CMX: The Effects of an Educational MMORPG on Learning and Teaching Computer Programming

Christos Malliarakis, Maya Satratzemi, and Stelios Xinogalos

Abstract—Computer programming has for decades posed several difficulties for students of all educational levels. A number of teaching approaches have been proposed over the years but none seems to fulfil the needs of students nowadays. Students use computers mainly for playing games and the Internet and as quite a few researchers state this aspect of computers should be taken into account in the way we educate them. Towards this direction, this paper aims to examine the effects of using an educational Massive Multiplayer Online Role Playing Game (MMORPG) on teaching and learning computer programming. The educational features of an MMORPG called CMX are presented along with a design framework that was devised taking into account previous work in designing educational games. The effects of CMX on teaching and learning computer programming are assessed through a study with first-year undergraduate students. Seventy six students used CMX over a period of five weeks for learning various procedural programming concepts. Students evaluated various aspects of CMX by filling in a questionnaire that was based on an evaluation framework, which was devised in accordance with the design framework of CMX. Moreover, the results of a midterm exam that took place prior to using CMX and students' accomplishments in the context of CMX were recorded and analyzed. The results show that the majority of the students was entertained by playing the game while learning, and felt motivated to continue based on the game's scenario due to the variety of activities included. In regards to the students' performance, a pre-test and a post-test were carried out in the experimental group, i.e., the participants of this study, and the control group, i.e., students of the course that continued to get taught the same concepts and performed the same assignments as the experimental group, but traditionally. The pre-test and post-test analysis of the performance results for both groups showed that the majority of the students in the experimental group increased their performance in computer programming. Furthermore, students stated they had a positive attitude in regards to re-using CMX in the future in order to learn additional programming concepts. The positive results of this study pave the way for CMX being used in the classroom and expanding the game's functionalities that will further increase students' performance and support teachers in delivering the required knowledge. Moreover, the work reported in this paper offers game designers and teachers methodological and empirical results for game-based learning in such a difficult domain as is computer programming. What is more, the design and evaluation frameworks presented are general enough that they can be easily adjusted and/or extended for designing and assessing educational games in other domains as well.

Index Terms—Educational game, MMORPG, computer programming, design framework, evaluation framework

1 Introduction

The field of computer programming education has been studied for numerous years and research indicates that significant difficulties and challenges are faced by teachers and students respectively [1], [2]. These challenges regard the complexity of the domain, which includes complicated knowledge with abstract concepts and the demanding development of practical cognitive skills, such as logical thinking, problem solving and high level abstraction [3]. These issues hinder computer programming education, as students usually characterize the courses as boring and difficult to understand. Furthermore, students consider the teaching methods used as uninteresting and state that they are not provided with enough diverse exercises in order to practice the concepts they learn [3].

To this end, researchers have endeavoured to identify and apply solutions that could alleviate these challenges and enhance computer programming education. A representative encouraging suggestion involves the utilization of educational games in the classroom, which are gradually being accepted as a promising educational tool [4] because it is evidenced that they increase students' performance [5] and motivation [6]. 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 [7]. Additionally, they improve students' focus of attention, including their ability to observe and identify objects in both simple and complex systems [8] and become more creative [9]. Furthermore, educational games support students in becoming more receptive to experiences [10], as well as increasing their scores [11], and determination [10]. 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.

• *The authors are with the Department of Applied Informatics, University of Macedonia, 156 Egnatia Street, Thessaloniki GR-546 36, Greece. E-mail: {malliarakis, maya, stelios}@uom.gr.*

Manuscript received 5 July 2015; revised 29 Mar. 2016; accepted 11 Apr. 2016. Date of publication 25 Apr. 2016; date of current version 16 June 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TLT.2016.2556666

Particularly, multiplayer online games can provide multiple benefits when utilized in education [12]; for example, Massive Multiplayer Online Role Playing Games (MMORPGs) continue to be popular as they reinforce motivation and creativity and allow skill development, such as problem solving and communication [13]. The incorporation of MMORPGs in education is encouraged by related studies, since they can foster learning by enabling students to practice, think critically and plan strategically [14].

The main contribution of the work reported in this paper is that it offers game designers and teachers a methodology and substantiated results for game based learning in such a difficult domain as is computer programming, through the development of an educational MMORPG. The paper presents the design framework that was constructed which led to the development of the MMORPG named CMX, and the evaluation framework constructed to conduct the assessment of CMX's effectiveness in teaching computer programming. The study was conducted on first year Computer Science university students and the evaluation process was based on six research questions that aimed to examine all features of the game and how these affect students' learning experience in computer programming. More specifically, the features examined include: the game's performance, its entertaining and motivating elements, the educational aspects of the game, and how students assess their interactions within the game. Finally, the evaluation investigates whether the game's use in computer programming courses can indeed increase students' performance.

The remainder of the paper is organized as follows: Section 2 presents related work on educational games developed to support computer programming; Section 3 describes a framework that guides one in designing an educational game and includes a brief overview of CMX; Section 4 describes CMX's evaluation framework and the study methodology; Section 5 presents the data collection and the statistical analysis carried out based on the results and all the corresponding findings; finally, Section 6 provides the discussion and conclusions as well as recommendations for future work that will increase the game's efficiency in supporting computer programming education.

2 Background

Educational games are a novel learning method that has been gradually used for teaching various educational domains. Educational games are considered any serious games that primarily aim to teach instead of entertain and that support "learning by doing" education [7].

The basic idea behind the development of educational games is the exploitation of existing successful outcomes, such as state of the art computer games towards the creation of innovative and attractive learning experiences. The main strength of educational games is that they can support the educational needs of the "digital natives", i.e., individuals who have been accustomed to using computer games and browsing the Internet from a very early age.

Educational games are popular across all genders, ethnicities and socioeconomic lines. Furthermore, their usage seems to be linked with the development of positive skills such as attention and visual-spatial abilities [15], [16], performance in the classroom [17], [23], creativity, willingness to new activities and environments [19], motivation for success and engagement [22].

Various research studies have been carried out on games whose features support the teaching and learning of computer programming. The majority of these games include a particular scenario that aims to cover a specific computer programming unit, whereas some games - fewer in number - cover multiple learning objectives and theory units.

The educational game Catacombs [15] belongs in the first category whose aim is to familiarise students with variables definition and usage of if-statements and loops. The player becomes a wizard and answers questions in order to fill out a code correctly, utilizing a specifically designed mini-language. The evaluation of the game in real world educational settings showed that the students found the experience both entertaining and beneficial, while they found the game to be interesting and attractive. Specifically, two evaluations of the game were conducted with 13 and 8 random students respectively, all of whom had basic knowledge in Computer Science and were enrolled in the corresponding Department of the University of North Carolina (UNC).

Saving Sera [16] is an educational game where the player tries to save a princess named Sera, who has been abducted by the monster Gargamel on her 16th birthday. The players are required to undergo various quests, either by filling out proper code units in specific exercises or by sorting code segments in their proper place so that they can progress to the next step. These activities enable students to learn and practice on recursion and if-statements. Saving Sera was evaluated by university students, where the evaluation process included the recording of a video as well as an interview after the game. The results indicated that students found Saving Sera entertaining and educational, but would have preferred additional scaffolding. As with Catacombs, the game was evaluated in two rounds, with 13 students and 8 students respectively, randomly chosen with varying levels of computer science knowledge. The students were mainly from the University of North Carolina from the Computer Science Department in Charlotte.

Elemental: The Recurrence [17] is an educational game for teaching computer programming. Initially, a brief demonstration of recursion introduced students to the theory and afterwards students were required to apply the depth first search algorithm by moving the protagonist across a binary tree and complete three missions. It is a three-dimensional game that helps students learn recursion and the depth first algorithm by utilizing the C# programming language, and it was developed using the game engine DarkWynter. Forty-three university students from the Department of Computer Science at UNC, who were either taking or had completed the "Data Structures and Algorithms" course, took part in the application of Elemental. The evaluation showed that the majority of students wanted to learn even more complex programming concepts through the use of a computer game.

Prog&Play [18] is a web-based strategy game that enables interactions amongst users. It is multiplayer and has been developed using the open source strategy game Kernel Panic. When used in education, students program their characters and try to form alliances in order to win. Students can select the language they want to program with from amongst choices such as Ada, C, Java, OCaml, Scratch and Compalgo.

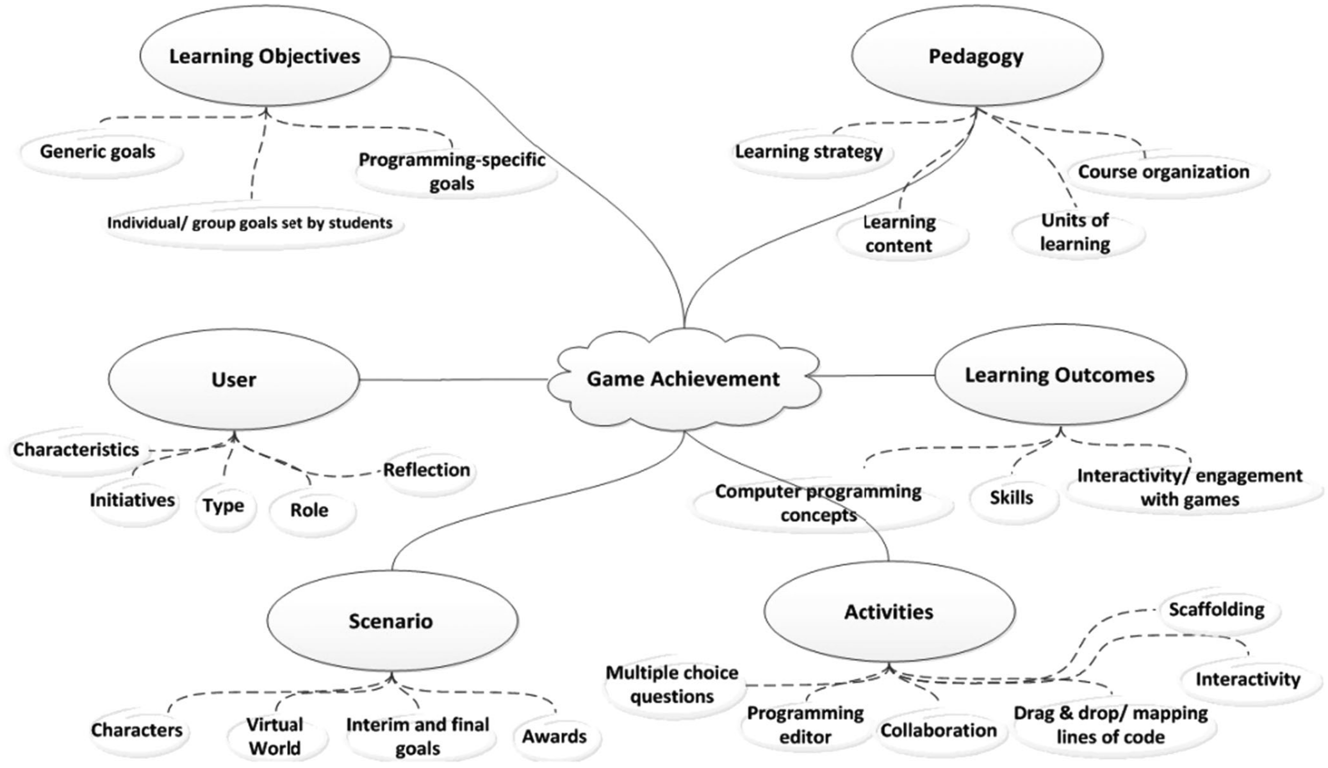

Fig. 1. CMX design framework.

Fifteen (15) novice computer science students from the Institute of Technology at the Paul Sabatier University Toulouse III in France played Prog&Play. The initial results showed positive feedback regarding the game's entertaining elements and its usefulness as an educational tool.

Robocode is an educational game that aims to cover multiple learning objectives. Robocode [19] is a two-dimensional environment that aims to teach the Java programming language and includes a virtual arena and a programming editor. With Robocode, students try to program a robot that will fight another robot in a virtual arena. Through this process, students learn the basic commands of procedural computer programming and object-oriented programming, such as inheritance and polymorphism. Robocode was evaluated by 83 participants that were randomly drawn on from the online forum of the Robocode community. During this process, they were asked to write programming code that would solve a specific problem, which involved creating an army of robots that would have to beat their opponents. Students seemed to enjoy playing the game and the overall evaluation results were positive.

All the above educational games present interesting solutions that can increase programming performance. However, they focus on specific units of learning and support limited programming activities. In view of this an educational game was designed and developed that includes multiple functionalities and diverse activities that support both students and teachers, enhance all the different phases of the game, and increase the motivation to learn.

3 The Design Framework and CMX Overview

CMX was designed and developed to enhance the learning and teaching of computer programming and to be used in the classroom as a supporting tool [20]. CMX belongs to the MMORPG genre as it includes several common features such as persistent game environment, some form of progression, social interaction within the game, in-game culture, system architecture, membership in a group, and character customization [21]. More specifically, in CMX as in nearly all MMORPGs, the development of the player's character is a primary goal, players have tools to facilitate communication between them, it is based in a client-system architecture and CMX provides different types of choices, activities and interactions that players can choose. It aims to increase students' participation so that they are able to have more practice in the concepts they are taught, but does not diminish the teacher's role as the tutor. The sub sections below describe the framework constructed for the design of CMX and elaborate on the game's scenario, the different roles (player/student and teacher/administrator/supervisor), and the features incorporated.

3.1 The Design Framework

The CMX design framework was constructed after studying the existing frameworks [22], [23], [24], [25], [26], [27] and the features currently supported by educational games in the teaching of computer programming. The framework includes concepts that should be taken into consideration in the design of educational games on computer programming [28]. It is abstract enough to be employed by future designers and developers and detailed enough to act as a solid guide without any ambiguity, as is shown in Fig. 1.

The most prominent concepts that define the game's design are:

1. User. This concept is essential for educational game designers as it involves the system's end users, i.e.,

the students. To this end, designers should take into consideration the students' profiles, their specific characteristics and interests (including the type of user, and their role inside the game), their initiatives, i.e. the types of decisions and choices they can make within the game and their knowledge on the domain to be taught, so that they can create features that allow teachers to define how the game will assist the users (reflection, initiatives).

2. Learning Objectives. This refers to the game's specific learning goals. This includes what teachers want the students to learn when playing the game and how this objective can be assessed in order to determine whether the set goals have been achieved. These goals could be a) generic, such as "the students should become more collaborative", b) programming-specific, such as "the students should learn how to program using the if-statements", or even c) individual or set by the students, such as "I want to gain more points/spells to win the game". It is also important to emphasize that learners that are more in contact with the learning content during learning are also more likely to remember the information taught. Hence, serious games should include learning content in all parts of the game and provide features such as interaction and feedback that will engage learners with the content in a continuous manner [41].
3. Pedagogy. The educational element of the game is another important feature that should be taken into consideration by game designers. This includes the specific learning strategy to be used during class, as well as the learning materials to be taught, the units of learning per class, learning content etc. Thus, the game should support features that will allow teachers to organize the educational material (course organization) and units of learning in different ways based on the course's specific characteristics and set educational objectives [29]. Furthermore, one of the most common and effective ways to incorporate constructivism in educational games is including scaffolding features. According to the constructivist theory, educational technologies should provide maximum scaffolding to students, so that they will be supported throughout their interaction with the tools [42]. That way, students can feel safer getting familiarized with a new environment with multiple learning contents and a variety of activities. However, the more knowledge students receive and absorb, the more the scaffolding features should gradually decrease [42].
4. The scenario and activities are essential components that can significantly determine if the educational game is successful or not. These two elements establish the game's plot, as well as the overall navigation pathways that the students can take. They also are fundamental in linking the game's entertaining aspect with its educational aspect. To this end, elements such as scaffolding, game-student interactions and collaboration should be supported based on this concept, including any other features that will entertain and motivate students (e.g., awards, specialized characters, virtual worlds, interim and final goals, points etc.). It is also essential that the activities designed will be educational, comprehensive, usable and motivational so that the student will be engaged enough to want to proceed to the next one [30]. These activities can include multiple-choice questions, writing lines of code using a programming editor, dragging & dropping lines of code, as well as others that foster collaboration and interactivity.
5. Learning outcomes usually represent the students' final performance, in regards to what they managed to learn, how well and how this helps them in their future learning experiences. In CMX, this also includes the student's satisfaction levels towards using the educational game for learning computer programming and their willingness to play in the future in order to learn (interactivity/ engagement with the games). Students' performance is also connected with the support they are provided with inside the game in order to achieve maximum level of understanding of the taught concepts. This support can be achieved in educational games through providing mechanisms of detecting students' movements across the environment, so that everyone can have a full overview of the course of the game at any time. This also allows any students to ask or provide help to players in need. Other types of support include specific hints when students fail to progress after multiple efforts, explanatory messages in case mistakes are made, as well as collaboration tools (e.g., chat, forums etc.) that allow easy communication between the players. This way, educational games become more personalized and adaptable, reducing the possibility that students will be displeased by their gaming experience and providing security within the environment.
6. Game achievement. This comprises all that the student has accomplished during the game, including the learning outcomes (e.g., the theory that was learnt and the exercises successfully completed), the game achievements (e.g., points and spells gathered), the player's attitude of accomplishment, and being positively inclined to play again in order to improve their performance [31].

Infrastructure is also an essential component of game design. This refers to the architecture and technical requirements that need to be taken into consideration in order to ensure proper game performance. This way, the system will operate without errors, bugs and latencies [32].

3.2 The CMX Scenario

The main environment of CMX replicates a factory which pollutes the ecosystem with toxic waste, and the last remaining piece of earth is also in danger of contamination. In this alternate reality, a team of individuals called crackers are activists who attempt to invade the factory in order to shut down its main server and thus stop it polluting the environment further. However, the factory is equipped with employees named hackers, who are paid to protect the server and the plant's on-going operation. A virus has infected the main server making it vulnerable to attacks.

The crackers seize this opportunity to discover the hidden passwords that will enable them to reach the server, enter the passwords, and shut it down. On the other hand, the hackers are also trying to find the passwords so as to destroy the virus.

Students are initially divided into one of the two teams, crackers and hackers. The players enter the game, read the game's guidelines and start to interact with each other, e.g., beating the game's guards or players of the opponent team while trying to pass each game's stage. It should be noted that in the game players are taught the programming language C. Initially, all players of both teams are trained individually by special characters existing inside the world, which are called Senseis. This training process is essential, as students can learn the theory regarding programming concepts, as well as practice with the help of the Senseis the material they have been taught. In this way, they can accomplish the various missions and proceed to the next levels, where each level provides them with one of the required passwords that bring them closer to the factory's main server. More specifically, there are three different levels that the students need to pass, each represented by a different type of Sensei (Sensei, Iron Sensei and Gold Sensei). Every Sensei provides students with a password that if they execute the requested tasks successfully, unlocks a specific Sensei of the next level. The students initially try and locate the first level of Senseis within the graphical environment. These Senseis first provide the students with the theory in units of learning that students can study and absorb. Once students declare within the game that they have studied the theory and want to proceed, the Senseis provide them with a series of multiple-choice and right/wrong questions that correspond to the theory they just studied. Students can also request to study the theory multiple times in case they realize they have not understood it properly. The Senseis give feedback to the students in regards to their answers, and once students answer the questions correctly, they are given a password that will allow them to proceed to the next level and meet the Iron Senseis.

As soon as students navigate through the world and locate an Iron Sensei, they must go near the character in order to interact with it. Then, they are requested to enter the correct password, i.e., the password they got from the Senseis in the training process. Once they have entered the passwords, the Iron Senseis can identify that each player has successfully passed the first level of training, which comprises the authentication process. Then, the students are required to study a series of tiles of codes that together form an executable C program. They must then construct that code by dragging & dropping and placing the tiles in the correct order. The system examines the code submitted and provides corresponding feedback to each student. If the code is wrong, the system presents an error message and allows the student to try again. If the code is correct, the student is given a new set of passwords which he/she has to provide to the Gold Senseis for authentication.

During the training process, students are required to write actual programs using commands of the C language on a programming editor provided within their interactions with the Gold Senseis. More specifically, a problem in the form of text is provided to the students, which they have to interpret into an executable program and write the appropriate code. The code is then compiled by an embedded compiler and the result is either a 'correct' or 'error' message. If a mistake has been made, an explanatory message appears which enables the student to fix it. Correct codes give the students the final password that provides access to the main server. It should be noted that CMX can be easily customized in order to support another programming language, as all that is needed for this customization is the installation of the corresponding compiler.

During training, students can communicate with each other by using the message chat tool that is incorporated in the game and help each other when they have difficulty comprehending concepts or executing a task. The students that manage to find the final password and unlock the server are the winners. The game and its final goals can easily be customized based on the educational goals the educator wants to set. This also includes the setting of the parameters that determine which players are determined as winners and how. For example, the teacher can determine if the winners will be the students that reach the final stage by finding the server's password, or the students that reach the server at specific time instances, or if all the students in a team must first answer all or specific questions correctly etc.

Furthermore, a mini map tool is embedded that displays a micrographic outlook of the environment so that players can have an overview of all players' positions in the game. This feature enhances students' collaboration since they can form alliances based on their respective positions in the game world and endeavour to progress by helping each other.

Fig. 2 shows the environment in which the students try to locate and interact with all the Senseis or the other players from the same team during the training process of the game in order to be properly trained and in which they endeavour to reach the main server to submit the final password that will destroy it. Students can protect themselves and beat their opponents by using and winning more weapons, spells and experience points as they fight their way towards the main server.

3.3 The CMX Roles and Features

CMX includes and can be further expanded with a large variety of computer programming concepts without having to focus on only one learning unit. This is one of the supplementary features included during the design process, aiming to support multiple capabilities for teachers and students. Another feature is the ability for the teachers to manage the game's environment, monitor students' behaviours during the gameplay and assess their progress. Furthermore, teachers are allowed to determine each of the Sensei's behaviour regarding how they will guide and engage the students. This way they can simulate their own guiding and scaffolding role into each Sensei character. For this purpose, they can include a variety of tasks, as well as advice, explanatory messages and hints that can assist students when they make a mistake or miscomprehend a concept.

Teachers can also configure the game's graphical environment, as shown in Fig. 3. This functionality presents multiple benefits, as it allows the entire game to have a different look for the various classes and students depending on their specific particularities. This fosters and stimulates

Fig. 2. CMX environment.

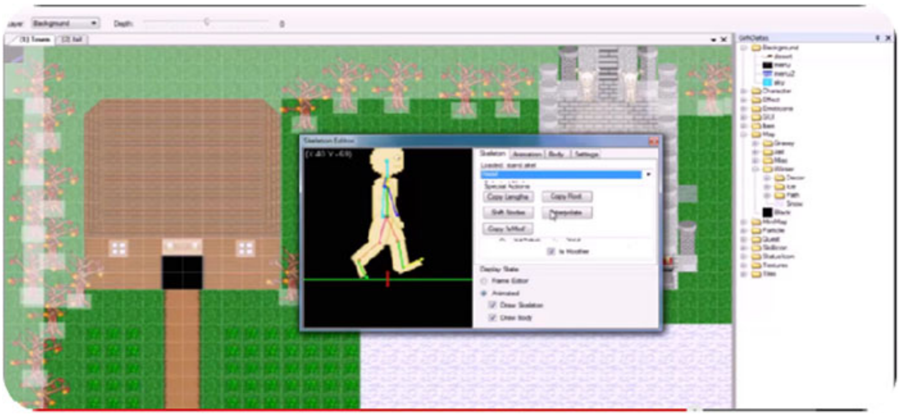

Fig. 3. Configuration of the CMX environment.

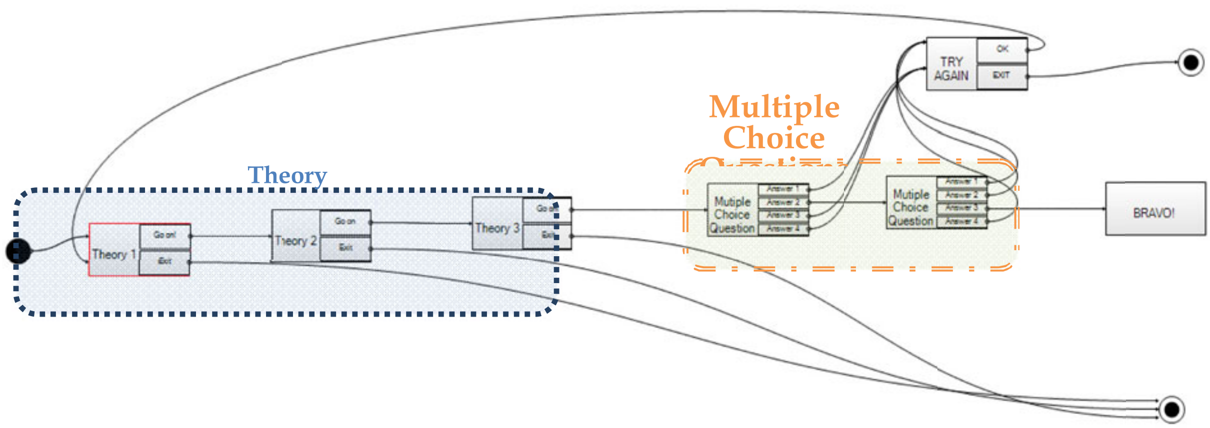

Fig. 4. CMX dialogue editor.

students' interest and motivation, since the game is redesigned according to their preferences. Fig. 3 presents a representative screenshot of the CMX editor environment.

Teachers also configure the CMX environment by formulating the educational content to be taught in each course section. The teacher uses the game's dialogue editor in order to design the theory units, the practical activities to be added for each theory unit, as well as the sequence flow that will guide students to maximum knowledge acquisition. Fig. 4 is a screenshot of the dialogue editor that demonstrates each instance of the virtual dialogue carried out between the student and the Sensei as a graphical state with multiple possible answers that determine the progression of the dialogue in different states, as shown by the arrows. This way, each Sensei leads the student from the theory statements to the evaluation states (i.e., multiple choice questions), and based on the student's answers, the teacher provides the corresponding feedback with

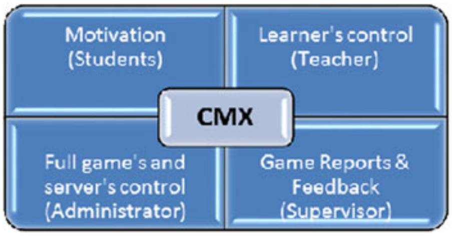

Fig. 5. User Roles in CMX.

congratulatory messages or with hints that should be taken into consideration during the next attempt. Thus, each dialogue with the Senseis is represented by a fully customizable state diagram.

Based on the above, CMX was designed and developed with the intent to draw attention to and engage all roles that are involved in an educational game. To this end, it combines features that support all these roles, as shown in Fig. 5.

More specifically, CMX was developed mainly to support students in learning computer programming, by fostering their motivation and allowing them to be more active while participating in the learning process. This is facilitated with the inclusion of a number of features such as activities/tasks during the training phase and entertaining elements they can use while they are trying to reach the server (e.g., points, spells, rewards, special weapons, points etc.). Another important feature is the provision of feedback in regards to their progress, the possible mistakes they make during training, their current location or status etc. It should be noted that although these entertaining elements, such as spells and weapons are not directly related to the learning outcomes, they can be indirectly connected to the educational aspect of the game and the learning outcomes, since they act as motivational catalysts that will help students to proceed within the game. Examples include a Sensei rewarding the student with a spell for a correct answer, or a giving them a hint to locate a hidden spell within the virtual world. Types of spells can be healing abilities, energy boosts, fast movements etc.

Additionally, the game also supports teachers in maintaining complete control over what learning content is taught and what kind of activities the students will engage in during training. The game also allows for the role of administrator (usually but not necessarily the teacher) to configure the entire environment by changing the characters and the graphics, to monitor students' behaviour and their progress, check the server's operation and adapt the game's difficulty levels when necessary. A final role is that of the supervisor, who can be either a teacher or the administrator, and is able to construct reports that can provide insights on the game's status, the overall progress of the participants and other analytics.

Another important concept is computer programming knowledge and skills. This concept is very closely connected to the game as it represents what students will learn by playing. This concept can significantly change the entire game environment since it affects the number and types of tasks to be included, the units of learning to be inserted, the resources to be made available by the system's server etc. To this end, it is important that the design and development of any educational game takes into consideration the particularities of the knowledge to be taught and the skills to be developed by using the game.

Fig. 6 shows a list of the features that comprise an educational game for computer programming that supports all phases of education: design, implementation and evaluation. Specifically, CMX combines entertainment with education: while playing, students are able to perform activities, get feedback on their progress and mistakes, collaborate with their classmates and assess their knowledge comprehension.

Also, teachers/administrators are able to customize the entire game, monitor the learning experience, provide scaffolding to students, and include content and activities that will allow the necessary knowledge to be comprehended and skills to be developed. Finally, teachers and supervisors are able to assess all students' activities and receive analytics feedback and evaluation reports from within the game.

4 The Evaluation Framework, the Methodology, and the Research Questions

This section elaborates on the framework constructed for the analytical evaluation of CMX in regards to its effectiveness in increasing student performance in computer programming. In addition, the methodology is presented.

4.1 The Evaluation Framework

Taking into consideration the design framework, an evaluation framework was constructed that assists teachers in

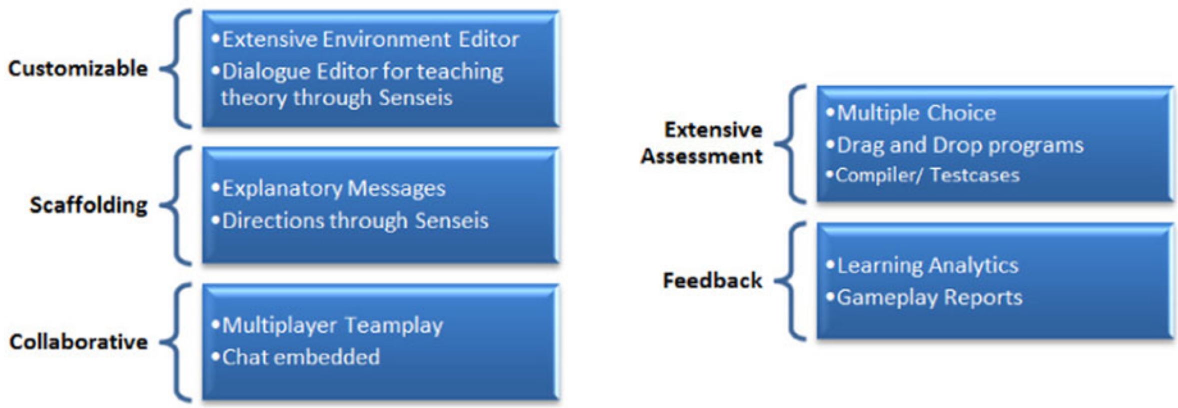

Fig. 6. CMX features.

assessing all the features of the educational game [33]. More specifically, the framework includes six axes.

1. Student knowledge in programming and interest in games
2. The game's performance
3. The game's entertaining and motivating elements
4. Game-student interactions and the game's level of difficulty
5. The game's educational aspects
6. Student performance

In this paper, we are applying a revised version of the CMX evaluation framework [33].

4.1.1 Student Knowledge in Programming and Interest in Games

The student's profile represents user characteristics such as knowledge, behaviour, and goals [34]. The CMX design framework has shown that the end user, i.e., the student, should be taken into particular consideration, since the learning objectives and outcomes are directly related to the student's particularities. Student profiles were studied during evaluation of the game by focusing on the interest the student has in games, their specific knowledge, and their skills in the field of computer programming. Overall, this can show whether existing knowledge of a domain or interest in participating in the learning process affects a student's performance in the game.

4.1.2 The Game's Performance

The performance of CMX during gameplay is a vital factor for its success in education. Proper system function with no delays or bugs can lead to high levels of user satisfaction and immersion in the environment [35]. To prevent system delays, it is essential to make sure that there are ample resources, such as bandwidth, CPU capacity, RAM etc. One way to ensure players' satisfaction is to sustain optimal server performance, which is especially important in MMORPGs that involve large numbers of users and real-time interactions. Performance can be hampered due to network traffic caused by latencies of the Internet, causing 'lags' which alter the graphical environment, thus disrupting users' gameplay [32].

The present study researches ways in which resource consumption can be improved appropriately so as to guarantee game performance. This led to the design of a mathematical model [37] that estimates the different factors that affect the server load and disrupt the proper operation of the game. When this model is applied, the server releases only the resources needed for the game and vice versa, and in so doing prevents both overload and network traffic. This allows the students to continue to be immersed in the game, and transmissions of feedback to the teacher (e.g., learning analytics data and results) are carried out successfully [36]. The assessment process involves students being asked about their experience of the game as regards its performance. In other words, questions regarding players' satisfaction of movement, interactions and decision-making, as well as message transmission can provide insight that can be used to enhance the game's performance.

4.1.3 The Game's Entertaining and Motivating Elements

One of the most significant advantages of educational games is the ability to motivate and engage students in the learning process. This can be achieved by making the elements of the environment interesting, attractive and entertaining. It is essential that the games capture students' attention from the start and that the students continue to be engaged in the game irrespective of the length of time that involves [36]. One way to do this is by providing features that attract students' attention and draw them into the game [31], such as a stimulating plot that has a wide range of activities at varying levels [38], which although challenge the student, do not discourage them from continuing the game. The activities should kindle students' interest and motivation by being at an appropriate level and in no way should overload their cognitive and memory capacities [39].

It is, therefore, essential to investigate students' perceptions of CMX regarding its entertaining features and motivational value. More specifically, the research examines to what degree students were entertained by specific aspects of the game, which were included in order to increase their engagement and satisfaction. Examples of these are: the plot, the graphics, the learning activities and all other interactions with the game's elements and players.

4.1.4 Game-Student Interactions and the Game's Difficulty

Students are able to interact with the game and within the game throughout their entire learning experience. Such interactions can include: the tasks they execute, their navigations across the environment, their chat sessions, their usage of the mini map tool that allows them to locate other players and characters and then navigate to different parts of the world and their attempts to shut down the factory's server. All these features are strongly interconnected with the learning objectives set by the teachers at the beginning of the course.

The game's difficulty is associated to students' attitude towards their interactions with the system. This is highly dependent on the one hand, on the specific skills that they need in order to accomplish an activity, and on the other, on the challenges they encounter while trying to achieve this [38]. For example, if an activity is too difficult for the skills-ability of the student, then the usual reaction is anxiety, if on the other hand, the activity is too easy and the student can complete it quickly, then the most common reaction is boredom [38]. Thus, it is essential that an educational game has activities at appropriate levels of challenge for the students. Challenge generates motivation as well as providing entertainment, and thus helps students improve their performance [39]. For this reason, students need to feel a high sense of satisfaction with the game's activities. This can be achieved by having students: engage in and accomplish tasks with an appropriate degree of difficulty [39], confront challenging opponents [42], achieve specific goals [41], develop skills while engaging in situations that have the right degree of danger and intensity [42]. Students will feel rewarded on accomplishing these challenges and will want to play again in the future [39].

Another important aspect regarding difficulty is being able to navigate one's character across the environment without encountering problems and to be able to use the available objects in the world as these were intended, which prevents disruption and thus enables students to accomplish the learning objectives [40]. In addition, players should be able to explore the game's world at their own pace [38], [41] and in so doing have a more satisfying experience within the environment due to the fact that they can control it themselves.

According to the research, it is investigated to what degree students were satisfied with the various interactions and whether they found them difficult or (un)-interesting. This would help in that if the majority of students, for example, considered a specific task particularly difficult, then that could lead to a lack of motivation to carry out the interaction or even to continue playing the game. CMX's success as an educational game is based on its user-friendly interactions. The research also examines to what extent students found the interaction with the various game elements easy or difficult, whether the game provided appropriate feedback, and whether or not there was consistency in the choices, the situations and the actions that can be taken. Finally, students were asked whether there were elements within the game that supported collaboration.

4.1.5 The Game's Educational Aspects

The elements that differentiate the game from its traditional counterparts are yet another important aspect of the game's evaluation. Since the game's educational content alone does not effectively lead to better learning or teaching [30] there need to be other features that enhance its educational content, such as, the integration of scaffolding or appropriate feedback, the ability for players to develop skills and gain knowledge, and the provision of resources that assist students while they play [29].

The game's efficiency to teach computer programming is also investigated. It is a key factor to ascertain whether the theory in combination with the exercises given, had a positive impact on students' comprehension level. Additionally, it is researched whether students found the game both enjoyable and educational whether they would like to be taught concepts with this game, and if they found the learning process easier with this game.

4.1.6 Students' Performance

The key value of CMX's efficiency is measured by its ability to increase students' performance. When the players are totally absorbed in the environment and plot of the game, solving the given problems, then performance can be enhanced [43]. If a student becomes a better programmer while playing CMX, then the game has achieved its objective. It is also of great importance that players feel a sense of control in regards to manipulating their character [35] and being able to move freely across the game's world [43]. Finally, in order for players to be stimulated and maintain their interest in the game, they should be provided with multiple navigation pathways and different ways of winning [35], [41]. To determine how students' performance can be improved, it is important that their evaluation feedback regarding both the entertainment and educational aspects of the game is studied, along with the log data, which records students' performance in the game step-by-step. These records are then compared to students' pre- and post-tests.

4.2 Methodology of the Study

CMX was evaluated in the context of the first year compulsory course "Procedural Programming" taught at the Applied Informatics Department at the University of Macedonia. An announcement was made to all students just after the mid-term exam for participants to use CMX in the course labs. The 76 students that agreed to use CMX formed the experimental group, while the remainder of the students (234) whose preference was to use a typical IDE formed the control group. No bonus or credit was offered to the students in the experimental group. Our decision to let the students freely choose between CMX and an IDE was influenced and justified by our departmental principle of equal and fair treatment for every student. Certainly, both the students and the teachers were aware of the fact that his or her choice of system could affect his or her final grade either negatively or positively. Additionally, what needs to be underlined is that CMX had already been implemented on a small group of students in a previous pilot study leading to very positive results [33]. Therefore, it stood to reason that if we tried it under the actual teaching conditions of a relevant course and for a clearly longer time period, the results would be more interesting, than those of the pilot study. Both groups learnt the same concepts and performed similar activities but in different ways. Specifically, the experimental group used the CMX as a tool of study (e.g., study the theory, answer multiple choice questions) and practice (e.g., construct, write and execute programs), while the control group studied using the relevant books and teacher's slides and practiced writing programs using an IDE (Integrated Development Environment). At the end, both groups of students had to take the same final exam. The 76 students had to evaluate CMX through a questionnaire that was based on the evaluation framework. In order to evaluate the framework's sixth axis that refers to student performance, a pre-test and post-test were used, which comprised the grades of the midterm and final exams respectively.

For the experimental group five different versions of the game were designed, and each version covered a different computer programming learning unit. The first was an introductory unit and taught students the sequence structure and the basic commands of the C programming language. The second learning unit dealt with the selection structure, the third was on loops, the fourth taught students about functions, and the fifth taught students about arrays.

The educational material in each game was supplemented with ten activities that students were required to execute during their interactions with the three levels of Senseis in each game version. It should be noted that the game's assignments were designed in accordance to the assignments given to the students that were participating in the course in the traditional way inside the classroom.

More specifically, students initially interacted with four Senseis, where the educational material included:

- Theory for each learning unit and multiple-choice questions that evaluated the theory comprehension levels
- Multiple choice questions regarding the values of variables after a program's execution, and
- Validation questions regarding the correct syntax of programs.
- Students also interacted with four Iron Senseis through drag & drop activities. During these interactions, students were required to put the lines of code provided in the right sequence that together form an executable C program. Such programs can be either simple or advanced calculations. For example:
  - Calculating the sum of odd numbers of a given sequence of numbers.
  - Calculating the absolute value of a number.
  - Converting months to days and days to hours.
  - Creating more complex programs such as finding prime numbers
- Finally, students interacted with two Gold Senseis through writing lines of code. During these interactions, students were required to construct programs such as:
  - Entry of variable values in a program focused on each learning unit.
  - Calculation of simple functions, such as finding the maximum or the minimum value, calculating the sum of given elements, and
  - Implementation of an entire program, such as the bubble sort algorithm in a one-dimensional array or search algorithms.
  - Implementation of more complex programs with specific data and programming of multiple functions, where methods and variables need to be implemented in combination.

The next step of the evaluation process involved the accumulation and analysis of the results in order to answer the research questions presented in the following section.

4.3 Research Questions

The main research questions, presented below, represent each of the six axes of the evaluation framework.

RQ1: Was CMX's game performance satisfactory? Do students' positive views on the game's performance affect their performance while playing CMX?

RQ2: Was CMX entertaining and motivating? Do students' positive views on the game's entertaining and motivating elements affect their performance while playing CMX?

RQ3: Was game-student interactions and the game too difficult? Do students' positive views on the game-student interactions and the game's difficulty affect their performance while playing CMX?

RQ4: Were CMX's educational elements effective? Do students' positive views on educational elements affect their performance while playing CMX?

RQ5: Does adopting CMX improve students' performance?

RQ6: Do previous gameplay experience and interest in computer programming affect students' performance in playing CMX?

5 Data Collection and Analysis

A set of factors are examined that are directly related to the evaluation framework described in section 4. An online questionnaire was created to assess the factors. The questions were further grouped into the six axes of the evaluation framework and the corresponding research questions.

The first four questions of the questionnaire refer to students' progress in the training phase, while the remainder of the questions use a 5-point Likert scale (with 1 being the most negative attitude and 5 the most positive). In each of the four research questions, we also examine whether there is a relation of students' view on the researched game's elements and their performance during playing the game. This will provide a broader and more objective overview of the results, since correlation between performance and positive evaluation of the game would indicate that the inclusion of the game in the classroom is a positive learning experience.

The results were also combined with log data recorded in the game's system, as well as any informal discussions carried out in the classroom. Such log data include:

- Numbers of interactions with each Sensei level
- Number of successful and un-successful endeavours to retrieve each password within the game
- Player profile and status at multiple time stamps (e.g., weapons, experience etc held, position in the map etc)
- Number, types and time stamps of bugs or malfunctions caused by the game's system.

The above were studied and compared to the answers provided by the students in the online questionnaire to examine the results' validity. Both findings were combined to bring forth the following results in regards to CMX's efficiency as an educational tool for computer programming. We also correlate every RQ with the high performance of the players within the game. In our study, we define students' performance as "high" when they correctly write both programs indicated by the Gold Senseis and thus find the final password that shuts down the server, and we define students' performance as "good" when they manage to complete one of the Gold Sensei programs.

5.1 RQ1: Evaluating the Game's Performance

The game's performance during students' learning experience was evaluated to identify any issues that need to be addressed in a future version of the game to ensure proper operation of the server and the system. All the questions test whether students encountered any delays or problems during their time within the game and are available in Fig. 7. Students' replies in the questionnaire showed that they did not encounter any delays or problems during learning and playing in CMX (Q.2.1-Q.2.4). Moreover, according to the log data, no malfunction in the game was detected that could harm or hinder its efficiency or negatively affect the students' performance and learning experience during the game.

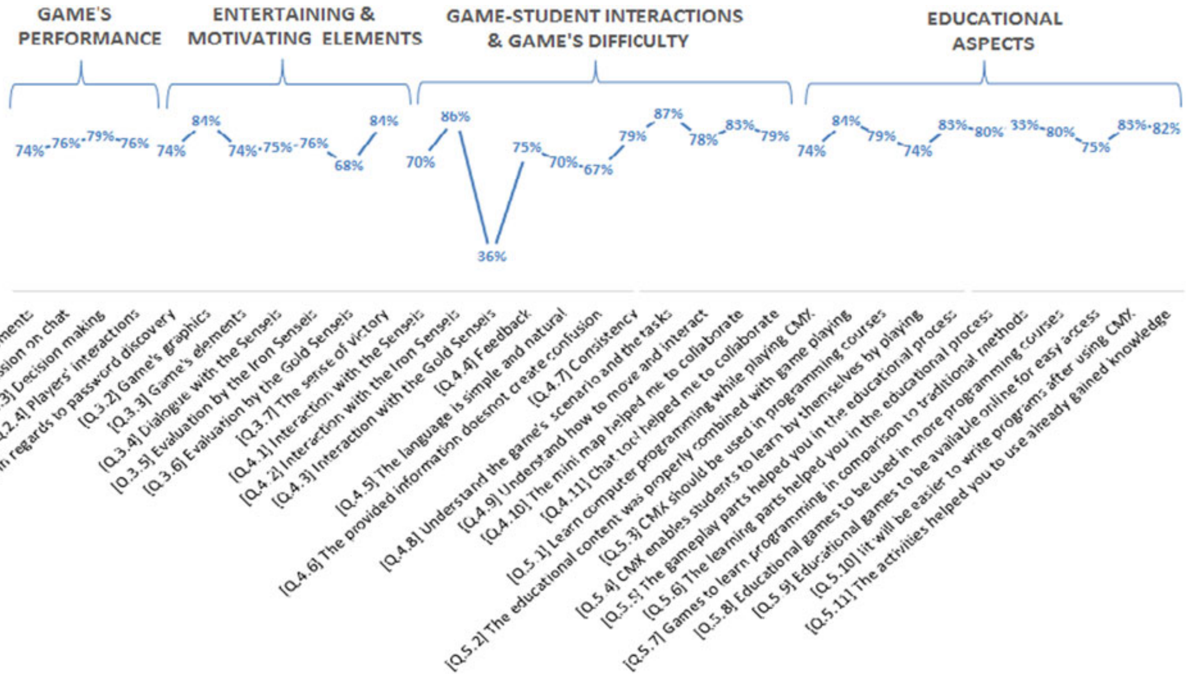

Fig. 7. Students' overall attitude towards CMX.

Next, we researched whether the ratings of the game's performance axis are related to the students' performance within the game, aiming to determine if the game's performance influenced players' performance. The statistical analysis indicated that there is a significant correlation between students who had a good performance in CMX and stating that the system's response during decision making was very satisfactory ($r = 0.207$, $p = 0.037$). A significant correlation was also found between students' high performance and the students that found the system's response during their interactions with the game's elements as very satisfactory ($r = 0.206$, $p = 0.37$).

5.2 RQ2: Evaluating CMX's Entertaining and Motivating Elements

The evaluation of an educational game's entertaining aspect is threefold as it:

a) supports features that are similar to those computer games users are already familiar with,
b) entertains students during learning with numerous characters, obstacles and challenges, and
c) motivates students to complete their tasks.

As can be seen from the findings of the questionnaire, regarding students being entertained by the game's plot for password recovery (Q.3.1), and their enjoyment of the game's various elements (e.g., weapons, health rating, inventory, spells etc) (Q.3.3), the percentages were 73.7 percent (Q.3.1: Mean = 4.01, Std. Dev. = 0.966) and 73.6 percent (Q.3.3: Mean = 4.00, Std. Dev. = 0.801) respectively, while a large 84.2 percent stated that they enjoyed the game's graphics (Q.3.2: Mean = 4.22, Std. Dev. = 0.741).

It is also investigated to what extent students were entertained by interacting with the different levels of Sensei and whether they found the corresponding activities engaging and motivating. More specifically, 75 percent claimed that they were entertained and motivated by their dialogues with the Senseis (Q.3.4: Mean = 3.96, Std. Dev. = 0.871), and 76.3 percent stated that they enjoyed using drag & drop tools to correctly order pieces of a programming code (Q.3.5: Mean = 4.17, Std. Dev. = 1.051). A significant number (68.4 percent) said that they enjoyed and were motivated by being able to write code which can be automatically assessed via the test cases (Q.3.6: Mean = 3.82, Std. Dev. = 1.029). The vast majority (84.2 percent) stated that the sense of purpose (finding the hidden passwords) and of victory (destroying the server) increased their motivation to better learn the programming theory so that they could achieve their goals (Q.3.7: Mean = 4.28, Std. Dev. = 0.759). Finally, we compared the questionnaire answers with the log data recorded in the system, and the results confirmed our initial findings, which indicate that 84.2 percent of the students interacted with all Senseis while 76.3 percent used all the game's entertaining elements. Such elements include the chat tool where they could communicate and collaborate with the other members of their team, the usage of weapons and potions to defeat their opponents, the overall navigation and exploration of the virtual world, the usage of virtual money to buy and sell artifacts etc. The overall impression of the students' answers for this axis is positive regarding the interactions and activities of the game.

It is also investigated if students' performance was affected by their positive views on the game's entertaining and motivating elements. The statistical analysis indicates that there is a significant correlation between students who had a good performance in CMX and stated that they enjoyed the game's plot in regards to password discovery ($r = 0.726$, $p = 0.001$). Moreover, there was a significant correlation between students' high performance and the students that enjoyed the game's graphics ($r = 0.211$, $p = 0.034$); the students who had a positive opinion regarding the dialogues with the Senseis and the multiple choice questions ($r = 0.231$, $p = 0.022$); and the students who enjoyed ordering lines of code with drag & drop commands during their being trained by the Iron Senseis (r = 0.650, p = 0.001).

5.3 RQ3: Evaluating Game-Student Interactions and Game Difficulty

The players' interactions with the Senseis were explored with a focus on the level of difficulty. Furthermore, the technical aspect of CMX was investigated, by gathering input regarding four important system usability features, such as feedback, language, design and consistency.

According to the results of the questionnaire, a high percentage of students (69.7 percent) found that the Senseis were 'easy' or 'very easy' to interact with and to accomplish the challenge (Q.4.1: Mean = 3.93, Std. Dev. = 1.011). This positive response increases significantly to 85.6 in relation to the Iron Senseis (Q.4.2: Mean = 4.24, Std. Dev. = 0.728). In contrast, only 36.3 percent of the participants stated that they found their interactions with the Gold Senseis 'easy' or 'very easy' (Q.4.3: Mean = 4.01, Std. Dev. = 0.931).

From the findings, it can be seen that 75 percent of students were satisfied with the feedback provided during the game, such as information on each player's current location in the environment or their status (e.g., experience, points, weapons etc.) (Q.4.4: Mean = 4.00, Std. Dev. = 0.894). Also, a large number of students (69.7 percent) agreed that the language used was fairly simple and natural and they did not have any difficulty in navigating across the different stages and menus of the game (Q.4.5: Mean = 3.93, Std. Dev. = 1.181). It is worth noting that a significant 67.1 percent found that the volume of information provided does not create confusion (Q.4.6: Mean = 3.80, Std. Dev. = 1.033), while a majority of 79 percent stated that the choices, situations and actions were expressed/executed in the same way across the entire programming environment (Q.4.7: Mean = 4.09, Std. Dev. = 0.996). Finally, 86.8 percent found that they could easily understand the scenario and the tasks (Q.4.8: Mean = 4.26, Std. Dev. = 0.957), 87.8 percent could easily understand how to move and interact with other players (Q.4.9: Mean = 3.96, Std. Dev. = 1.012), while 82.9 and 79 percent stated that the mini map feature (Q.4.10: Mean = 4.24, Std. Dev. = 0.864) and the chat tool (Q.4.11: Mean = 4.12, Std. Dev. = 0.864) helped them to collaborate with other players.

According to the investigated log data, no malfunction was detected that could hinder the game's efficiency. More specifically, we compared the execution times of each activity and detected correlation between the activities' execution time and the overall performance of the students in the game, while no extreme values were noticed or a large variation of execution time values. This shows engagement of all students in all stages of the game and constant interaction with the game's elements, with approximately no idle times amongst the playing time, which indicates interest and motivation of the users to continue playing and learning

Furthermore, it was investigated whether students' performance was affected by their positive views on the interactions with the game and the game's difficulty. The statistical analysis indicates that there is a significant correlation between students getting a good performance in CMX and stating that they enjoyed their interactions with the game's elements. There was a similar correlation to students who found the game to "provide constant and full information on what is occurring" (r = 0.228, p = 0.024), and the language used in the game (e.g., the dialogue, messages, menus etc) was simple and natural (r = 0.658, p = 0.001). Finally, high performance also correlated with a positive view on the game's consistency, i.e., that the same choices, situations and actions are expressed/executed in the same way throughout the entire programming environment (r = 0.203, p = 0.040).

5.4 RQ4: Evaluating CMX's Educational Aspects

Another important axis that was evaluated concerns the educational aspects of CMX. The participants were asked in fourteen questions to assess CMX in regards to its efficiency in teaching computer programming.

The results demonstrate that the vast majority of students were very positively inclined towards all CMX elements which allowed them to learn programming by playing (Q.5.1: Mean = 4.05, Std. Dev. = 1.031) with 73.7 percent. A high 84.2 percent of students had a positive attitude towards educational content and game playing (Q.5.2: Mean = 4.17, Std. Dev. = 0.915), while 78.9 percent stated that CMX should be incorporated in the computer programming courses (Q.5.3: Mean = 4.21, Std. Dev. = 0.805), and if this is done then 73.6 percent believed that it could support self-learning (Q.5.4: Mean = 4.00, Std. Dev. = 0.800). In addition, the majority of students claimed that their educational performance could be enhanced with CMX gameplay components, such as chat tool, etc. (Q.5.5: Mean = 4.32, Std. Dev. = 0.820), and CMX learning parts/components, such as theory, hints, explanatory messages, etc. (Q.5.6: Mean = 4.08, Std. Dev. = 0.876) with 82.9 and 80.3 percent, respectively.

It is worth noting that 82.9 percent of students stated that they preferred learning computer programming through the use of games rather than the traditional learning methods applied in the classroom (Q.5.7: Mean = 4.17, Std. Dev. = 0.929), and 80.2 percent would like educational games to be included in future computer programming courses (Q.5.8: Mean = 4.16, Std. Dev. = 0.767). A significant 82.9 percent believed that after playing CMX, they would find it easier to write programs (Q.5.10: Mean = 4.09, Std. Dev. = 1.085), while 81.6 percent stated that the activities on CMX helped them to use the knowledge already gained to accomplish the required goals (Q.5.11: Mean = 4.13, Std. Dev. = 0.914). Finally 75 percent of the students expressed their desire for CMX and other similar games to be available online for easy access (Q.5.9: Mean = 4.07, Std. Dev. = 0.822).

Furthermore, it was investigated whether students' performance was affected by their positive views on CMX's educational elements. The statistical analysis indicates that there is a significant correlation between students who had a good performance in CMX and stated that they managed to learn computer programming while playing the game (r = 0.204, p = 0.038). Furthermore, a correlation was also found between students who had a good performance and stated that CMX enabled them to learn by themselves by playing (r = 0.215, p = 0.031). Finally, students who had a good performance also stated that the gameplay parts of CMX helped them in the educational process (r = 0.247, p = 0.016).

TABLE 1
Comparison Table of Midterm Exam Grades and Gold Senseis Passed (N = 76)

No. of Gold Sensei	Grade in midterm test of "Procedural Programming" course						
	<5	5	6	7	8	9	10
0	40.6%	33.6%	14.8%	12.3%	0%	0%	0%
1	34.4%	38.5%	51.1%	53.3%	20%	0%	0%
2	25.0%	27.9%	34.1%	35.3%	80%	100%	100%
Total	100%	100%	100%	100%	100%	100%	100%

TABLE 2
Comparison Table of Midterm Exam Grades and Final Exam Grades

Factor	Group	Count	Mean	Standard Deviation
Midterm examination	With CMX	76	4.53	3.019
	Without CMX	234	3.36	3.400
Final examination	With CMX	76	4.87	2.995
	Without CMX	234	2.33	3.191

5.5 RQ5: Evaluating Student Performance on CMX

The last axis to be assessed regards the actual performance of the students while playing the game. In order to achieve a high performance, a student is required to pass all or most of the Senseis and thus have retrieved all or most of the hidden passwords.

According to the students' answers and the log data, 63 percent of students successfully completed the first Sensei level (Q.6.1: Mean = 3.91, Std. Dev. = 0.829) while 81 percent passed at least 2 of the 4 Iron Sensei challenges (Q.6.2: Mean = 3.82, Std. Dev. = 1.029). Over half of the students (58 percent) succeeded in passing both of the Gold Senseis, managing thus to complete the entire scenario of the game.

Although a significant number of students appear to have made substantial progress in the game, with over half finishing all tasks, in order to be able to assess whether or not CMX enhanced their programming performance a comparison is made with the results in their mid-term exam in the "Procedural Programming" course, i.e., prior to their being introduced to CMX. Table 1 below presents the percentage of students and their grades (out of 10) in the midterm exam (those that got less than 5 are in one group), and the percentage of Gold Senseis passed. Only the Gold Sensei results are used in the comparison because these tasks were the most complex and thus the most demanding. Plus the tasks provided by the Gold Senseis are very similar with the assignments included in midterm and final exams for evaluating students' knowledge. Accomplishing these shows the highest performance students can achieve.

It can be seen from the table that over half (59.4 percent) of the students who had not passed the mid-term exam were able to complete one (34.4 percent) or both (25.0 percent) of the problems in the Gold Senseis and write their own code using the C programming language. For students who got a grade 5 in the exam, meaning that they just managed to pass, 38.5 percent managed to accomplish one task and 27.9 percent accomplished both tasks and correctly wrote two programs. Together this reached a total of 66.4 percent. Furthermore, a total of 85.2 percent of the students that got a grade 6, and 88.6 percent that got a grade 7 managed to write one or both programs correctly, while, 80 percent of those who got an 8 in the exam, successfully wrote both programs. As was perhaps to be expected, those who scored 9 and 10 had 100 percent success rate. On the other hand, there were negative results in that 12.3 percent of the students who got a grade 7, 14.8 percent who got 6, and 33.6 percent who got 5 did not manage to solve correctly any of the programs provided by the Gold Senseis. This confirms previous studies which showed that gamification does not have a positive impact on all students [44]. Taking a closer look at these 33.6 percent of students with a grade 5 the findings show that 85.2 percent answered at least two of the Iron Senseis assignments correctly and 95.1 percent successfully completed all of the first level Senseis. This shows that even though these students did not pass the Gold Senseis level, their behavior within the game did not differ to the other students who managed to do so. They used all of the entertaining elements of the game, the chat tool, spells, weapons etc and they interacted with the other team members and their opponents in their efforts to win.

Following this, the student grades (out of 10) from the midterm and final exams were compared.

As it can be seen from Table 2 the students that played CMX had a mean of 4.53 grade in their midterm examination, while the students that did not play CMX had a midterm exam mean grade of 3.36. Correspondingly, after the CMX implementation in the experimental group, these students improved their mean to 4.87, while the control group reduced their mean score to 2.33. These results provide an overview of the performance improvement of the experimental group compared to the control group. Nevertheless, we proceeded to compare the grade deviation for each student and correlated it with whether they played CMX or not.

Table 3 shows that the average grade deviation seems to be affected by whether the students played the CMX game or not. More specifically, the average grade deviation shows a statistical difference and the usage of CMX positively affects the experimental group.

5.6 RQ6: Evaluating the Factors that Influence Students' High Performance

Students' interest in computer programming was assessed by a self-reflection question, where they had to evaluate their own level of knowledge regarding specific computer programming learning units. According to the findings, a large percentage (70-80 percent) did not consider themselves to be advanced or expert in programming concepts, while a significant percentage (40-60 percent) stated that they 'know nothing or very little' about these same concepts.

Students' interest in computer games shows whether they are already familiar with the features and environment of existing games. Based on the questionnaire results, 43.4 percent of the students stated that they play games every day or 3 to 4 days a week, whereas 56.4 percent stated that they play 1 to 2 days a week or very rarely. In relation to the number of hours students spent on computer games, the majority or 81.8 percent claimed that they played for 2 to

TABLE 3
Grade Deviation of Experimental and Control Groups

Factor	Group	Number	Mean	Standard Deviation	t-value	Significance at $p < .05$
Grade deviation	With CMX	76	0.34	1.419	2.428	0.020
	Without CMX	234	-1.03	1.820	2.429	0.021

4 hours a week. To the question on the amount of time students spent participating in computer programming courses, 68.2 percent claimed that it was '3 to 5 hours' or 'more than 5 hours' a week. These results show that the vast majority of students already spent a lot of time playing computer games and were therefore both familiar and comfortable with their features and functionalities. Furthermore, students spent a substantial amount of time studying computer programming, which indicates that there is a need for assisting mechanisms, such as new technologies.

Moreover, it was investigated whether students' interest in computer programming and previous gameplay experience influenced students' adept performance during the game. More specifically, we examine students that managed to pass to the final training phase and successfully completed at least one of the two assignments provided by the Gold Senseis. A comparison was made using a t-test and the results are shown in Table 4.

5.6.1 Frequency of Gameplay

This factor examines if there is a correlation between the number of hours per week spent playing computer games (e.g., arcade, other MMORPGs etc), and CMX goal accomplishment. Students were categorized into those that frequently play computer games and those that do not, and a t-test was performed in relation to their performance on CMX. There was no statistically significant difference between the scores of those students who are already familiar with computer games as opposed to those who do not usually play ($p = 0.126$). Therefore, frequency of gameplay is not a factor that affects students' performance in successfully completing all the tasks in CMX.

5.6.2 Interest in Computer Programming

Another factor examined is students' interest in computer programming outside the classroom. Students were asked questions on whether or not they do any activities related to computer programming because they enjoy them and not because they are course requirements. The students were grouped into two categories, i.e., those that are generally interested in computer programming and those that are not, and a t-test was performed to determine whether this factor has any correlation to students' performance. Based on the statistical results, this factor also showed no statistically significant difference between the scores of the two groups ($p = 0.244$).

6 Discussion and Conclusions

Educational games are recent tools that help increasing students' motivation and interest in learning by providing an entertaining scenario and graphical environment as well as promoting engaging activities. Learning with such a tool is even more impending in the computer programming domain, where students face particular difficulties. To this end, we have developed an educational MMORPG called CMX, which aims to teach introductory computer programming.

This study evaluated the learning effectiveness as well as the entertaining and motivating elements of CMX, an educational MMORPG targeted at computer programming within university students of the Applied Informatics department in the University of Macedonia. The study investigated how to overcome difficulties regarding computer programming education through CMX, including ways to increase students' motivation, enable students to actively participate while practising programming concepts, allow teachers to control the game's environment, as well as monitor students' progress and performance.

Fig. 7 presents an overview of students' responses for questions regarding the main axes of the evaluation framework.

More specifically, the graph shows the percentage of students who responded with a 4 or 5 point on the Likert scale, indicating a positive attitude towards the particular question and the aspect evaluated. Thus, the graph can provide insight into the general attitude of students in regards to the game and its elements that was generally positive. It is worth noting that the only question where there is a high negative response (36 percent) refers to students' interaction with the Gold Senseis (Q.4.3: Mean = 4.01, Std. Dev. = 0.931). However, this result was expected since Gold Senseis represent the highest level of difficulty within the game, where students need to analyze two problems, design the solution and implement the corresponding program. Moreover, there were two questions whose mean, although relatively higher than 3.5 is nevertheless slightly lower than the mean of the rest of the questions. More specifically, 68.4

TABLE 4
Comparison Table of Factors and Gold Senseis Passed ($N = 76$)

Factor	Group	Number	Mean	Standard Deviation	t-value	Significance at $p < .05$
Frequency of gameplay	Frequent	33	1.48	0.667	0.346	0.731
	Not Frequent	43	1.53	0.592	0.340	0.735
Interest in computer programming	Interest	35	1.60	0.604	1.127	0.263
	No Interest	41	1.44	0.634	1.132	0.261

percent of the students evaluated positively the programming editor used for writing complete programs (Q.3.6: Mean = 3.82, Std. Dev. = 1.029) and 67.1 percent evaluated positively the aesthetic and minimalistic design of the game (Q.4.6: Mean = 3.80, Std. Dev = 1.033) and more specifically the volume of information provided. Although these percentages are relatively high, they are still lower than those for the other questions, which imply that improvements should be made to the programming editor, the volume and the way this information is displayed to students so that there is no confusion.

Besides students' replies in the questionnaire the following data was collected and analysed: log data recorded by the game; students' performance in the programming tasks incorporated in the game; data from the midterm exam of the course that took place prior the utilization of the game and were treated as a pre-test; and data from the final exam that took place after the utilization of the game and were treated as a post-test. Regarding the midterm and final exams data were recorded and analysed for all students attending the course, namely the students that utilised CMX (experimental group) and the ones that did not (control group). The analysis of the aforementioned data helped us draw the following conclusions in respect to the research questions of the study.

Based on the log data, no malfunction in the game (RQ1) was detected and this was confirmed by students' replies in the questionnaire that evaluated positively the game's performance (Q.2.1-Q.2.4).

Students' majority evaluated positively the entertaining and motivating elements of the game (RQ2) that refer both to its plot and mechanics (e.g., graphics, password discovery, weapons, health rating, inventory, spells) and the learning material and tasks (e.g., Senseis, developing programs with drag and drop, test cases). Besides students' replies (Q.3.1-Q.3.7), log data also showed that 84.2 percent of the students interacted with all the Senseis and 76.3 percent of the students used all the game's entertaining elements although it was not necessary for completing successfully the game, which indicate that students were entertained and motivated to take full advantage of all game features.

Game-student interactions including the feedback provided, the language used, the design and consistency of the game, as well as the tasks represented by the Senseis did not cause difficulties to the majority of students (RQ3). As it was already mentioned, the interaction with Gold Senseis (Q.4.3) and the volume of information provided (Q.4.6) had the greatest negative impact, as the positive assessments were 36 and 67 percent respectively. Engagement of all students in all stages of the game and constant interaction with the game's elements, with approximately no idle times amongst the playing time was recorded in the log data as well.

There were positive results regarding the educational aspects of CMX (RQ4). Over 80 percent of the students found the game entertaining and were motivated to follow the scenario step-by-step without coming across any major difficulties during the learning process. Roughly the same percentage stated that they would indeed use CMX again when learning programming in the future. Considering that frequency of gameplay and interest in programming outside coursework were not factors that affected students' performance (Table 4), indicates that these results are not affected by the fact that the students chose to use this technology.

When studying the experimental group's student performance in the programming tasks incorporated in CMX in correlation with the aforementioned features/factors using a t-test, the following conclusions were drawn:

- There is a significant correlation between students who had a good performance in CMX and stated in relation to game's performance (RQ1) that: (1) the system's response during decision making was very satisfactory; (2) the system's response during their interactions with the game's elements was very satisfactory.
- There is a significant correlation between students who had a good performance in CMX and stated in relation to the game's entertaining and motivating elements (RQ2) that: (1) they enjoyed the game's plot in regards to password discovery; (2) they enjoyed the game's graphics; had a positive opinion regarding the dialogues with the Senseis and the multiple choice questions; (3) they enjoyed ordering lines of code with drag & drop commands.
- There is a significant correlation between students who had a good performance in CMX and stated in relation to their interactions and game difficulty (RQ3) that: (1) they found the game to provide constant and full information on what is occurring; (2) the language used in the game (e.g., the dialogue, messages, menus etc) was simple and natural; (3) the same choices, situations and actions are expressed/executed in the same way throughout the entire programming environment.
- There is a significant correlation between students who had a good performance in CMX and stated in relation to the games' educational aspects (RQ4) that: (1) they managed to learn computer programming while playing the game; (2) CMX enabled them to learn by themselves through playing; (3) the gameplay parts of CMX helped them in the educational process.
- The frequency of gameplay (RQ6) is not a factor that affects students' performance in successfully completing all the tasks in CMX. Other studies that researched the facilitating role of previous game experience on students' performance and goal achievement [45] also did not find any statistically significant difference.
- There is no statistically significant difference between the students that were already interested in computer programming as a field and the students that showed no particular interest (RQ6).

In conclusion, comparing the average grade differences between the pre- and post-tests for both the experimental and control groups, there was a definitive negative statistical difference for results of student performance in the control group, who had no connection with the CMX system. In contrast, there was a positive statistical difference in the results of student performance in the experimental group. This indicates that CMX had a likely impact on enhancing student knowledge in computer programming. It must be noted that although the first year programming students in the experimental group agreed to use CMX, the majority stated that they had little programming knowledge and were not particularly interested in programming activities further than their required coursework. Additionally, the study findings showed that gameplay frequency did not have an effect on student performance on completing the CMX-assigned tasks. In the study, all students—in both the experimental and control groups—spent the same amount of time working in the labs, did the same assignments, and took the same midterm and final exams. There is the possibility that since game based learning is a new educational approach which uses a technology and a learning process unfamiliar to students, they could have assessed CMX negatively or even presented worse results. There have been some literature findings which have shown this, in spite of students' initial positive attitude to its use. However, in the present study, there was no such response from the participants in their experience with CMX.

As is the case in most studies, there are potential threats to validity. In our case, the validity and generalizability of the results might be compromised by the fact that the experimental and control groups were not randomly formulated. However, the fact that some students chose to use CMX does not necessarily indicate that they were familiar with that technology as a learning tool. On the whole, that is probably a situation where students were willing to build on experience and knowledge or having specific learning styles. Based on these observations, the correlation of learning styles and CMX in the programming learning process could be the subject of further research [46]. Furthermore, as the students in the experimental group achieved higher pre-test results than those in the control, it could suggest that they were better students to begin with, who, nevertheless improved after using CMX, while their peers' performance decreased. Due to the above limitations, it is recommended that the study be repeated using a random sample and even a larger one. It would also be of interest for the study to be repeated for a number of consecutive years and a comparison made of the empirical findings.

On the whole, the study findings are encouraging towards the further exploration of game-based-learning as a learning method for computer programming education, and more specifically towards the incorporation of CMX and other MMORPGs in the classroom. To this end, CMX will continue to be extended and enhanced, since all the issues that were identified during the study will be addressed in the future version(s) of the game, as well as all the concerns that were raised by the students and the teachers in order to maximize CMX's effectiveness. Furthermore, it is planned to expand the game's material incorporating more learning units, and the game will be applied in more real world educational settings in order to further test its validity and gather more data on how its features can be improved. Future work will also include the examination and employment of the widely accepted Technology Acceptance Model [47] when CMX is being applied, as well as its evaluation based on the framework's concepts, such as system use, behavioural intention to use, perceived usefulness, perceived ease of use etc. Finally, CMX can be used in other fields or by other educational institutions, as it is freely available online to the educational community.

REFERENCES

[1] T. Koulouri, S. Lauria, and R. D. Macredie, "Teaching introductory programming: A quantitative evaluation of different approaches," *ACM Trans. Comput. Edu.*, vol. 14, no. 4, 2014, Art. no. 6.

[2] W.-C. Chang and Y.-M. Chou, "Introductory C programming language learning with game-based digital learning," in *Proc. Adv. Web Based Learn.*, 2008, pp. 221–231.

[3] R. Moser, "A fantasy adventure game as a learning environment: Why learning to program is so difficult and what can be done about it," in *Proc. ACM SIGCSE Bull.*, 1997, vol. 29, no. 3, pp. 114–116.

[4] L. Johnson, S. Adams Becker, V. Estrada, A. Freeman, P. Kampylis, R. Vuorikari, and Y. Punie, "Luxembourg: Publications Office of the European Union, & Austin, Texas: The New Media Consortium, " *Horizon Rep. Eur. 2014 Schools Edition*, pp. 1–52, 2014.

[5] M. Papastergiou, "Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation," *Comput. Edu.*, vol. 52, no. 1, pp. 1–12, 2009.

[6] T. M. Connolly, A. E. Boyle, E. MacArthur, T. Hainey, and J. M. Boyle, "A systematic literature review of empirical evidence on computer games and serious games," *Comput. Edu.*, vol. 59, no. 2, pp. 661–686, 2012.

[7] C. J. Ferguson and A. Garza, "Call of (civic) duty: Action games and civic behavior in a large sample of youth," *Comput. Human Behavior*, vol. 27, no. 2, pp. 770–775, 2011.

[8] C. S. Green and D. Bavelier, "Learning, attentional control, and action video games," *Current Biol.*, vol. 22, no. 6, pp. R197–R206, 2012.

[9] L. A. Jackson, E. A. Witt, A. I. Games, H. E. Fitzgerald, A. von Eye, and Y. Zhao, "Information technology use and creativity: findings from the children and technology project," *Comput. Human Behavior*, vol. 28, no. 2, pp. 370–376, 2012.

[10] M. Ventura, V. Shute, and W. Zhao, "The relationship between video game use and a performance-based measure of persistence," *Comput. Edu.*, vol. 60, no. 1, pp. 52–58, 2013.

[11] M. Ventura, V. Shute, and Y. J. Kim, "Video gameplay, personality and academic performance," *Comput. Edu.*, vol. 58, no. 4, pp. 1260–1266, 2012.

[12] F. Garzotto, "Investigating the educational effectiveness of multiplayer online games for children," in *Proc. 6th ACM Int. Conf. Interaction Des. Children*, 2007, pp. 29–36.

[13] J. B. Beedle and V. H. Wright, "Perspectives from multiplayer vídeo Gamers," *Games Simulations Online Learn.*, pp. 150–174, 2007.

[14] M. D. Dickey, "Game design and learning: A conjectural analysis of how massively multiple online role-playing games (MMORPGs) foster intrinsic motivation," *Educational Technol. Res. Develop.*, vol. 55, no. 3, pp. 253–273, 2007.

[15] T. Barnes, H. Richter, A. Chaffin, A. Godwin, E. Powell, T. Ralph, P. Matthews, and H. Jordan, "The role of feedback in Game2-Learn," in *Proc. CHI*, 2007, pp. 1–5.

[16] T. Barnes, H. Richter, E. Powell, A. Chaffin, and A. Godwin, "Game2Learn: Building CS1 learning games for retention," *ACM SIGCSE Bull.*, vol. 39, no. 3, pp. 121–125, 2007.

[17] A. Chaffin, K. Doran, D. Hicks, and T. Barnes, "Experimental evaluation of teaching recursion in a video game," in *Proc. ACM SIGGRAPH Symp. Video Games*, 2009, pp. 79–86.

[18] M. Muratet, P. Torguet, F. Viallet, and J.-P. Jessel, "Experimental feedback on Prog&Play: A serious game for programming practice," in *Proc. Comput. Graph. Forum*, 2011, vol. 30, no. 1, pp. 61–73.

[19] J. O'Kelly, and J. Paul Gibson, "RoboCode & problem-based learning: A non-prescriptive approach to teaching programming," *ACM SIGCSE Bull.*, vol. 38, no. 3, pp. 217–221, 2006.

[20] C. Malliarakis, M. Satratzemi, and S. Xinogalos, "Towards a new massive multiplayer online role playing game for introductory programming," in *Proc. 6th Balkan Conf. Inform.*, 2013, pp. 156–163.

[21] M. Nosrati and R. Karimi, "General trends in multiplayer online games," *World Appl. Program.*, vol. 3, pp. 1–4, 2013.

[22] S. de Freitas and S. Jarvis, "A framework for developing serious games to meet learner needs," in *Proc. Interservice/Ind. Training, Simulation Edu. Conf.*, vol. 2006, NTSA, Baltimore, 2006.

[23] A. Yusoff, R. Crowder, L. Gilbert, and G. Wills, "A conceptual framework for serious games," in *Proc. 9th IEEE Int. Conf. Adv. Learn Technol.*, 2009, pp. 21–23.

[24] K. Salen and E. Zimmerman, *Rules of Play: Game Design Fundamentals*. Cambridge, MA, USA: MIT press, 2004.

[25] K. Kiili, “Content creation challenges and flow experience in educational games: The IT-Emperor case,” *Internet Higher Edu.*, vol. 8, no. 3, pp. 183–198, 2005.
[26] M. Song, and S. Zhang, “EFM: A model for educational game design,” in *Technologies for E-Learning and Digital Entertainment*. Berlin, Germany: Springer, 2008, pp. 509–517.
[27] R. Ibrahim and A. Jaafar, “Educational Games (EG) design framework: Combination of game design, pedagogy and content modeling,” in *Proc. IEEE Int. Conf. Electr. Eng. Inform.*, 2009, vol. 1, pp. 293–298.
[28] C. Malliarakis, M. Satratzemi, and S. Xinogalos, “Designing educational games for computer programming: A holistic framework,” *Electron. J. E-Learn.*, vol. 12, no. 3, pp. 281–298, 2014.
[29] E. Wenger, *Communities of Practice: Learning, Meaning, and Identity*. Cambridge, U.K.: Cambridge Univ. Press, 1999.
[30] S. M. Fisch, “Making educational computer games educational,” in *Proc. Conf. Interaction Des. Children*, 2005, pp. 56–61.
[31] E. Brown and P. Cairns, “A grounded investigation of game immersion,” in *Proc. Extended Abstracts Human Factors Comput Syst.*, 2004, pp. 1297–1300.
[32] M. Suznjevic, O. Dobrijevic, and M. Matijasevic, “MMORPG player actions: Network performance, session patterns and latency requirements analysis,” *Multimedia Tools Appl.*, vol. 45, no. 1–3, pp. 191–214, 2009.
[33] C. Malliarakis, M. Satratzemi, and S. Xinogalos, “CMX: Implementing an MMORPG for Learning Programming,” in *Proc. 8th Eur. Conf. Games Based Learn.*, 2014, Art. no. 346.
[34] S. Natkin and C. Yan, “User model in multiplayer mixed reality entertainment applications,” in *Proc.ACM SIGCHI Int. Conf. Adv. Comput. Entertainment Technol*, 2006, Art. no. 85.
[35] H. Desurvire, M. Caplan, and J. A. Toth, “Using heuristics to evaluate the playability of games,” in *Proc. Extended Abstracts Human Factors Comput. Syst.*, 2004, pp. 1509–1512.
[36] C. Malliarakis, M. Satratzemi, and S. Xinogalos, “Towards optimizing server performance in an educational MMORPG for teaching computer programming,” in Proc. 11th Int. Conf. Numerical Anal. Appl. Math., 2013, vol. 1558, no. 1, pp. 345–348.
[37] C. Malliarakis, M. Satratzemi, and S. Xinogalos, “Integrating learning analytics in an educational MMORPG for computer programming,” in *Proc. IEEE 14th Int. Conf. Adv. Learn. Technol.*, 2014, pp. 233–237.
[38] D. Johnson and J. Wiles, “Effective affective user interface design in games,” *Ergonomics*, vol. 46, no. 13–14, pp. 1332–1345, 2003.
[39] N. Lazzaro and K. Keeker, “What’s my method?: A game show on games,” in *Proc. Extended Abstracts Human Factors Comput. Syst.*, 2004, pp. 1093–1094.
[40] J. P. Gee, “Learning by design: Games as learning machines,” *Interactive Edu. Multimedia*, vol. 8, pp. 15–23, 2010.
[41] M. A. Federoff, “Heuristics and usability guidelines for the creation and evaluation of fun in video games,” PhD dissertation, Indiana Univ., Bloomington, IN, USA, 2002.
[42] P. Vorderer, T. Hartmann, and C. Klimmt, “Explaining the enjoyment of playing video games: The role of competition,” in *Proc. 2nd Int. Conf. Entertainment Comput.*, 2003, pp. 1–9.
[43] P. Sweetser and D. Johnson, “Player-centered game environments: Assessing player opinions, experiences, and issues,” in *Proc. Entertainment Comput.*, 2004, pp. 321–332.
[44] M. Terlecki, J. Brown, L. Harner-Steciw, J. Irvin-Hannum, N. Marchetto-Ryan, L. Ruhl, and J. Wiggins, “Sex differences and similarities in video game experience, preferences, and self-efficacy: Implications for the gaming industry,” *Current Psychol.*, vol. 30, no. 1, pp. 22–33, 2011.
[45] M. Kebritchi, A. Hirumi, and H. Bai, “The effects of modern mathematics computer games on mathematics achievement and class motivation,” *Comput. Edu.*, vol. 55, no. 2, pp. 427–443, 2010.
[46] G.-J. Hwang,et al., “A learning style perspective to investigate the necessity of developing adaptive learning systems,” *Educational Technol. Soc.*, vol. 16, no. 2, pp. 188–197, 2013.
[47] F. D. Davis Jr, “A technology acceptance model for empirically testing new end-user information systems: Theory and results,” PhD dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 1986.

Christos Malliarakis received the BS degree in applied informatics, the MS degree in information systems, and the MS degree in informatics and administration, in 2007, 2010, and 2011, respectively. He received the PhD degree in serious games from the Department of Applied Informatics at the University of Macedonia, Greece. He is an author or coauthor of more than 25 research papers on serious games for computer programming.

Maya Satratzemi is a professor in the Department of Applied Informatics, University of Macedonia. Her current main research interests include the area of educational programming environments and techniques, didactics of informatics, adaptive and intelligent systems, collaborative learning systems, and game-based learning. She has published a significant number of papers in international journals and in international and national conferences. She was a conference cochair of the 8th ITiCSE (ACM).

Stelios Xinogalos is an assistant professor in the Department of Applied Informatics, University of Macedonia. He is an author or coauthor of more than 70 research papers on programming environments and techniques,object-oriented design and programming, educational environments, and games for programming.