the address of the course, receiving the classification (in percentage) and the description of the errors obtained within a maximum period of 5 minutes after submission. This tool is also used for the assessment of practical tests in the computer, allowing students to know their grading soon after the test (immediate feedback).

3.4 The teaching method

The teaching method was developed based on the PBL approach and focused on a set of problems centred on the development of digital games. Thus, each digital game will promote the development of specific skills. A selection of games was defined to promote a smooth progression, for which the triadic certification method was used [6]. This method, adapted in [8] ensures that the set of games that were selected allows working the programmatic contents up to the level of the Bloom taxonomy specified in the learning objectives.

Each problem consists of 4 steps that allow the learning of the basic concepts. Beginning by playing the game, the student realizes the objectives. Next, a tutorial leads to the construction of this game, causing the student to become aware of how the various structures of the programmatic contents map into the mechanics of the game. After this phase, variations of these mechanics appear as assignments, promoting the mastery of these programming constructs. Finally, appealing to the student's creativity, the construction of a new game with similar mechanics is proposed as a challenge.
The topics and associated games used in the scaffolding phase (10 weeks) are described in Table 2.

Table 2 – Mapping of programmatic contents and games

<table>
<thead>
<tr>
<th></th>
<th>Introduction to the programming language: Development environment (IDE) and command line execution; lexicon and syntax; variables and primitive data types; expressions. A first game. Conditional control structures.</th>
</tr>
</thead>
</table>
| 1 | * Flip the coin (Heads or Tails)  
    | * Game "Rock, paper, scissors" |
| 2 | Recursion and iteration: loops in programming and the game loop. Procedural abstraction: functions. Definition of tests and detection of semantic errors (debugging). |
|   | * Game to guess the number from 1 to 100  
    | * Blackjack |
| 3 | Data structures: Vectors.  
    | * Mastermind  
    | * Slot machine |
| 4 | Data structures: Strings.  
    | * Hangman  
    | * Word puzzle |
| 5, 6, 7, 8 | Data structures: lists, queues, stacks and tables. |
| 7, 8 | * Card game – lists  
    | * Car racing game – queues  
    | * Towers of Hanoi – stacks  
    | * Adventure text game – tables |
    | * Pong |
| 10 | Data abstraction: Classes of objects.  
    | * Snake |

This mapping was done based on the “Creative Programming Matrix” (Figure 4) developed to support the teacher in the planning of the classes.
This matrix maps the syllabus (rows) with game tutorials (columns). For each tutorial, we define the name of the game, the number of the tutorial and the week to deploy it. For each topic of the syllabus, we identify which level on the Bloom taxonomy the student should accomplish. In the cell intersecting each topic with a specific tutorial, we define the level to achieve in this tutorial. So, at the end we have a progression on the teaching of each topic and for each tutorial a specification on the learning objectives. To aid in this iterative task, three other columns calculate, the number of tutorials addressing the topic (N), the average of the level the tutorials achieve (Avg) and the maximum level on the bloom taxonomy reached (Max). This last one should be the same as the previously defined (Bloom). The levels of the Bloom taxonomy used are:

1. Knowledge
2. Comprehension
3. Application
4. Analysis
5. Synthesis
6. Evaluation

### RESULTS

Before developing this solution, there was a first study on a simplified gamification approach that was implemented on an Introductory Programming course in Informatics Engineering. This study provided the right motivation to improve the gamification design. The methodology proposed in this article was also implemented in an Introductory Programming course in the Master in Multimedia of the University of Porto.

#### 4.1 Preliminary study

This study involved 36 students enrolled in an Introductory Programming course in Informatics Engineering and 6 faculty from the Informatics Engineering Department that teach programming courses.

##### 4.1.1 Faculty

The faculty replied voluntarily to a 6 questions questionnaire with a Likert scale (1 to 5). The results are presented in Figure 5. From this data, we can find that students are generally motivated and there is sufficient documentation, but they do not study regularly. Theory classes tend to have low participation of students, contrary to theoretical-practical classes (that are mandatory). Not all faculty provide the students with an option to differentiate their learning path.
4.1.2 Students

The students replied voluntarily to a 7 questions online questionnaire with a Likert scale (1 to 5). The results are presented in Figure 6. From this data, we can confirm that students understood well both the scoring system and the levels. The questionnaire also shows that it is not unfair to limit the final level to those students with higher grades on the practical examinations and that evaluation should provide distinct opportunities to different students. Overall, they observed that the gamification project helped them to work weekly, one of the problems revealed by the faculty.

Figure 5 - Faculty questionnaire

Figure 6 - Students questionnaire
4.2 Evaluation of the results on the 2018/19 edition of the course

In the edition of 2018/19, this project was implemented in an introductory programming course in Multimedia with 20 students enrolled. The course was optative, so the students were considered motivated to learn to programme.

As seen in Figure 7, the average of the final grade (red line) has been increasing for the past 3 editions, from about 12 to 16, in a scale of 20. But with a higher rate of withdrawal (red region – top) and failing (orange area – bottom). Although very preliminary, we observe that with this methodology results kept improving, increasing the average final grade and decreasing the withdrawal and failing. Nevertheless, 3 students quitted the course. They explained that the gamification makes them realize that they did not have the time to study regularly and with the effort required in introductory programming courses. This is a good result of the gamification project, as gives the students the feeling of the workload required and their progression.

5 CONCLUSIONS AND FUTURE WORK

The skills associated with computational thinking are essential for the Digital Media professional. Particularly in Multimedia, the development of skills for the development of interactive applications is fundamental. This article presents a new teaching methodology that seeks to develop these competencies creatively, taking on the digital game metaphor for understanding the main programming concepts. To support the teachers in the deployment of this methodology we propose the Creative Programming Matrix, that provides a framework to create the tutorials to support the PBL approach.

Additionally, a gamification project is proposed to promote a natural progression in the learning of programming, essential for a first approach. From a set of clear objectives, the target behaviors and the profile of the students were defined, the cycles of involvement and the necessary tools were developed, not forgetting that learning should be fun...

Preliminary results show the positive effects of this methodology and more support materials will be developed to support the theory parts with game-based examples. Other courses are also being adapted to this new approach, namely in the Humanities Faculty. More studies will also be conducted to validate the methodology.

ACKNOWLEDGEMENTS

This work was partially financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project “CHIC – Cooperative Holistic view on Internet and Content” (Mobilizer no. 24498).

REFERENCES


