CREATIVITY AND GAME-BASED LEARNING

Florica Tomos¹, Olga Shabalina², Christos Malliarakis³, Oana Balan⁴, Peter Mozelius⁵

¹ University of South Wales (UNITED KINGDOM)
² Volgograd State Technical University (RUSSIAN FEDERATION)
³ University of Macedonia (GREECE)
⁴ Cardiff University (UNITED KINGDOM)
⁵ Mid Sweden University (SWEDEN)

Abstract

Creativity is an action and has its roots embedded within the human beings. It follows that creativity is also a human capability. Furthermore, teachers are required to use creativity and teaching methods in a creative manner. This paper will approach teacher creativity and student creativity in learning programming through game-based learning. Game-based learning itself involves high order skills such as: problem solving, critical thinking and creativity. Computer programming is a very difficult subject and students face challenges in learning programming. There are a few strategies used in programming courses in Higher Education: learning to program by playing games, learning programming through game development and learning to program by the development of games for learning programming. This study conducted a comparative analysis. Further, data was collected by course evaluation and discussions with students and teachers. Findings indicate that the three approaches are beneficial for learning programming and have also a beneficial impact on student academic performance and increase student motivation. The authors recommended a combination of these three approaches with a traditional design for students enrolled on programming course. Finally, the authors designed a Model of learning for students enrolled on programming courses.

Keywords: Creativity, Programming Education, Game-Based Learning, Motivation.

1 INTRODUCTION

Students in Higher Education find difficult to learn programming and therefore, teachers and Higher Education Institutions (HEI) need to find viable solutions to ensure not only student learning, but their motivation and performance. In this struggle to find optimal solutions, teachers and students have to use creativity within the programming courses. Consequently, this study will have a dual perspective on both students and teachers and their approach to creativity in teaching and learning. The final aim of the study is to design the best Model of Learning for students enrolled on programming courses. This research was conducted at three universities in Stockholm, Volgograd and Macedonia and researchers used Mixed Method Research in order to ensure the validity of the results. The study comprises the following sections: the first section contains an introduction with a summary of the literature on creativity, game-based learning approaches, student motivation and academic performance. The following section undertook the analysis on three different approaches to game-based learning in different country context, but within the same programming course. Section three explains the findings of the research by adopting a comparative approach and section four, which is the last section, comprises the conclusions of the study, and designs a Conceptual Model of Learning for students on Programming course.

1.1 The Phenomenon of Creativity and the Theory of Experiential Learning

This study builds upon previous research in the field of programming education [1]. The previous research argued that, within the process of learning to program, educators should consider a three dimensional perspective: teacher, students and the process of creativity itself [1]. In this study, the authors argue that creativity is the foundation of teaching and learning, being embedded as a process, within teachers’ approach to delivery of knowledge and students’ approaches to learning. This section is a summary of the literature on creativity, with different perspectives and arguments, which either supports or contradicts the perspective of this study.

Some authors [2] linked creativity to learning, while others authors [3, 4] related creativity to knowledge. Creativity was defined as a ‘trait’ including a multitude of competences [5]. However,
other authors [6] defined it as a competence or a capability, required to achieve success. These latest authors also stated that creativity is essential in educational process and can be a social process, when supported by technology [6]. In other words, creativity is no more an individual but a collective process. Moreover, it seems that students can be creative when adhering to pairs and working within groups. A study [7], exploring creativity within the social media environment, in 3D architecture, investigated the nature of creativity and found that creativity has cultural and social aspects within it [8]. Creativity, therefore, is an intricate phenomenon [1], which works well and develops better when knowledge is shared within communities and groups [3, 4, 9]. Nevertheless, creativity is defined as a process [5], and implies an action. This affirmation is based on Kolb [10] experiential learning theory. Kolb [10] argued that creativity emerged from the experiential learning process. In other words, the process of experiential learning leads to creativity and allows creativity to develop. As a result, the experiential learning theory is considered the theoretical basis of the argument within the research. This view [10] is also supported by research on optimal learning patterns and creativity emergence in a field [11].

This research argues that creativity is a process, emerging from the experiential learning and is the foundation of both teachers’ approach to delivery of instructions and knowledge and students’ approach to learning. The study approached learning to program by means of game-based learning. The investigation was conducted across three countries: Russia, Sweden and Greece, at University of Volgograd, Sweden University and University of Macedonia. The researchers explored three different approaches to learning to program by game-based learning: a) learning to program by playing games; b) learning programming through game development; c) learning to program by the development of games for learning programming. The findings, urged the authors to combine the results of the three investigations in a novel and imaginative way and to design a Conceptual Model of Learning to Program for the students enrolled on Programming courses across the three countries, with the potentiality of being enhanced and applied globally.

1.2 Game-Based Learning, Motivation and Academic Performance

The review of the literature demonstrated that, there is not sufficient research on learning to program through game-based learning [1]. This study builds on research by Shabalina et al. [10] as well as additional research literature and brings a novel and authentic Conceptual Model for Learning to Program. It was demonstrated that game-based learning is a learning technique efficient for learning to program [1] and that, it is a student-centred learning approach [12]. Although there are various types of games, this research will only consider educational games provided by teachers, and games designed by the students for the main purpose of learning, experientially, to program [13].

Although, all three approaches to game-based learning are beneficial as being student-centred learning approaches, it seems that the game development approach, or in other words, digital game authorship (DGA), is more efficient, from two perspectives: (1) the impact that have upon student deep learning and program understanding, and (2) the level of high order, intellectual skills (problem solving) which are involved and linked to creativity. This, requires student engagement and motivation, embeds interaction and feedback [14, 12]. The benefit of game development approach, over game playing approach, is that students have the opportunity to control and build knowledge by experience, gradually, than acquiring knowledge from the educator [12]. This is followed by an increase in student motivation and performance. Authors [15] argued about technology usage, student engagement, self-directed learning and academic achievement. Although, their study did not confirm the presence of a direct and unique link between technology use and academic performance, however, there is an indication regarding a relationship between technology and performance. Moreover, there is a relationship with the individual learning process. Additionally, Ayob et al. [5] provided us with tests measuring creativity and informed us about the experiential learning that supports high order intellectual skills of creativity: ability to elaborate, originality, combination, synthesis, internal visualization and focus, just to mention some of them [16].

Burguillo [17], based on game theory [18] and competition-based learning, confirmed, that learning techniques, motivated students and possible increased their academic performance. This suggests that, the hope of achieving a good grade at the end of the programming course, for instance, could be considered as motivational, rewarding and leading to student emphasis on their academic performance. It was stated that, an active and competitive environment with hands-on learning, generate creativity and can lead to performance [2]. Finally, more research comes to strengthen these opinions, confirming that constructing educational games increases student motivation and
enhances learning [19]. It is presumed therefore, that, there is a relationship between game-based learning (technology), motivation and performance, within the context of creativity and experiential learning. This assumption was investigated by this research and is the basis of the design of a Conceptual Model of Learning to Program.

2 METHODOLOGY

There is an increase in the popularity and use of Mixed Methods Research and this research decided to use a combination of research methods such as teacher’s evaluation and observations, and discussions with students, in order to enhance the validity of the research and strengthen the results of this study [20]. In the following sub-sections, the authors will introduce information regarding game-based learning in learning programming and computer games as design objects in teaching game development process. Next, the authors will approach each strategy used in programming courses, with its own impact upon student motivation and performance, in a particular country context (Greece, Sweden and Russia) but within a similar programming course in Higher Education Institutions.

2.1 Game-Based Learning in Learning Programming

Game-based approach achieves high learning results in subject domains that are difficult to study and where gaining skills is of importance [24]. To a great extent it can be applied to programming issues. The game-based approach in teaching different subject domains implies the use of specially designed games (so called educational games) as learning tools. In terms of teaching programming issues it means games designed for teaching programming technologies, languages and tools. The idea of games for learning programming technologies is based on program code visualization that helps learners to understand the idea of programming paradigms. These games are realized either as frameworks or integrated development environments (IDE) or their combination. Frameworks provide rich possibilities for program paradigm visualization. IDEs in turn are very good for learning programming language syntax and semantics, and also introduce the possibility of learning the IDE efficiently. Existing games for teaching program development are realized mostly as simulators, supporting one or several program development life cycle phases: project planning, systems analysis, requirements definition, systems design and implementation.

But in addition to this, the game development process itself can be also used both for teaching program development and training programming skills. Using computer games as design objects in teaching the software development process has also several important advantages. The purpose of the software as a game is naturally understood by the majority of the students; many students are familiar with this field and can form adequate requirements for this kind of programming systems. They are motivated to work on a game as they are interested to see the result. Game development also trains team work skills, the setting of project management priorities, planning and conflict resolution. On the other hand computer games are complex programming systems. Game design needs knowledge of system requirements analysis, software design methodologies, tools and IDEs and programming skills. Game deployment also needs methods of interactive application design, lexical and syntax analysis, searching algorithms, cross-platform programming and 2D and 3D graphics. However, in the case of using games as design objects it is very important to find those people who would agree to be potential customers of these games, who would be able to form game requirements and provide game testing.

2.2 Learning Programming through Playing Games

The limitations in learning programming can be successfully addressed by technologies that can incorporate attractive scenarios to stimulate students’ interest, include engaging tasks with clear goals that can increase students’ motivation to participate as well as provide an appealing graphical environment where students can be immersed and experience the learning process to its maximum potential. All these features can be supported by a new type of technologies that has emerged over the years, namely educational games [26], i.e. games that are developed specifically to combine education and entertainment. Studies show that educational games are used by any gender, ethnicity and socioeconomic level, and they also seem to foster skills and learning outcomes such as becoming more active and engaged [25], improve their focused attention as well as their ability to observe and identify objects in simple and complex systems and become more creative [34]. Furthermore, educational games support players to become more receptive to experiences [35], and increase their determination [34]. Such games have been developed for many educational
domains and usually incorporate a number of interesting features, such as attractive graphical interface, an engaging scenario, tasks/quests, interaction activities, characters etc. Nonetheless, there is a need for educational games to support additional features that can further underpin learning and teaching, such as multiple scenarios, different levels of difficulty in tasks to increase interest, hints during quests when needed and messages with explanatory information on possible mistakes during activities to enable scaffolding, monitoring capabilities for teachers etc.

CMX was designed and developed to enhance learning and teaching of computer programming and to be used in the classroom as a supporting tool [27]. It aims to increase students’ participation so that they are able to practice more with the concepts they are taught, without however replacing the teacher’s role as the tutor. The main environment of CMX (Fig. 1) replicates a toxic factory which pollutes the ecosystem with toxic waste, putting in danger the last remaining land of the world. In this alternate reality, a team of individuals called crackers are activists that are trying to invade the factory and shut down its main server so that it stops polluting the environment. However, the factory is equipped with employees named hackers, who are paid to protect the server and the factory’s ongoing operation. A virus has infected the main server, and has made the server vulnerable to attacks. Thus, the crackers are seizing this opportunity to find the passwords hidden inside the factory and reach the server to enter them and shut it down, while hackers are trying to find the passwords in order to destroy the virus [28].

![Figure 1 CMX environment](image)

It was also designed an evaluation framework that will assist teachers in evaluating all the features of a serious game [30, 32]. More specifically, the evaluation framework that has been designed includes 6 different axes. The first axis regards the interest of students in computer programming and their specific knowledge and skills. The second axis regards the players’ performance, while the third axis assesses the game’s performance. The fourth axis aims to evaluate the game’s difficulty and game-student interactions; the fifth axis will evaluate the game’s entertaining and motivating elements while the last axis will evaluate CMX as an educational game for computer programming. It should be noted that the evaluation framework was previously used during the first utilization of the game in university students [31]. The following table (Table 1) shows a mapping of each concept of the design framework to an axis of the evaluation framework [29].
Table 1. Mapping CMX design framework with CMX evaluation framework

<table>
<thead>
<tr>
<th>Design Framework concepts</th>
<th>Evaluation framework axes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure</td>
<td>Game’s performance</td>
</tr>
<tr>
<td>User</td>
<td>Interest of students in computer programming and their specific knowledge and skills</td>
</tr>
<tr>
<td>Learning objectives</td>
<td>Game’s difficulty and game-student interactions</td>
</tr>
<tr>
<td>Pedagogy</td>
<td>Educational elements of the game</td>
</tr>
<tr>
<td>Scenario and activities</td>
<td>Entertaining and motivating elements of the game</td>
</tr>
<tr>
<td>Learning outcomes</td>
<td>Players’ performance</td>
</tr>
</tbody>
</table>

The evaluation results showed that the majority of the students increased their knowledge on computer programming by using the game. More specifically, students that scored a grade of 5 to 7 in a computer programming course managed to correctly answer multiple choice questions regarding the theory taught as well as correctly place multiple tiles of code in the correct order to make up an executable C code and write their own programs. Additionally, students with scores equal to or higher than 8, completed all the game’s tasks, and thus confirming that the game’s activities successfully support computer programming education. It was also examined the students’ intention to use an educational game such as CMX in the future for computer programming education. The evaluation results concluded that the participants enjoyed the game’s entertaining elements and were motivated to progress within the scenario’s steps and did not face major difficulties during the learning process [33]. This positive experience is also confirmed by the fact that the majority responded that, they would indeed re-use CMX when learning programming in the future.

2.3 Learning Programming through Game Development

Two courses with rather different student groups have been analysed by course evaluations, observations and discussions mainly the course batches during 2015 – 2016, but outcomes from earlier course versions have also been considered. All courses were given at a department of computer and systems sciences.

2.3.1 Web programming

This is a 7.5 ECTS five week course for students on the first year of two Bachelor’s programmes on Interaction design and IT and market communication. Students mostly come from non-technical backgrounds and are in general relatively new to programming and with a majority of students between 20-25 years old. The syllabus should cover object-orientation in the Python programming language as well as an introduction to web programming. Most game construction assignment should be solved with HTML5, CSS3, JavaScript and/or jQuery but there are also assignments with game-construction with Python and Tkinter.

Typical games are variations of classic game ideas such as Memory, Hangman or Mastermind, but there are every year some submissions of new unique game ideas. There are also multimedia extended versions of simpler game ideas like Stone, Paper and Scissors, or various kinds of quiz constructions. It is important to require a certain degree of uniqueness in every game to stimulate creativity, but also to prevent plagiarism. Course batches have consisted of 100 – 150 students with a completion rate around 70% which is relatively high for a programming course. Grades were given in the A-E scale with six optional assignments for a quantitative accumulation for the higher grades. In the two last course batches around one third of the students chose the game construction assignments and often in combination with the assignments that use multimedia techniques in HTML5 Canvas and JavaScript/jQuery. Fewer students chose the more complex game assignment that should be constructed with Python and Tkinter.
2.3.2 Multimedia programming in Python

Course participants have had a large geographical spread and also with variations in the age with the youngest students below 20 years old and the oldest so far 65 years old. The course was given as a voluntary summer course running on a 50% study pace from early June to late August. Course batches have the last five years consisted of groups between 60 – 400 students with the possibility that all activities can be completed in distance mode from all over the Swedish nation.

The entire course was built around the idea of that multimedia design and game construction can stimulate intrinsic motivation and creativity. Online lectures, lessons and the course book have an emphasis on analysis and synthesis of digital games. Almost all assignments were also given with game construction as a framing theme. In the final project assignment a multimedia game should be designed and implemented, where students in the first course batches were completely free to create whatever game idea they found engaging. Later, the instructions have changed, and students had to design and implement a learning game, which leads us forward to the next section.

2.4 Learning Programming through the Development of Games for Learning Programming

Speaking about creativity, people think first about games. It is generally accepted that games (at least the good games) empower creativity and a game player must be innovative in order to be effective [21]. Games require the players to make decisions, issuing them with the opportunity to impact the story and even to some degree to outline the world in which the game is played out. Game players should be free in selecting distinctive game actions; the game should give a tone of options, should contain components of arbitrariness, and while game actions chosen by the player should emphatically impact on the game process.

Games for learning programming can also not only exploit students for gaining knowledge and mastering new skills, but also provide an environment for dynamic, critical, often collaborative and creative learning, allowing students to investigate strategies, ideas and aptitudes quickly and securely inside an environment designed with particular learning modules [22]. Development of computer games in teaching the program development process has also several important advantages in sense of teaching creativity. Game developers have to create an original concept of the game (to decide what the game is going to be, how it looks, how it plays, etc.), to create dialogues, game characters, they have to work in more diverse teams that require more creativity and interpersonal communication skills than traditional software developers [23].

But even more challenging is a combination of the two approaches, when students, while studying program development process, develop not just regular games, but games for learning programming [24]. A key problem in the development of any educational game is the problem of finding a way of keeping the balance of play and learning components, ensuring both the game attractiveness and the opportunity to achieve its learning goals. Thus, the students developing a game for learning programming have to analyze firstly a learning course that is supposed to be implemented in the game and to find a way of its integration to a game context. After that, they have to create all the rest game components. Thus, teaching programming through the development of games for learning programming, students have to be “twice more creative” in comparison to the development of common digital games.

3 FINDINGS

3.1 Web programming

The game construction assignment together with the described multimedia assignment have been the most popular of the non-mandatory add-on assignments. More complex game construction with Python and Tkinter have not attracted more than 5-10% of the course batches. In particular, students from the Interaction design programme have chosen these more open-ended assignments. Another mandatory, but very popular assignment has been the design and development of an ePortfolio in HTML5, which also has been a catalyst for creative design. Most students find the more closed-ended assignments on object-orientation and web scraping difficult and boring even if they agree on their usefulness and alignment to real life development in the industry. Several students from the IT and market communication programme have chosen the web scraping assignments since they seem more relevant for their further studies. Motivated students from both programs, also, find the object-
orientation important, but, not as stimulating and fun as multimedia and game construction. In this course it has been a priority to stimulate creativity with workshops and challenging assignments, but this must not only be by game construction and voluntary open-ended activities. The assignment with many creative and surprising solutions is ePortfolio where students from both programmes show their personal interpretation of innovative web design at the same time as they learn HTML5. Game construction should be seen as one of several ways to kick-start creativity in programming courses.

3.2 Multimedia programming in Python

The combination of multimedia programming and game construction has worked great for most participants for this diverse group. During six years it is not more than two participants that have been reluctant to the idea of building digital games. Within the initial course design, students were completely free to choose whatever game idea they fancied, as there was a stereotypical submission. There are too many replicas of classic game ideas, and also plagiarism issues, which rather is the opposite of creativity. To increase variation and originality, instructions were updated with some given game themes for the construction of a learning game. With the large variations in pre-knowledge and gaming experience, there have also been large variations in the quality of the project assignments. For the higher course grades originality and creativity in design are required in the grading criteria together with more technical programming aspects. Marking programming assignments is not always great fun for a programming teacher but in this course it has mostly been a pleasure.

3.3 CMX Design & Game Construction

Although, the development of games for learning programming, proved to be twice creative, than the normal digital games, and even though, multimedia and game construction was highly stimulating, the authors of this study, have reached the opinion that a combination of these three strategies will more effective for student learning, motivation and creativity development, within programming courses across three universities. It is suggested that, the experiential learning, through playing games, game development and authorship of games within programming courses, needs to be regarded as a holistic opportunity for students, providing alternatives, flexibility and increase in motivation, which finally is the foundation of creativity and leading also to student creativity.

4 CONCLUSIONS

The authors conclude with the idea that a combination of these three teaching and learning strategies of game-based learning will enhance student motivation and will also ensure student creativity. Moreover, creativity is a two dimensional process, which connect together teacher creativity with student creativity. It is teacher creativity, which nurtures student creativity by supplying creative educational games to learn experientially programming. The results indicate the role of the teacher and his/her creativity in mentoring and guiding students through the game development process. Also, such findings emphasize students’ creativity and motivation in developing and constructing games to learn programming. The conclusion supports research by Kolb [10] and Shabalina et al. [1]. The engagement in multiple learning to program strategies, by game-based learning, reinforces students’ learning and ensures deep understanding of the programming, where creativity is a complex phenomenon and plays such an important role for both student and teacher. There are implications for both students and teachers, as well as for the educational practitioners in programming and policy makers. Further research can be conducted with surveys and discussions, teachers’ observations and evaluations across universities and countries. Following this research and based on the literature review and the findings of the research, the authors constructed a suggestive Conceptual Model of Learning to Program, which is presented below in Fig. 2:
REFERENCES


