Designing Educational Games for Computer Programming: A holistic Framework

Christos Malliarakis, Maya Satratzemi and Stelios Xinogalos
University of Macedonia, Thessaloniki, Greece
malliarakis@uom.gr
maya@uom.gr
stelios@uom.gr

Abstract: Computer science is continuously evolving during the past decades. This has also brought forth new knowledge that should be incorporated and new learning strategies must be adopted for the successful teaching of all sub-domains. For example, computer programming is a vital knowledge area within computer science with constantly changing curriculum and its teaching remains a difficult endeavour. On the other hand, students start from a very early age to interact with computers through games and other entertaining multimedia software. Therefore, they seem to be keen on environments with impressive special effects and graphical interfaces where they interact with the environment's elements. In response, teachers are trying to connect computer programming learning with computer operations that students are familiar with, which does not include textual editors for programming lines of code with no other interaction. Educational games used in computer programming courses are considered to benefit learning, because they motivate students towards actively participating and interacting with the game's activities. Thus, we have developed an educational multiplayer game that aims to further enhance computer programming education by addressing occurring problems. This process, however, requires proper planning during the design of educational games, and thus the availability of adequate guidelines that include all characteristics that should be incorporated in such games. This paper aims to introduce and elaborate on a holistic framework that has been constructed as a guide towards the development of this game. To this end, we study existing frameworks that have been proposed for the design of educational games and document features currently supported by educational games that teach computer programming. We conclusively propose the framework we have constructed for the design of our game. This framework can be used for the design of other computer programming-specific educational games and extended for other educational domains.

Keywords: computer programming; educational programming environments; educational games; holistic framework; learning process

1. Introduction

Computer science is one of the most fundamental courses worldwide, and the field seems to present a continuous and radical progress during the past decades due to the technology's global advancements. This has also brought forth new knowledge that should be constantly incorporated in each course and new learning strategies should be adopted for their successful teaching. Studies have indicated that some sub-domains of computer science education, such as computer programming, present significant problems (e.g. lack of motivation, complex programming elements, lack of scaffolding etc) that hinder their successful teaching and learning (Lahtinen et al. 2005; Ragonis & Ben-Ari, 2005). Thus, there is an increasing need for employing technologies that can tackle these difficulties, incorporating the new knowledge and adopting new learning strategies to provide efficient and effective computer programming education.

Studies have indicated that three main categories of technologies exist that can address the aforementioned objective; these are educational programming environments, microworlds (Brusilovsky et al, 1997) and educational games (Gunter et al, 2008). Educational games have started evolving rapidly over the past few years, especially due to the fact that students today are being called the "Nintendo Generation" since they have become very familiarized with computer games in their everyday lives (Guzdial & Soloway, 2002). Additionally, educational games seem to be more apt for being used in computer programming education, since their provision of attractive graphical environments, interesting scenarios and high interactivity

Reference this paper as: Christos Malliarakis, Maya Satratzemi and Stelios Xinogalos "Designing educational games for computer programming: A holistic framework" *The Electronic Journal of e-Learning* Volume 12 Issue 3 2014, (pp281-181) available online at www.ejel.org

motivate students to learn computer programming concepts through achieving goals within an environment they are already familiar with.

To this end, we have developed an educational game called CMX that aims to provide functionalities that help students to learn computer programming elements and teachers to organize their courses efficiently. The process of developing an educational game, however, requires taking into consideration multiple aspects that regard e.g. the technology, the pedagogy, the domain etc. This is facilitated by frameworks that serve to inform designers and developers of educational games on what elements should be supported.

The availability of such frameworks that provide adequate guidelines for designing and developing games for the educational domain is still considered to be a work in progress where more empirical studies are needed (Fisch, 2005; Dondlinger, 2007; Wong et al, 2007; de-Freitas, 2006). Even though a number of frameworks provide interesting concepts that should be taken into consideration, their practical application in designing educational games for specific learning domains and their corresponding evaluation and adaptation is still lacking.

This paper aims to present a holistic framework that has been constructed to guide the design of our game. The framework takes into consideration related work and is augmented with concepts that intend to support features that will provide a highly motivational and interesting virtual world for computer programming courses. The second section presents an overview of existing frameworks that have been proposed by published works for the design and development of domain-independent and computer programming-specific educational games as well as specific features that should be supported. The next section elaborates on our framework by presenting the methodology followed, a brief overview of the concepts included as well as a concise comparison of our framework with all identified existing frameworks. The paper concludes with a summary of the work done.

2. Related work

This section presents interesting frameworks proposed for the design of educational games in general. We continue to distinguish features supported by the most commonly known educational games that focus on computer programming education, following a top-down methodology approach.

2.1 Educational Games Frameworks

2.1.1 Four – dimensional framework

The Four-dimensional framework has been proposed by de Freitas & Jarvis (2006), as shown in Figure 1.

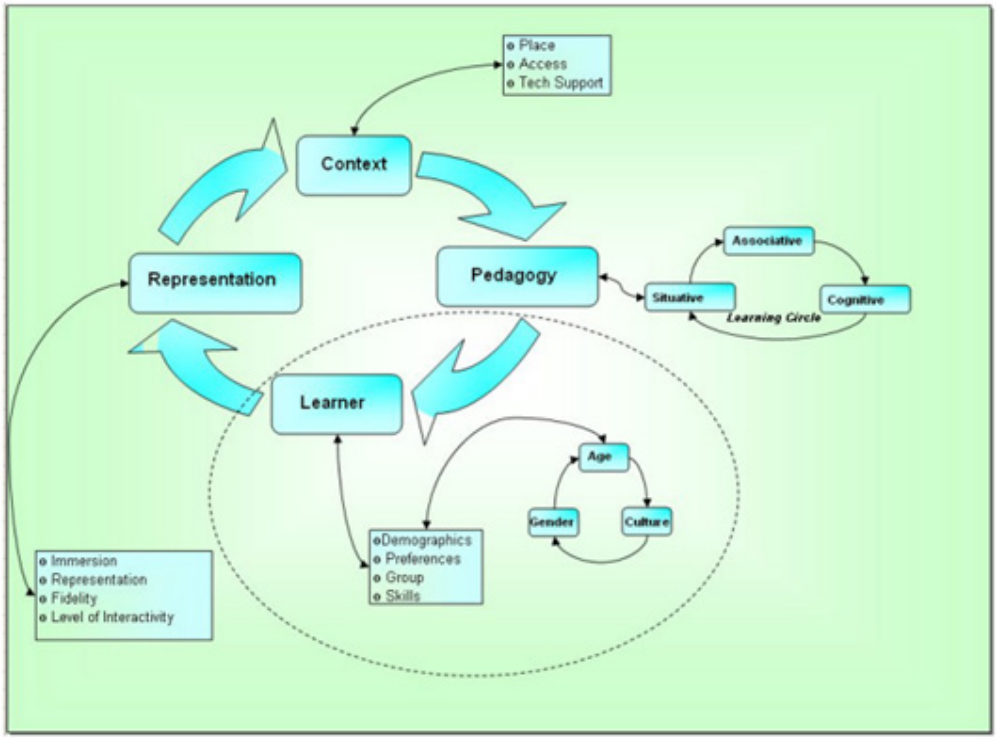

Figure 1: Four – dimensional framework (de Freitas & Jarvis, 2006)

The work done and portrayed within the framework shows that this model comprises of four basic principles, as follows:

- Context. Each game is characterized by a specific context that will guide the scenarios as well as the ways students and teachers will interact with its features. During the context's establishment, one must define characteristics such as required infrastructure, technical specifications, location of usage, type of game (e.g. role playing, multiplayer etc), activities to be performed etc.
- Representation. This concept refers to all representations that are required to be properly portrayed within the game. For example, each player needs to be represented by avatars that will have specific characteristics based on the context of the game. Additionally, the virtual world should represent interesting scenery that will be integrated with all the features of the game in a harmonized and meaningful way. Successful representation is vital for the increased motivation of students, since they need to be intrigued in order to want to learn by playing. An important metric that can determine this is the quality of the graphics employed during the representation, which needs to be high so that it can create better and more immersive simulations.
- Learner. This concept relates to all features corresponding to the learner within the game. Some of these include the ages of the students to be taught, their preferences, the availability and level of previous knowledge on the specific learning domain, our learning objectives regarding the game's learning outcomes etc. Additionally, it is considered important by the relevant literature to try and promote learning through groups in educational games (Sandford, 2006), as collaborative learning is gradually being employed in education (Kayes et al., 2005).
- Pedagogy. The most important factor that distinguishes an educational game from a computer game is the pedagogical aspect, i.e. the fact that the entire game and its activities are developed in order to fulfil learning objectives and to result in learning outcomes. To this end, the development of such games should depend on the study and incorporation of learning strategies. These strategies later on determine how the game will be integrated into the learning process so that it will produce the desired outcomes. Some

representative examples of employed learning strategies are problem-based learning or experiential based learning, which allow a variety of pedagogical models for learning processes, especially when they use online technologies (Mayes & de Freitas, 2006). The pedagogies used should promote learning processes that will be constructivist and cognitive so that they will allow the creation of new knowledge by the students as well as their in depth comprehension of the taught material through active and collaborative participation. Additionally, learning processes have to be instructive and associative, that is they are required to progress in a logical manner and provide assistance when needed. Finally, collaborative learning when knowledge is exchanged through practice is also a feature that should be supported by learning processes within educational games (Mayes & de Freitas, 2004).

2.1.2 Conceptual framework

The second reviewed framework has been proposed by Yusoff et al. (2009) so as to provide a reference guide for the design of educational games, as shown in Figure 2.

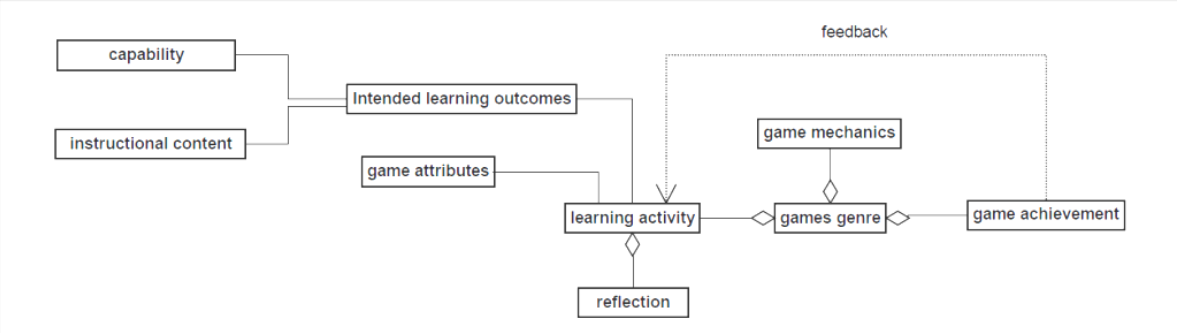

Figure 2: Conceptual framework for educational games (Yusoff et al., 2009)

A brief overview of the framework's features is provided as follows:

- Capability. This relates to the skills students should develop through their interaction with the game within the learning process. Such skills include analysis, recall, evaluation, effective execution of tasks, proper attitude and logical thinking.
- Instructional content. The game should be compliant with the educational material that students have to learn while interacting with it; therefore a thorough examination of the units of learning to be studied is essential and will determine the types of activities and assessment methods students will engage in during the game.
- Intended learning outcomes. Learning outcomes represent what goals students should be able to achieve once they successfully complete all assigned tasks. As it is depicted in Figure 2, a learning outcome is closely inter-connected with the determined capabilities and content. Different learning outcomes correspond to a particular set of capabilities and content.
- Game attributes. This concept includes all characteristics that aim to increase motivational and participatory learning. Such are scaffolding, i.e. assistance given to the students when needed; interaction, i.e. types of engagement and feedback required by the game towards the student and vice versa; learner control and sequence, i.e. level in which students can navigate across the virtual world based on assigned activities or on their own without any guidance, incremental learning; i.e. each learning outcome is achieved incrementally through the execution of a set of activities and not all in the end of the game; rewards, i.e. incentives provided to students that accomplish their goals as an acknowledgement of their successful endeavours and to students that are close to accomplishing them as a motivation to try more; authentic learning, i.e. the virtual world and the activities that will simulate an interesting and attractive environment with which students are already familiar, either from their real life or from their interactions with commonly used computer games.

- Learning activity. Each activity plays an important role to the game and focuses on a specific set of tasks that need to be completed. It is important that all learning activities promote motivation so that students will remain interested and immersed in the game's scenario. The design of an activity should take into consideration the learning outcomes that have to result once students successfully execute the given tasks and thus should allow the development of the intended capabilities as well as the comprehension of the corresponding materials. Lastly, the established game attributes will define the activity development.
- Reflection. Students should be able reflect on their experience within the game and be provided with an overview of their progress when requested (Garris et al. 2002).
- Games genre. This concept describes the type of the specific game that is to be developed. Specific genres are accompanied by different features so it is important to define from the design phase what type of an educational game this will be (e.g. strategy, open-world sandboxes, role-playing etc).
- Game mechanics. This concept relates to the technicalities (e.g. management of resources, environment layout, database etc) that should be taken into consideration during the game's development depending on the game's genre, learning activities or instructional content.
- Game achievement. The final concept refers to all the ways the game can represent a student's achievement level and is also a significant metric of learning assessment. For example, the final scores each student has gathered along with any additional resources or "rewards" provide an overview of individual achievement and thus show to which level the intended learning outcomes were accomplished.

2.1.3 The Design, Play, and Experience Framework

The "Design, Play and Experience" framework aims to depict the relation between the designer and the player and explicitly demonstrate the concepts that correspond to each layer that is designed depending on the phase and user type. More specifically, the designer focuses on the "Design' phase and has to initially identify the learning objectives that will guide the activities to be designed and that will be used as an evaluation metric once the development of the game is complete. Additionally, all other layers to be designed and shown in Figure 3 are initially based on the learning objectives and are later on configured and refined based on feedback from incremental and iterated engagement with the game, a process represented by the "Experience" phase. This is clearly depicted by the arrow connecting the "Experience" phase with the "Design" phase, expressing how continuous and practical engagement can affect the design phase of a next version of the game (Salen & Zimmerman, 2004).

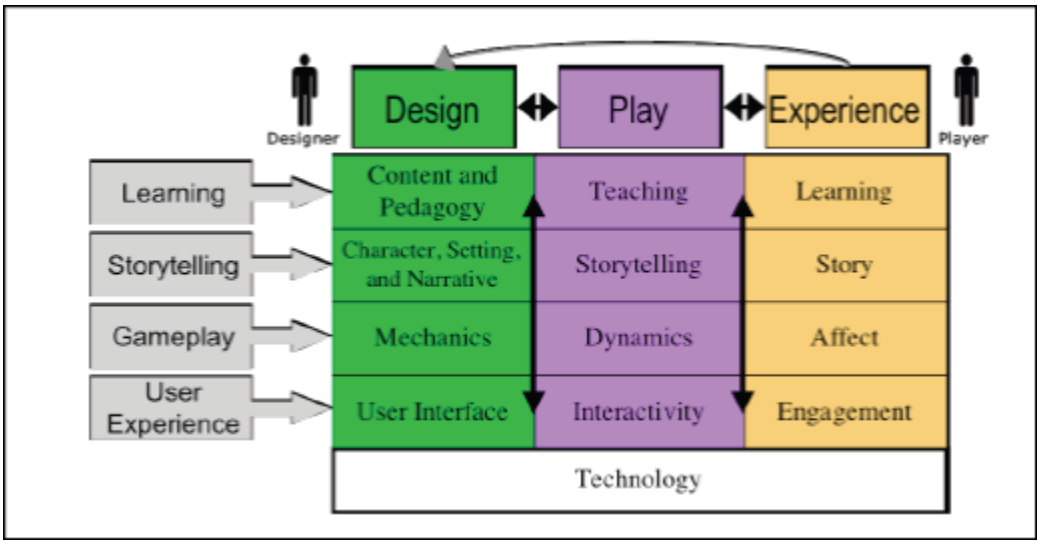

Figure 3: The Design, Play, and Experience Framework (Salen & Zimmerman, 2004)

The "Play" phase represents the interaction of players with the game's features. Thus, this phase is closely related to all characteristics of each individual player, such as knowledge background, skills etc. To this end, during the "Play" phase designers must take into consideration the target audience that will use the educational game and that will produce the different experiences during the "Experience" phase.

According to this framework, four different layers guide the design of an educational game. A brief overview of these layers is provided as follows:

- Learning. During the "Design" phase, the educational content that will be taught along with the pedagogical theory that will guide the learning process corresponds to the learning layer. In the "Play" phase, this refers to the actual teaching process, which is when students play the game. The "Experience" phase finally represents the learning that is accomplished through the teaching, and thus documents whether the set learning objectives have been achieved as learning outcomes.
- Storytelling. Storytelling provides valuable information during the game's design that will guide the virtual world development as well as all the scenarios to be supported within the game. The designer sets the stage by designing the different characters to be included, the overall environment setting, the narrative and the different layouts that will structure the world (Rouse, 2001). Even though the designer sets the story, each player produces his individual storyline through the way he engages with the game's activities during the "Play" phase. Similarly, the final player's story as it will be created after the execution of all assigned tasks will represent the accomplished learning outcomes in the "Experience" phase.
- Gameplay. This layer includes all information regarding the players' allowed actions within the game (Adams & Rollings, 2007). Initially, the designer has to define the specific mechanics of the game, such as the learning objectives, the challenges and allowed actions. Once these are integrated in the game, they are represented by the dynamics during game playing, i.e. all the different actions players take in the game that lead to individual pathways and interactions with the game's features. The experience drawn from the play by each user is called an "affect" and represents all emotions players are left with after they are finished (e.g. satisfaction, disappointment, desire to try again etc).
- User experience. The final layer of the reviewed framework includes the most visual part of the game, which has to be as entertaining and accessible as possible in order to increase motivation and participation (Saltzman, 2000). The design phase supports the planning of the user interface and aims to provide multiple and easy to use interactivity opportunities during game playing. Finally, this will lead to experiences that engage students and accomplish in-depth comprehension of the learning material and successful skills development.

2.1.4 Experiential Gaming Model Framework

The experiential gaming model framework describes learning as a circular process that constructs cognitive schemas through activities within the game's world (Kiili, 2005). The experiential gaming model framework aims to help game designers to understand the learning mechanism that should be employed in educational games, as shown in Figure 4. According to this model, the direct interactions and experiences players have with the game world create a circular learning process that includes all necessary steps to ensure successful learning objectives achievement. To this end, the model suggests that the activities that will be supported by an educational game should not only be cognitive but behavioural also. This way, players will behave within the activities of the game in a way that will allow them to build cognitive structures.

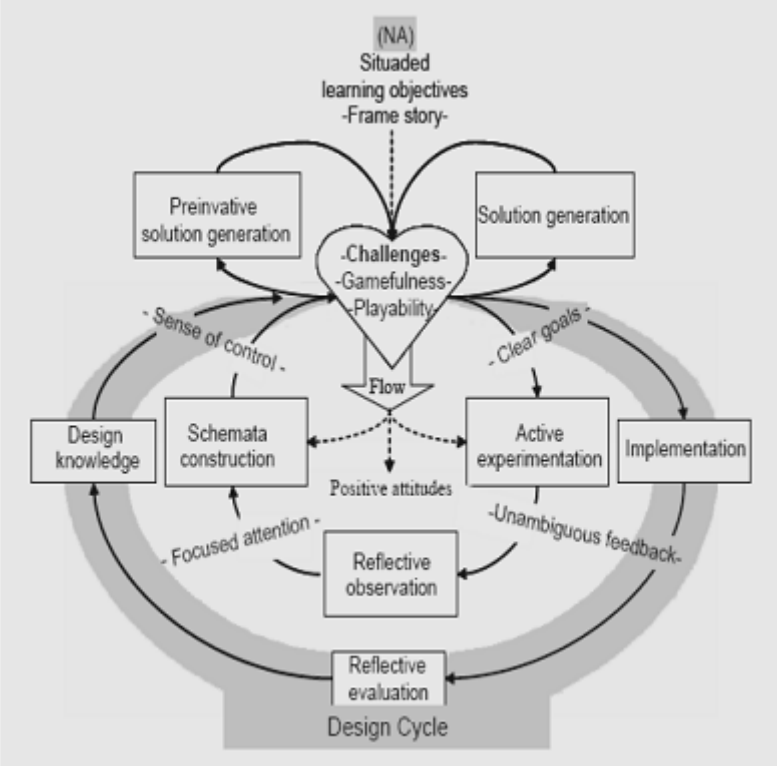

Figure 4: Experiential Gaming Model (Kiili, 2005)

According to Kiili (2005), important elements of an educational game should be the scenario that will set up the learning objectives, the challenges students will have to face as well as the flow of the game. Additionally, the world of the game should allow active experimentation with the environment's features so as to develop positive attitudes towards the game. Furthermore, feedback should be provided from the game during interaction that will allow students to reflect on their knowledge gained and to construct proper schemata and evaluate their own performance.

This experiential gaming model emphasizes that deep engagement of students in the learning process should be enabled in all of the game's functionalities; students should always be learning while playing, whether they are chatting with other players or reading an educational material or executing a task or reflecting on their performance etc.

2.1.5 EFM: Model for Educational Game Design

A multi-dimensional model called EFM (Effective learning environment, Flow experience and Motivation) has been proposed regarding the proper design of educational games by Song & Zhang (2008), as shown in Figure 5. According to this model, an educational game's environment should be able to support seven fundamental requirements that are highly interconnected with the other axes proposed in the model. These requirements include the availability of appropriate tools within the game as well as the sense of motivation along with the sense of direct engagement to activate students in the learning process. Moreover, the designers should ensure to avoid any possible distractions that could sway students' attention away from learning and enable a constant sense of challenge that will allow them to go through all of the activities scheduled by the teachers and accomplished the clearly set educational goals.

The model also suggests that these sever dimensions of the requirements will determine the experiences within the game and their flow. Finally, the authors indicate four main strategy components that will stimulate motivation for students, namely "Interest", "Goal", "Feedback" and "Challenge". The activities and experiences

within the game's world should be designed so as to enable all these four motivation strategies in order to create an effective educational game.

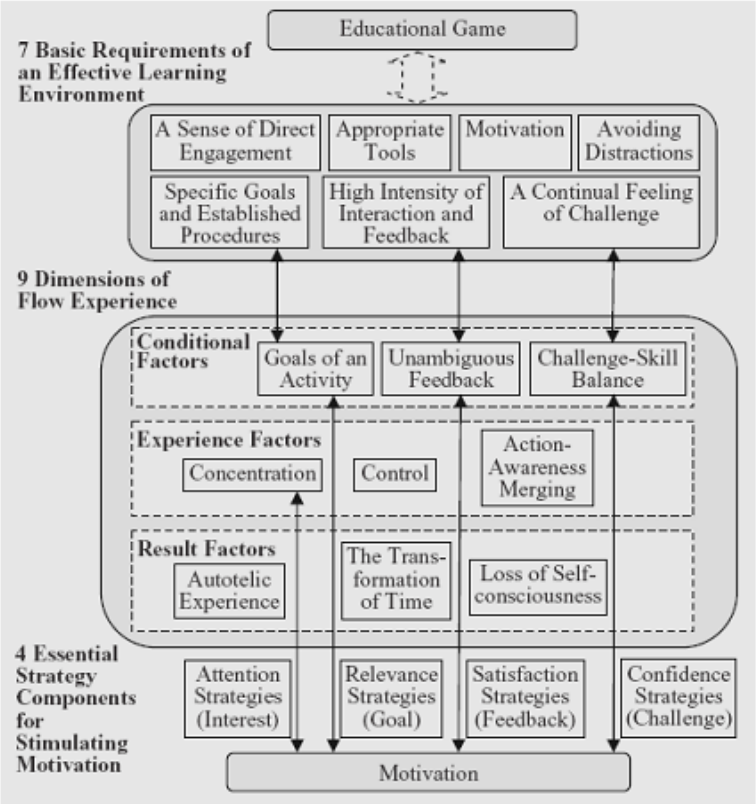

Figure 5: EFM: Model for Educational Game Design (Song and Zhang, 2008)

2.1.6 Educational Games Design Model Framework

The model proposed by Ibrahim & Jaafar (2009) includes three major factors that should be taken into consideration when designing educational games, as shown in Figure 6. The factors are game design, pedagogy and learning content modelling. Each of these is further analyzed within the model to provide analytical information on how to incorporate its features to the game.

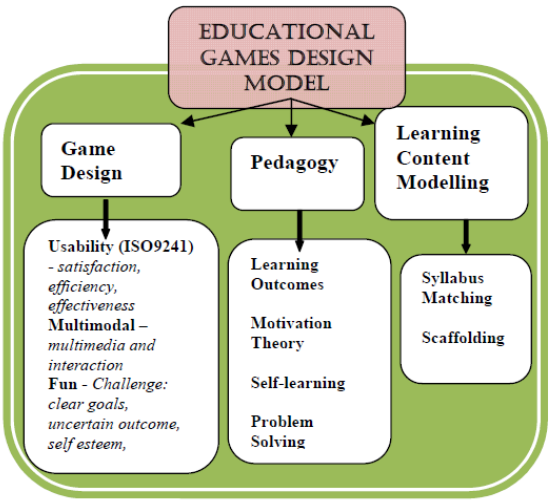

Figure 6: Ibrahim's and Jaafar's Educational Games Design Model Framework (Ibrahim and Jaafar, 2009)

Initially, the authors suggest that during game design, the usability of the game should be ensured, and more specifically tested under the ISO 9241 usability standard (Pinelle and Wong, 2008). An educational game is considered usable when it provides satisfaction to its players, it is effective in achieving the goals set before playing and it is efficient in allowing consistent and responsive functionalities. Moreover, it is considered important to allow multimodal content in educational games (e.g. text, audio, videos, animation, graphics etc) as well as the ability for players to directly interact with this content and receive appropriate feedback. Finally, game design should also support the entertaining element of a computer game, which is essential (Prensky, 2001) because games should be fun to interact with even in educational settings. To this end, designers are expected to include functionalities that will allow teachers to set clear educational goals and activities that will challenge students during playing, will engage them in navigating through the game's features by increasing their curiosity even though they will not know the outcome (Malone, 1980) and will boost their self –esteem.

Furthermore, the authors include the pedagogical factor in the design model, which suggests that the computer games should be designed according to the educational domain and incorporate proper learning strategies to ensure that the game will indeed result in the desired learning outcomes. More specifically, the model is drafted so as to support learning outcomes that belong within the three first levels of the Bloom Taxonomy, namely knowledge, comprehension and application (Ibrahim & Jaafar, 2009). These three levels of knowledge should be accommodated through appropriate theory availability and modules that motivate learners to reach the 3rd level of knowledge. Additionally, educational games designed according to this model should support self – learning and thus reflection mechanisms; these games allow students to teach themselves by playing, reading the learning materials and assess their own performance. Thus, the authors indicate that such games could be developed as web-based environments so that they can be accessed by learners at any time. The active role of learners in the educational games will increase their competences since traditional learning where knowledge is simply delivered will be replaced by a series of problem solving activities that will stimulate learners' minds and allow them to learn how to solve problems by interacting with materials and tasks within the game. This calls for specific types of learning materials that allow self-learning and problem solving.

Therefore, it is important to also consider the last factor included in the design model, namely the learning content modelling. The content incorporated within the game should be available at specific parts of the game in order to ensure the proper solving of the given problem and thus result in the set learning outcomes. Moreover, features such as a syllabus, terms matching tools and scaffolding mechanisms would increase learners' sense of security since they will mostly be learning on their own, with the teachers available only during lectures time to provide guidance.

2.2 Features in games teaching computer programming

We continue to study existing games that focus on the computer programming learning domain and distinguish specific features that need to be supported by future educational games.

2.2.1 The SCRATCH environment

The MIT Media Lab research group developed Scratch aiming to create an environment that would assist creative people to easily accomplish their goals. This aim is also depicted in Figure 7.

Figure 7: Ultimate aim of the MIT Media Lab (Resnick, 2007)

The Scratch environment was based on the above design principles and introduces computer programming to students of ages 10 to 18 (Resnick, 2007). Towards this goal, students can create programs, interactive stories and games through drag & drop tiles instead of writing lines of code, thus avoiding making syntax errors. Interaction and scenarios play an important role in this game as they intend to stimulate the interest of young students by engaging them in a series of attractive assignments. The availability of multiple characters (sprites), backgrounds, sounds and images motivate students to create their own animations and games, play on their own and share their creations with their classmates, as well as reflect their creations to others.

2.2.2 Educational games for computer programming

A thorough investigation was carried out on frameworks and models proposed for the design of educational games that have been specifically designed and developed to support computer programming education. During this research, we identified one architecture that is further analyzed and a list of important features that should be supported by educational games constructed for teaching computer programming concepts.

More specifically, the identified architecture shown in Figure 8 was proposed by Maragos & Grigoriadou (2005) for the design of intelligent educational games that aim to teach computer programming. This architecture was employed for the design and development of the educational game TALENT, which supports algorithmic if-statements and loops through interactive role-playing.

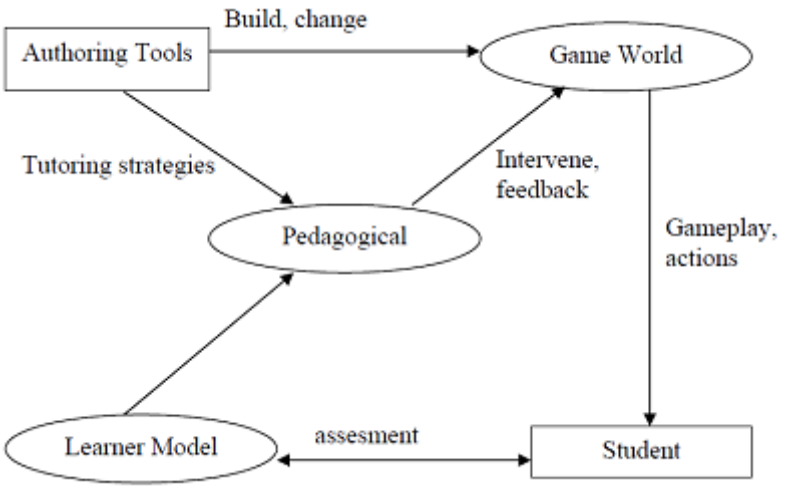

Figure 8: Architecture for Intelligent Educational Game (Maragos & Grigoriadou, 2005)

According to this model, the world of the game is constructed and can be changed through authoring tools by the teachers. These authoring tools should also allow the addition of pedagogies in the game's environment, such as tutoring strategies that support successful learning. This is usually done via a pedagogical agent, which is represented as a character or a tool within the educational game. The agent usually intervenes in the game's process and provides feedback whenever a student with a specific profile requires it.

Furthermore, the game supports multiple learner models that can be assessed according to each student's specific performances and profiles. These models are built in accordance to the pedagogical goals the teachers set while they are constructing the game's world and depending on the educational materials to be taught as well as the activities/ steps that the students have to go through.

All of the above components of the model suggested allow for great configuration rights by the teachers, thus providing abundant flexibility to create multiple different instances of the same game. Teachers can study their students' behaviours and correspondingly adapt the game's scenario, materials, activities, goals and desired outcomes.

Furthermore, we continue to identify the main features that should be supported by educational games designed and developed for computer programming courses. The work done is the combination of previous research (Malliarakis et al. 2012) and a more thorough overview of the educational games Catacombs (Barnes et al., 2008), Saving Sera (Barnes et al., 2008), EleMental (Chaffin et al., 2009), Prog & Play (Muratet et al., 2010), Robozzle (Li & Watson, 2011), Lightbot (Piteira & Haddad, 2011), Robocode (O'Kelly & Gibson, 2006), TALENT (Maragos & Grigoriadou, 2011), M.U.P.P.E.T.S. (Phelps et al., 2003), Wu's Castle (Eagle & Barnes, 2009), Playlogo 3D (Paliokas et al, 2011) and Gidget (Lee & Ko, 2011). These educational games follow a "play-learn-improve-win" pattern during learning, a pedagogical aim that is similar to the one employed in Scratch and depicted in Figure 7.

According to the review carried out, the main features supported can be considered as requirements specification for future educational games' design and development. A brief overview of the most commonly identified features is provided as follows:

- Multiplayer / Role – playing
- Interaction / Experimentation
- Collaboration
- Scaffolding
- Drag & drop lines of code
- Programming editor for writing lines of code
- Multiple choice questions
- Scenarios
- Compiler that allows interaction with errors
- Physical/ familiar metaphor
- Visualization of concepts
- Simplicity

Each game supports a different set of the features included in the list as well as of the features shown in the examined frameworks during their design. Thus, we have worked on developing a game that will include as many of these features and concepts as possible, while maintaining limited complexity of the environment.

3. Our framework

This section provides information regarding the design process of our CMX educational game that focuses on the education of the computer programming domain. The CMX educational game is a Massive Multiplayer Online Role-Playing Game (MMORPG) and aims to familiarize secondary school students that are novices to computer programming with concepts such as variables, if-statements, loops etc, and engage them in algorithmic logic. It is different than existing systems, since the game supports different functionalities for the student (player), the teacher (tutor), the administrators and other educational agents, according to each user type's role in the game.

As shown in the following figure, tutors, administrators and users send their data in the central system, while the educational agents constantly send and receive information to and from the central system. Finally, the system in the figure represents the host of our educational game and sends information to the game regarding each virtual player as well as a different instance of the virtual world.

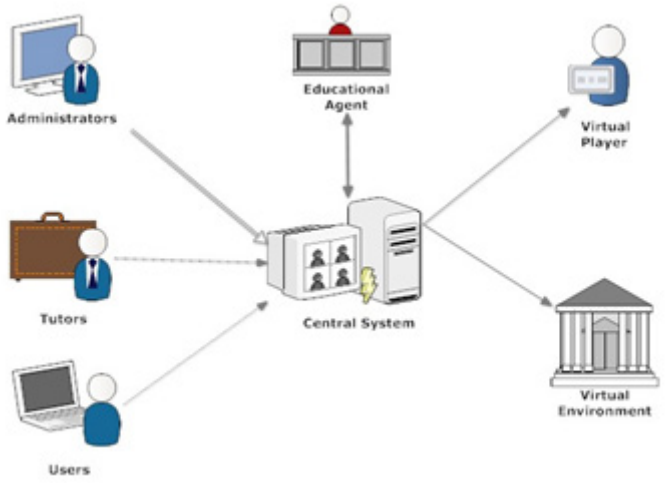

Figure 9: **MMORPG CMX data exchange**

It is essential to initiate the design of the game by studying its characteristics and defining generic metrics that will need to be instantiated based on the different user type. Therefore, we initially define a user-centric model for the design of educational games, as shown in Figure 10. Based on this model, the design phase takes into consideration each user type's initial aspirations that motivate them to use the game, the specific targets they set to achieve through the game, the metrics that will determine the level in which the targets are achieved as well as the feedback provided to them by the system depending on the corresponding target.

These inter-connected concepts are different for each user type and therefore relate to different aspects of the game. For example, the game administrator's aspirations can be the game's and server's problem-free operation and the corresponding targets can be the backup of the data produced, the frequent monitoring for possible malfunctions etc. Performance metrics could include the server's database capacity, the cache memory capacity etc, while feedback that will indicate whether the server is working properly can comprise of error messages produced by the game (e.g. in case new data cannot be saved).

Similarly, a teacher's aspirations can be the successful teaching of units on computer programming while the target can look like "at least three of the four game levels on loops will be achieved by all". Thus, the set metrics can be each level's and student's score, log files from the game etc, while the feedback can be a report with the student's progress at a given time/level.

Furthermore, a student's aspiration can be to enjoy his time within the game, to distinguish himself through accomplishing the assigned tasks and to learn. Based on these aspirations they can set their individual or group targets across the overall learning objectives. Their performance metrics can be the score/resources they have gathered, how many times they have had to retry before writing lines of code correctly etc. Finally, feedback can be explanatory messages from the agents, introductory narrative or video that explains the scenario, errors in the program they're writing etc.

This closely interconnected puzzle is shown in Figure 10 and provides a more abstract overview of the initial metrics taken into consideration during design phase, following a user-centric approach. This model takes the form of a puzzle since all user types need to be addressed equally and represent a different point of view of the game.

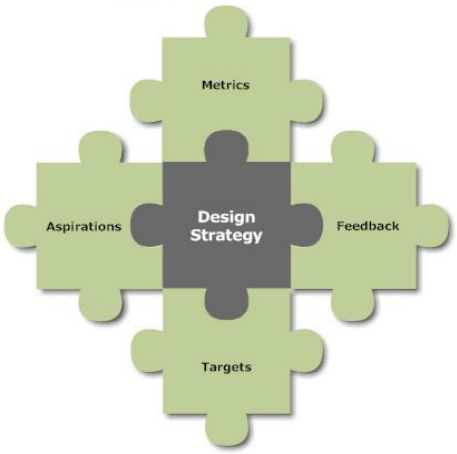

Figure 10: CMX Design Strategy Puzzle

The next step that will lead to the proper construction of our framework is the identification of the methodology our game will be based on. More specifically, as shown in Figure 11, students will initially engage in actions, which will in turn generate their desire for improvement through attractive scenarios. These actions will help students gain new knowledge regarding the targeted educational content and through collaborative activities, students will be able to produce the set learning outcomes. In the end, the learning objectives will be achieved after the game experience is properly evaluated.

Figure 11: Methodology steps of the CMX educational game

Finally, both aforementioned steps are reinforced from the review presented in the related work section. More specifically, based on the review of the existing frameworks that guide the design of educational games in general as well as of the features supported by educational games focused on computer programming education, we move on to propose an extended framework that includes the most commonly identified concepts as well as a series of sub-concepts that provide a more in-depth visualization of the steps to be followed during the design and development process. Based on this framework, which is shown in Figure 12, we have designed and developed an MMORPG educational game that aims to teach computer programming.

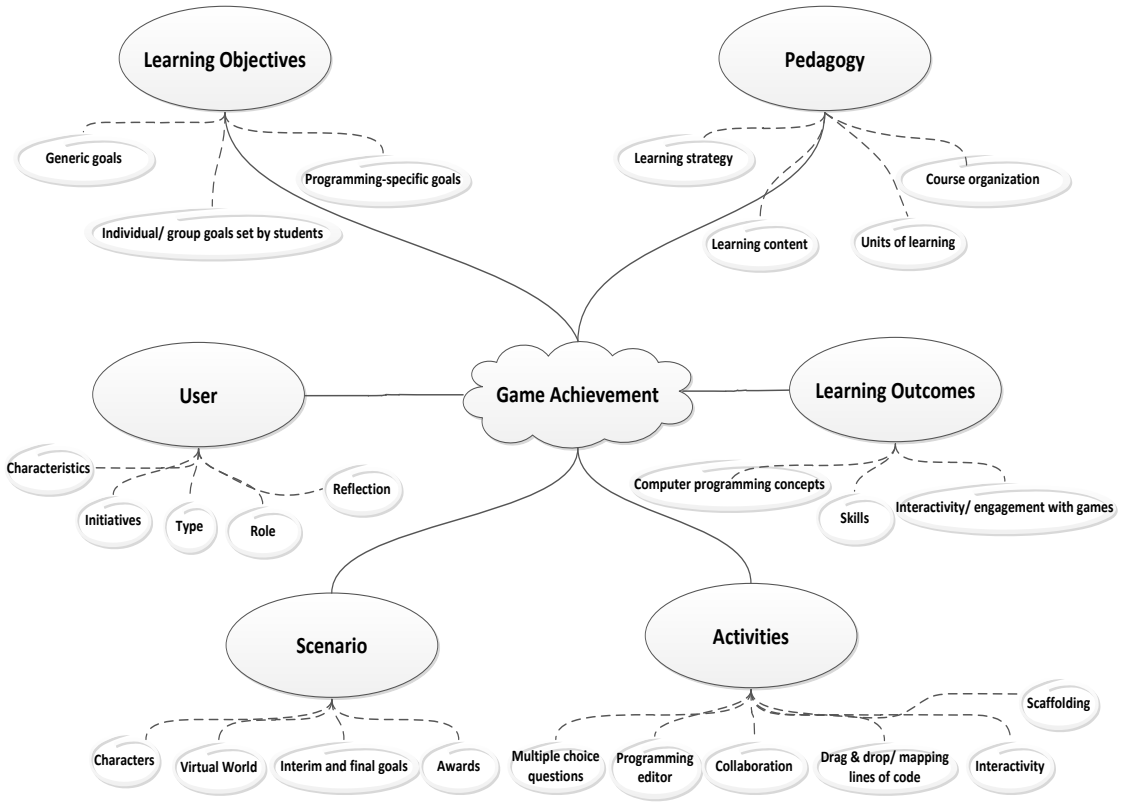

Figure 12: CMX Design framework

The CMX Design framework includes concepts that need to be represented within any educational game that aims to teach computer programming. It is abstract enough to be employed by future designers and developers and detailed enough to act as a solid guide without allowing many arbitraries. The most prominent concepts that define the game's design are:

- Infrastructure. The design initiates with the establishment of the infrastructure architecture, the technical requirements specification as well as the user interface and concepts visualization design. The infrastructure will have to support simplicity and ease of use.
- Learning objectives. Designers will have to initially define learning objectives that the game will be required to successfully support. These objectives can include generic goals (e.g. more than 80% of the students will complete all game's activities within the given deadline) or programming-specific goals (e.g. the 90% of the students will drag & drop correctly the lines of code regarding if statements). Additionally, students can also define their own goals depending on what they desire to achieve through their interaction with the game (e.g. "I have understood if statements so I want to complete all tasks related to loops").
- Pedagogy. The game's layout will strongly depend on the learning strategy to be employed during class as well as on the way the course will be organized (i.e. number of units of learning, educational material to be taught etc). For example, it is possible that the teacher will want to create a level per unit of learning, or merge specific units of learning into one game that will correspond to a certain set of learning objectives. Thus, it is essential to determine these features at an early stage.
- Learning outcomes. The determination of the learning outcomes is strongly inter-connected with the set learning objectives. Target outcomes can include the comprehension of the taught material, i.e. computer-programming concepts, such as variables declaration, if statements, loops etc, skills development, such as critical thinking, teamwork, leadership etc as well as interaction capabilities and engagement with innovative technologies and their features.

- User. A user in an advanced educational game can represent different types (e.g. student, teacher, administrator, agent), where each type signifies another aspect of the game's functionalities, as already explained in the CMX design strategy puzzle. Additionally, the specific characteristics of each user need to be determined (e.g. age, prior knowledge, preferences etc) as well the aspirations that will drive the user's interaction with the system (e.g. win the game-student, ensure problem free operation-administrator, instruct and assist students-teacher, guide and scaffold players-agent). Finally, each user will need to be able to reflect on the game experience for feedback gathering and future refinement of the game's functionalities.
- Scenario. The game's scenario should be thoroughly researched and planned out in order to produce an attractive and immersive virtual world (e.g. fighting arena, castle, forest etc) with interesting characters (e.g. wizards, robots, mentors, snowmen, prisoners etc) that are required to complete interim and final goals (e.g. save the princess, put the map's pieces in their correct order, navigate through the tree etc). Designers have to also define what types of awards players will be granted with during the game, in order to increase their motivation to continue learning.
- Activities. The design and development of individual activities is essential and will result to the interested and active participation of students. It is important that students will be able to interact with the world's elements and collaborate with others towards the achievement of all or some of the goals. Additionally, the environment should provide scaffolding mechanisms throughout all activities that will assist students during challenging tasks. Finally, a number of different ways in which students can contribute their knowledge should be included, since not all students learn better using the same techniques. Thus, an educational game should incorporate a programming editor, along with the ability for students to drag & drop lines of code as well as answer multiple choice questions.

The following table lists all features included in our proposed framework and depicts which features are supported by the corresponding layers of each of the seven frameworks studied during this research.

Table 1: Comparison of features supported by frameworks for educational games' design

CMX design framework	Four – dimensional framework	Conceptual framework	The Design, Play, and Experience Framework	Experiential Gaming Model Framework	EFM Model	Educational Games Design Model Framework	Architecture of Intelligent Educational Game
Learning objectives	X (Learner)		X (Learning)	X (Situated learning objectives)			X (Authoring tools)
Generic goals	X (Learner)		X (Learning)	X (Clear goals)	X (Goals)	X (Clear goals)	
Goals set by students							
Domain-specific goals					X (Specific Goals)		
Pedagogy	X (Pedagogy)					X (Pedagogy)	X (Pedagogical)
Learning strategy	X (Pedagogy)					X (Problem solving)	X (Pedagogical)
Course organization				X	X (Dimensions of flow experience)		X (Authoring tools)
Learning content		X (Instructional content)	X (Learning)	X (Design knowledge)		X (Learning content modeling)	X (Authoring tools)
Units of learning							
Scenario	X (Representa-tion)	X (Game attributes)	X (Storytelling	X (Frame story)	X (Established		X (Authoring tools)

			)		procedures)		
Characters	X (Representation)		X (Storytelling)				X (Authoring tools)
Awards		X (Game attributes)					
Interim/final goals			X (Storytelling)		X (Goals of an activity)		
Virtual world	X (Context)	X (Game attributes)	X (Storytelling)	X (Implementation)	X (Appropriate tools)		X (Game world)
Learning outcomes	X (Learner)	X (Intended learning outcomes)	X (Learning)	X (Solution generation)		X (Learning outcomes)	X (Learner model)
Skills		X (Capability)		X (Active experimentation)	X (Challenge-Skill balance)	X (Pedagogy)	
Knowledge			X (Learning)	X (Design knowledge)		X (Pedagogy)	X (Authoring tools)
Engagement		X (Game attributes)	X (User experience)	X (Schemata conctruction)	X (Sense of direct engagement)	X (Game design)	X (Game world)
Activities	X (Context)	X (Learning activity)	X (Gameplay)	X (Active experimentation)	X (Appropriate tools)	X (Game design)	X (Authoring tools)
Scaffolding		X (Game attributes)		X (Feedback)	X (High intensity of feedback)	X (Learning content modeling)	X (Game world)
Interactivity	X (Representation)	X (Reflection)	X (User experience)	X (Feedback, Reflective Observation)	X (High intensity of interaction)	X (Game design)	X (Game world)
Drag & drop						X (Learning content modeling)	
Programmin g editor							
Collaboration	X (Learner)						
Multiple choice questions							
Infrastructur e	X (Context)	X (Game genre)	X (User experience)				
User interface	X (Context)	X (Game mechanics)	X(User experience)				X (Student)
Compiler							
Game achievement		X (Game achievement)	X (Gameplay)			X (Game design)	X (Game world)

The concepts that were similar and included in the frameworks, even if they were not clearly represented by their models, were merged and included in our framework (e.g. the concepts within the Representation, Game attributes and Storytelling that referred to the game's scenario are shown with the "Scenario" concept in the CMX framework). The compiler and the programming editor are suggested specifically for the computer programming domain; however, the framework includes a variety of concepts that allow its exploitation by other courses as well. This framework was employed during the design and consequently development of the MMORPG CMX game and was a valuable support, especially during the requirements and architecture specification processes.

4. Conclusions

The configuration of computer games so as they can be integrated in the educational domain has generated a new trend in technologies used for education called educational games. These games are considered to make students more comfortable in actively participating in learning since they are already familiar with the

graphical environment and scenarios and provide adequate features that help progressive computer programming teaching (e.g. scaffolding mechanisms, collaboration etc). This way, students not only comprehend the knowledge taught but they also learn to develop competences that will be useful to them in their future career endeavours (e.g. problem solving, critical thinking etc).

Nevertheless, in order to make sure that educational games include all necessary features that will ensure successful teaching and learning, it is essential that they are first designed properly. This indicates the need for designers of educational games to follow an analytical guide that depicts all concepts that should be taken into consideration. Thus, in this paper we have reviewed existing frameworks that describe such concepts and documented the features specified by the relevant literature. Furthermore, we studied educational games currently used in computer programming courses that do not indicate to have followed a specific framework for their design and listed the features supported.

As a result, we took into consideration all the above work done, and created the CMX design framework, which includes all characteristics identified as essential and was used to design and develop the MMORPG CMX educational game designed and developed for computer programming teaching. The CMX design framework is abstract enough that can be used as a reference framework for designing other educational games for computer programming courses and can also be extended in the future to fit other educational domains.

References

Adams, E., & Rollings, A. (2007). Fundamentals of Game Design. p.277 River, NJ: Prentice Hall.

Barnes, T., Chaffin, A., Powell, E., Lipford, H. (2008). "Game2Learn: Improving the motivation of CS1 students", *Proceedings of the 3rd international conference on Game development in computer science education*, p.1-5, February 27-March 03, 2008, Miami, Florida

Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., and Miller, P. (1997). "Mini-languages: a way to learn programming principles." *International Journal of Education and Information Technologies*, 2, 65–83.

Chaffin, A. Doran, K., Hicks, D. and Barnes, T. (2009). "Experimental evaluation of teaching recursion in a video game", In Proc. *ACM SIGGRAPH Symposium on Video Games*, Sandbox '09, p. 79-86

de Freitas, S. and Jarvis, S. (2006). *A Framework for Developing Serious Games to meet Learner Needs. Interservice/Industry Training*, Simulation and Education Conference, 2006. Orlando, Florida.

de-Freitas, S. (2006). "Learning in immersive worlds: A review of game based learning", in JISC elearning progrmme. JISC: London.

Dondlinger, M.J. (2007). *Educational video game design: A review of the literature.* Journal of Applied Educational Technology, 4(1): p. 21-31.

Eagle, M., and Barnes, T. (2009). "Experimental evaluation of an educational game for improved learning in introductory computing", *SIGCSE Bull*. 41, 1 (March 2009), p. 321-325.

Fisch, S.M. (2005). *Making Educational Games "Educational"*, in *Conference on Interaction Design and Children IDC 2005*. 2005. Colorado, USA.

Garris, R. Ahlers, R. and Driskell, J. E. (2002). "Games, Motivation, and Learning: A Research and Practice Model", Simulation & Gaming, vol. 33, pp. 441-467, 2002.

Gunter, G. A., Kenny, R. F., and Vick, E. H. (2008). *Taking educational games seriously: using the RETAIN model to design endogenous fantasy into standalone educational games*. Educational Technology Research and Development, 56(5/6), 511-537.

Guzdial, M. and Soloway, E. (2002). *Log on Education: Teaching the Nintendo Generation how to Program*, Communications of the ACM, 45(4).

Ibrahim, R. and Jaafar, A., (2009). "Educational games (EG) design framework: combination of game design, pedagogy and content modeling", International Conference on Electrical Engineering and Informatics, 2009. vol.1, pp. 293-298.

Kayes, A. B., Kayes, D. C. & Kolb, D. A. (2005). "Experiential learning in teams" Simulation & Gaming 36(3): 330-354.

Kiili, K. (2005). Content creation challenges and flow experience in educational games: The IT-Emperor case Internet and Higher Education, pp183-198.

Lahtinen, E., Ala-Mutka, K. and Jarvinen, H. (2005). *A Study of Difficulties of Novice Programmers. In: Innovation and Technology*, Computer Science Education, p. 14–18.

Lee, M.J. and Ko, A.J. (2011). *Personifying Programming Tool Feedback Improves Novice Programmers' Learning*, Conference on International Computing Education Research (ICER), August 8-9, Providence, Rhode Island, USA, p. 109-116.

Li, F.W.B. and Watson, C. (2011). *Game-based concept visualization for learning programming*, Proceedings of the third international ACM workshop on Multimedia technologies for distance learning, p. 37-42, December 01-01, 2011, Scottsdale, Arizona, USA.

Malliarakis, C., Satratzemi, M. and Xinogalos, S. (2012). *Towards the Constructive Incorporation of Serious Games Within Object Oriented Programming*, Proceedings of the European Conference on Games Based Learning, p. 301.

Malone, T. (1980). *What makes thing fun to learn? Heuristics for designing instructional computer games.* ACM.

Maragos, K. and Grigoriadou, M., (2005), "Towards the design of Intelligent Educational Gaming systems" Proceedings of Workshop on Educational Games as Intelligent learning environments, Artificial Intelligence in Education, University of Amsterdam, Amsterdam.

Maragos, K. and Grigoriadou, M. (2011). "Exploiting TALENT as a Tool for Teaching and Learning", *The International Journal of Learning*, Volume 18, Issue 1, pp.431-440.

Mayes, T. and de Freitas, S. (2004). Review of e- learning theories, frameworks and models. JISC e- learning models study report. London. The Joint Information Systems Committee. http://www.jisc.ac.uk/elp_outcomes.html.

Mayes, T. & de Freitas, S. (2006) Learning and e- learning: The role of theory. In H. Beetham & R. Sharpe. Rethinking Pedagogy for the Digital Age. London. Routledge.

Muratet, M., Torguet, P., Viallet, F. and Jessel, J.P. (2010). "Experimental feedback on Prog&Play, a serious game for programming practice", *Eurographics*, p. 1-8.

O'Kelly, J. and Gibson, P. (2006). *RoboCode & problem-based learning: A non-prescriptive approach to teaching programming*. ACM SIGCSE Bulletin, Proceedings of the 11th Annual SIGCSE Conference Bonakdarian, 38, 3 (June 2006), p. 217-221.

Piteira, M. and Haddad, S. (2011). *Innovate in Your Program Computer Class: An approach based on a serious game*. OSDOC - Open Source and Design of Communication Workshop.

Paliokas, I., Arapidis, C. and Mpimpitsos, M. (2011). *PlayLOGO 3D: A 3D interactive video game for early programming education*. Third International Conference on Games and Virtual Worlds for Serious Applications, p. 24-31.

Phelps, A, Bierre, K, and Parks, D. (2003). *MUPPETS: multi-user programming pedagogy for enhancing traditional study*, Proceeding of the 4th conference on Information technology education , October 2003, Lafayette, Indiana, USA, p. 100-105.

Pinelle, D. and N. Wong. (2008). *Heuristic evaluation for games: usability principles for video game design*. in *Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems*.

Prensky, M., (2001). *Digital Game-Based Learning*. New York: Mc Graw Hill.

Ragonis, N. and Ben-Ari, M. (2005). *A long-Term Investigation of the Comprehension by Novices*, Computer Science Education, Vol. 15, No. 3, 203-221.

Resnick, M. (2007). *All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten.* Proceedings of the SIGCHI Conference on Creativity and Cognition.

Salen, K. and Zimmerman, E. (2004). Rules of Play: Game Design Fundamentals. Cambridge, Massachusetts: The MIT Press.

Sandford, R. (2006). Teaching with COTS. Paper presented at the JISC Online Conference Innovating e-Learning 2006. March 30th.

Saltzman, M. (2000). Game Design: Secrets of the Sages, Second Edition. Indianapolis, IN: Macmillan Publishing.

Song, M. and S. Zhang, (2008). *EFM: A Model for Educational Game Design*, in *Lecture Notes in Computer Science*, S. Link, Editor. Springer US. p. 509-517.

Yusoff, A.; Crowder, R.; Gilbert, L.; Wills, G. (2009), "A Conceptual Framework for Serious Games," in *The 9th IEEE International Conference on Advanced Learning Technologies* (ICALT 2009) vol., no., pp.21-23, 15-17 July 2009 doi: 10.1109/ICALT.2009.19.

Wong, W.L. L.N.; Cuihua, S; Eduardo, C; Fei, T; Shiyamvar, B; Harishkumar, N; Hua, W; Ute, R. (2007). *Serious Video Game Effectiveness*. In ACE 07, Salzburg, Austria.: ACM.