Serdica J. Computing **11** (2017), No 3–4, 249–278

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

TOWARDS AUTOMATIC GENERATION OF SERIOUS MAZE GAMES FOR EDUCATION

Boyan Bontchev, Radina Panayotova

Abstract. Serious games based on video mazes can be easily and effectively applied for learning purposes with the goal of facilitating technology-enhanced education. In order to practice game-based learning for various curriculums, educators need software platforms for automatized construction and flexible customization of such games. This article presents an open software platform named Maze Builder built on Unity 3D, which is especially designed for automatic generation and easy modification of maze video games. We discuss the maze game design process, the platform architecture and its data model, the results obtained from the performance tests, and a practical experiment conducted with teachers using the platform for generating maze games with educational tasks embedded into maze rooms. The initial results acquired from these experiments are very positive and encouraging with regard of the usability of the Maze Builder platform by domain specialists who are not IT professionals.

ACM Computing Classification System (1998): D.2.6, D.2.10, K.8.0.
Key words: video games, maze, generation, education, Maze Builder.

1. Introduction. Playing games is one of the oldest and most popular activities of human beings. Each game represents "a type of play activity, conducted in the context of a pretended reality, in which the participant(s) try to achieve at least one arbitrary, non-trivial goal by acting in accordance with rules" [1]. Computer games open a new era of gaming by providing a fascinating media rich in excitement and fun. Video games are a new media of creative expression, which is rather different from literature and cinema: a book describes the story to the audience in the third person and presents the story as evolving over time; a movie adds an audio-visual, sensory experience to books; finally, video games add a new dimension of interactivity in a two- or three dimensional space to cinema [2]. Thus, video games allow the player to be the subject of the story [3].

The major part of all computer games consists of video games and they are played mostly for entertainment. Nevertheless, many video games are applied as an effective tool facilitating traditional activities and processes in education, vocational training, manufacturing, advertising, healthcare and rehabilitation, and many others [4]. Such video games are often referred either as serious [5] or applied [6] games in so far as they are "*produced, marketed, or used for purposes other than pure entertainment*" [7]. Most serious games are designed especially with educational intent, hence some authors define them as digital games "*in which education is the primary goal, rather than entertainment*" [8]. Although the content dimension of such games can include games not only for education but also for health, advertising in marketing, training, science and research, or production as presented in the taxonomy of Sawyer and Smith [9], for sure any serious game should enable self-controlled, active and playful learning. In contrast with entertainment games, serious games address problem solving instead of rich experiences and provide important elements of learning instead of fun [10]. As well, unlike games for fun, they make assumptions necessary for workable simulations and should reflect natural (non-perfect) communication.

Modern digital games provide high interactivity of the gameplay combined with an immersive and intriguing 3D virtual environment. All that makes video game playing a very engaging and motivating process, which naturally helps the development of various skills and abilities, such as spatial thinking and cognition, strategic skills and ways for memory enhancement [11].

These issues help serious 3D video games to be broadly applied as a tool for game-based learning (GBL), which is defined as a specific type of gameplay with well-determined learning outcomes [12]. Many practical experiments of learning with video games have proven that GBL provides a great level of learning motivation and engagement at affective, behavioral, cognitive, and sociocultural levels [13]. In fact, any digital game has some learning aspects; however, educational games intentionally embed learning content, elements of instructional design and interactions meaningful in a learning context. For example, the didactic contents of any curriculum can be easily structured in hierarchical mazes, where a section or a cycle of the maze represents a module of the curriculum [14, 15].

Serious games for education are available at a high production cost [16] and, on the other hand, there is a shortage of free and customizable platforms for the automatic creation of such games [17]. Open or free generation platforms for rapid video game construction would facilitate a massive penetration of GBL in classrooms appropriate for any specific curriculum. The present article addresses this problem by proposing a software platform for automatic generation and easy modification of educational video maze games, whereupon the maze is defined in a declarative way including a description of both its structure and properties. The platform is named Maze Builder and is built on top of one of the most popular free game engines—Unity 3D. It offers a graphical desktop interface embedded into the Unity 3D menu, which allows an import of maze assets (images, textures and video) and automatic generation of the game from a textual description of the maze graph, characteristics of nodes and links, didactic elements, educational contents and visual representation. The teacher can define easily the textual description and content of the future game—both structured in an XML document. This document is used by the platform for the generation of a specific maze game with educational purposes. The generated game may have a number of didactic elements for each room of the maze—for example, wall panels with slides, doors with questions attached to them, maps, rings, circles, and rolling balls. It can be built either as desktop executable or as a web-based resource.

The rest of the paper is structured as follows. The next section provides our motivation to construct the Maze Builder platform including a comparative analysis of other existing similar platforms and tools and, as well, a comparison of existing game engines. Section 3 presents the platform itself starting from the system requirements and going through its modular and component architecture, data model, user interface, and results obtained from performance testing. Next, section 4 discusses a practical experiment conducted with educators using the platform for generating maze games. The constructed video mazes have educational tasks allocated into maze rooms. The section presents the case study design, procedure and participants, and results acquired from the practical experiments concerning platform usability for domain specialists who are not IT professionals. Finally, the conclusions summarize the work done and outline some directions for future work.

2. Motivation.

Our motivation to conceive and construct the Maze Builder platform is based on some of the problems identified with serious video games for education and, on the other hand, on the need of a software platform for an automatized construction of such games.

2.1. Problems with Serious Video Games for Education.

Despite the great success of GBL during the last decades, serious games continue having serious problems which hamper their ubiquitous use in education. The GALA Roadmap [16] identified several such problems and open questions, as follows:

- Higher development cost—serious games are designed with a specific (mainly educational) purpose different than fun.
- Lower attractiveness compared to entertainment games.
- A problematic transition between instructional design and actual game design implementation—how game mechanics impact and interact with the learning mechanics.
- Assessment problems—no effective tracking and analyzing of the right parameters related to learners' progress (knowledge gain, reflection, and application).
- How should psychological theories be used in the design of realistic and convincing non-playing characters (NPCs)?
- How do different pedagogical paradigms relate to serious game mechanics (the question addresses the need of a reference framework)?

Next to GALA, Shapiro [18] identified top ten obstacles for applying games in the education. He has surveyed 700 teachers especially for finding and ranking the major barriers to GBL. Among all the barriers, he stressed three essential technological impediments hampering the massive application of GBL, namely

- costs of purchasing games;
- difficulties in finding video games that fit a specific curriculum;
- uncertainty about the ways of integrating games into instruction.

Both the GALA consortium and Shapiro identified three problems forming the great challenge hampering the massive use of educational games—they should not be too expensive, need to be much more attractive, and should provide good interactions between game mechanics and learning mechanics. The modern platforms for generation of serious video games for education should address these very issues.

2.2. Platforms for Generation of Educational Maze Games. Many of the latest approaches for game-based education rely on using serious games with mazes for educational purposes [14, 15, 17]. Mazes are broadly used in entertainment games and that makes them very appropriate for interactive representation of content, where the player chooses one of the several options for an action. Furthermore, mazes can be combined with quizzes, puzzles and other mini-games situated at appropriated places into the maze that makes them suitable for GBL in any learning domain [19]. The player navigates between the rooms of the maze as specified by the connecting graph designed by the educator for a given learning curriculum. While passing from one room to another, he/she has to solve didactic tasks based on his/her current outcomes.

For applying maze games for education, educators with no IT skills need software platforms for an easy and automatized construction of hierarchical video mazes for their curricula. Such platforms should be customizable, i. e., teachers should be able to customize maze nodes with their preferred content [15]. At this moment, only some rather simple tools for creation of mazes are available. Quandary allows an easy creation of Web-based 2D mazes for action games.[1] Usually, a given state (dedicated to information about a concept or situation) is presented to the player, with several possible choices (actions) to proceed within the maze (course). After selecting an option, the player moves to the resulting state of the transition graph and explores its information and set of options. Another tool especially designed for developing interactive learning contents is Qedoc Quiz Maker.[2] The tool provides a flexible playback environment that can become a quiz player, an exam revision system, a corporate learning tool, or a survey instrument. It comes with more than a hundred different question types including special question types such as mathematical problem generators. It is useful for the creation of learning games such as memory games, anagrams and mystery words. Thanks to its powerful graphical editor, the Qedoc Quiz Maker is used for creating and distributing interactive educational and training modules.

Another software tool for easy creation and customization of 3D video mazes is designed and validated in the scope of the ADAPTIMES project [20]. The tool is constructed over the Brainstorm eStudio[3] platform and is intended to help automatic construction of maze games. It is validated by video games for learning entrepreneurship but can be applied for generation of mazes in various learning domains. Educators are able to create their 3D video mazes and customize their rooms including didactic tasks and embedded mini-games by using a simple XML editor. However, the tool is not freeware and is limited only within the research project.

Table 1 provides a comparison of the three platforms for automatic creation of educational games outlined above, together with the Maze Builder tool proposed in this article. Several criteria were selected to be applied at that comparison, as follows:

- distribution of the generated game—desktop, Web-based, mobile, or console;
- type of the generated game—2D, 2.5D, or 3D;
- specific target game platform (used by the generated games);
- programming language;

[1] `http://www.halfbakedsoftware.com/quandary.php`
[2] `http://www.softpedia.com/get/Others/Home-Education/Qedoc-Quiz-Maker.shtml`
[3] `http://www.brainstorm.es/products/estudio/`

- learning resources within the game—for example slides, test questions for opening doors or for proceeding further in the maze, image puzzles, etc.;
- customization—for example adding music, colors, fonts, illumination, etc.;
- open platform—the code is available for modifications or is locked;
- price—free or paid solution.

Table 1. A comparison of existing platforms for automatic creation of educational games

Criteria / Solution	Quandary	Qedoc Quiz Maker	ADAPTIMES	Maze Builder
Distribution	Web	Desktop	Desktop	Desktop/Web/ Mobile/Console
Type	2D	2D	3D	3D
Game platform	None	None	None	Unity 3D
Programming language	None	None	Python	C#
Learning resources	slides, test questions	slides, test questions, learning games	slides, test questions, arranging images	slides, test questions, arranging 3D objects[4], hidden 3D objects
Customization	images, music, video, font	images, font, sound effects	Textures, rooms	Textures, images, music
Open code	No	No	Yes	Yes
Price	Free	Free	Licensed	Free

According to these criteria, the Maze Builder platform appears superior to the other solutions existing at the moment (Table 1). The Maze Builder solution provides distribution to different platforms, a number of learning resources and extensibility due to its open code. It makes use of the Unity 3D, which offers both a good game engine and a great community of users.

[4] Balls, circles, and rings—all of them with text and graphics.

2.3. Game Engines. A game engine is a collection of software instruments that comprise the fundamental elements common to most digital games [21]. Most commercially-produced video games are designed and developed using an existing free or commercial game engine. Hence, the game makes use of the functionalities available for the specific game engine. For this reason, it was crucially important to select a game engine appropriate for the goals defined for the Maze Builder platform regarding customization, openness, and price.

Table 2. A comparison of existing 3D game engines

Criterion / Game Engine	Distribution	Supported lang's	Community and doc's	Price	Dev. req's	Graphic quality	Extensible
Unreal engine 4[5]	Windows, Linux, Mac OS X, Xbox One, PlayStation 4, HTML5, iOS and Android	C++, Blueprints	Big	Free	High	Very high	Yes
CryEngine[6]	Windows, Linux, PlayStation 3, PlayStation 4, Wii U, Xbox 360, Xbox One, iOS and Android	C++, C#, Lua	Big	$9.90 per month	High	Very high	Yes
Ogre 3D[7]	Windows, Linux, Mac OS X	C++	Little	Free	Low	Medium	Yes
Blender[8]	Windows, Linux, Mac OS X	Python	Big	Free	Medium	Medium	Yes
Godot[9]	Windows, OSX, Linux, HTML5, iOS, Android, BB10	C++, GDscript	Little	Free	Medium	Medium	Yes

[5] https://www.unrealengine.com/en-US/what-is-unreal-engine-4
[6] https://www.cryengine.com/
[7] http://www.ogre3d.org/
[8] https://www.blender.org/
[9] https://godotengine.org/

Unity 3D[10]	Windows, OSX, Linux, Windows Phone, iOS, Android, BlackBerry 10, Tizen, Xbox 360, Xbox One, Wii U, PlayStation 3/4, PlayStation Vita, Nintendo Switch, WebGL	C#, Java-Script	Very big	Free	Me-dium	High	Yes

Table 2 represents a comparison of existing 3D game engines according to seven criteria:

- distribution mode;
- supported languages;
- community and documentation;
- price;
- developer requirements;
- graphic quality;
- extensibility.

As Table 2 reveals, Unity 3D is one of the most popular game engines. In the middle of 2013 it was used by more than two millions of developers worldwide. It has a very large list of supported platforms, some of which are: Windows, OSX, Linux, Windows Phone, iOS, Android, BlackBerry 10, Tizen, Xbox 360, Xbox One, Wii U, PlayStation 3, PlayStation 4, PlayStation Vita, and Nintendo Switch. It also supports WebGL [22], which allows running the game in the browser. On the other hand, Unity has one of the lowest system requirements for developers compared to its competitors.

The Unity 3D game engine supports assets from major 3D applications like 3ds Max, Maya, Softimage, CINEMA 4D, Blender and more, meaning there are no real restrictions to the type of file formats that it supports. After the release of Unity 4.3, it also has native 2D capabilities, supporting sprites

[10] `https://unity3d.com/`

and 2D physics, making it a great game engine to use for the development of 2D games. Unity has a large asset library where a wide variety of assets can be downloaded for free or purchased [23]. As well, it provides an exhaustive documentation, where everything is given a full description supplied by a number of examples as well as video and text tutorials and live training sessions.

Unity's modular system and usability allow for quickly developing a prototype of an idea. Its free version, together with all the other merits (Table 2), formed our decision to use it for the platform development of Maze Builder.

3. The Maze Builder Platform. The project goal is to create an open software platform for the construction of smart and customizable 3D video maze games with intelligent virtual players (i. e., smart non-playing characters) and to validate it by practical experiments with the construction of game prototypes in the context of a socially significant, complex and content-rich teaching domain. The section represents an overview of the requirements demanded the platform, main design considerations, and some results from the testing phase.

3.1. Analysis of Platform Requirements. The analysis of the platform requirements is one of the mandatory conditions for both high-quality design and implementation of the software. With this purpose, we have reviewed in detail the conceptual model, functional and quality requirements for implementing the Maze Builder.

The 3D maze can be represented by a graph. The graph corresponding to the maze has some restrictions based on the fact that it represents a maze of square, equally large rooms. The rooms are the nodes of the graph, whereas the doors in the rooms correspond to the arcs. During the analysis phase, the following restrictions of the graph construction were identified:

- The maze connectivity graph must be planar.
- Each node (i. e., room) of the maze can have a maximum of four doors.
- Cycles are possible within the maze graph.
- A pair of incoming and outgoing doors is shown as a lack of door.
- Each door can be unlocked by correctly answering a question.

- Each question may have text and raster graphic.
- Walls have maximum one door each.
- Walls have maximum two slides each.
- Each slide may have text and/or raster graphic in JPG, PNG, TIFF or BMP format.
- Each room can have a game with balls, circles and/or rings (explained further in the article).
- Each room can have zero or *N* hidden objects.
- Each room can have textures for the walls, ceiling, and the floor.
- Each room can have no more than one map laid over the floor.
- Each room can have no more than one audio file for playing while the player is inside it (played once or in repeating mode).
- The maze should have one starting room.

On the other hand, we have identified the following quality requirements for the platform:

- Portability of generated 3D maze games to different target platforms.
- Good usability—the Maze Builder platform is aimed to be used by non-programmers and people with no experience with game design so the user interface is designed to be as simple as possible and the process is highly automated.
- High extensibility in terms of easy adding of new game objects and script components. The platform is built as a free open source extension to the Unity 3D editor so the code is available for future extensions.
- Easy maintenance—the Maze Builder platform is not being implemented for a specific version of the Unity 3D game engine so it should work well with a newer version of the game engine with only relatively small or no modifications.
- High performance—the Maze Builder platform relies on the performance qualities of the Unity 3D game engine.

The high performance results also from the fact that no specific implementation of memory allocation or garbage collection needs to be added. Still, the scripts contained in the Maze Builder platform need to be efficient and free of any large time/memory consuming code.

3.2. Platform Design. Maze Builder is going to allow an easy creation of smart educational video games by non-IT specialists. For this purpose, it allows automatic generation of educational mazes using a Unity3D-based maze builder. Fig. 1 presents the main building blocks of the Maze Builder platform for the construction of 3D video games for education. Maze Builder is designed as a pluggable package for the Unity3D game editor. After importing the package, a Maze Builder menu appears in the Unity3D game editor allowing the creator of an educational maze to import maze assets (pictures, textures, and audio files) and a maze description in XML valid towards a specific XML Schema (XSD). The XML maze description defines the maze, learning content and didactic tasks, which are controlled by game managers. The maze creator may use Unity3D editor to arrange the game objects for the didactic tasks, to play the game for checking if everything is as expected and, finally, to generate the maze game for given platform (PC, console or mobile device).

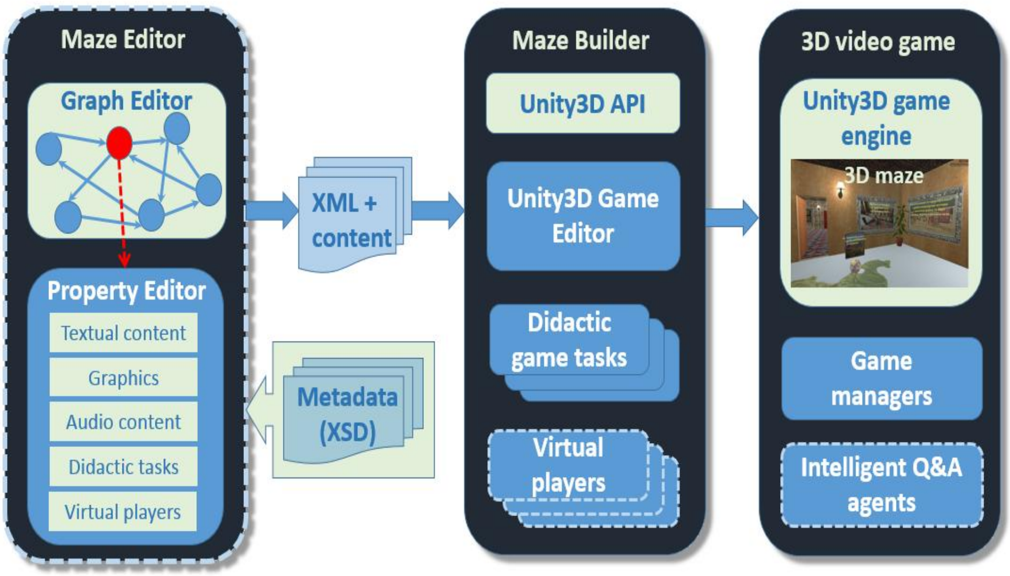

Fig. 1. The Maze Builder platform for construction of 3D video games for education [24]

Beside generation of mazes, Maze Builder will make use of three other additional innovative features, as follows:

- a metadata-driven approach for maze editing supported by visual construction of the maze graph and declarative game description and semantically structured representation of artifacts;
- virtual players (i. e., non-playing characters, or NPCs) helping the player in solving didactic tasks by giving some hints and answering his/her questions regarding the learning domain;
- an intelligent question and answer (Q&A) agent for providing virtual agents with appropriate answers to player questions during game sessions.

The blocks implementing these features are the Maze Editor, the virtual players, and the intelligent Q&A agent. They are planned to be developed for the next versions of the APOGEE platform [24] and, therefore, are shown in the figure with dotted lines.

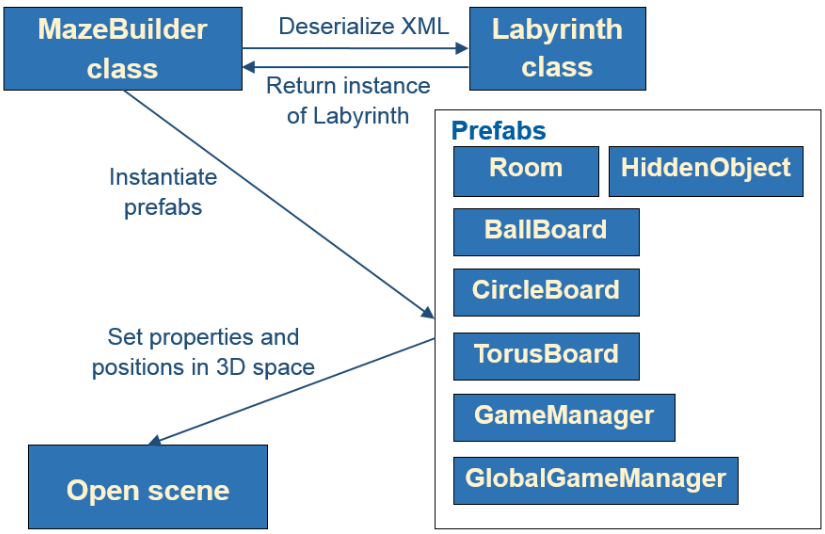

Fig. 2. Basic software architecture of the Maze Builder platform

The basic software architecture of the Maze Builder platform is represented in Fig. 2. The MazeBuilder class is an extension of the Unity 3D editor, which uses the Labyrinth class containing a class structure matching the XML document to deserialize the XML document. The result of that is an object, which is an instance of the Labyrinth class. The MazeBuilder class goes through all properties of this Labyrinth object and instantiates different prefabs[11] (each prefab [25] matching a certain object in the maze), sets properties of the prefab instances and positions the instances in the 3D space of the open scene in Unity. The Labyrinth class applies many prefabs and their respective scripts for creating maze rooms, doors between rooms, learning boards and hidden objects for each room, objects for the didactic tasks, and game managers.

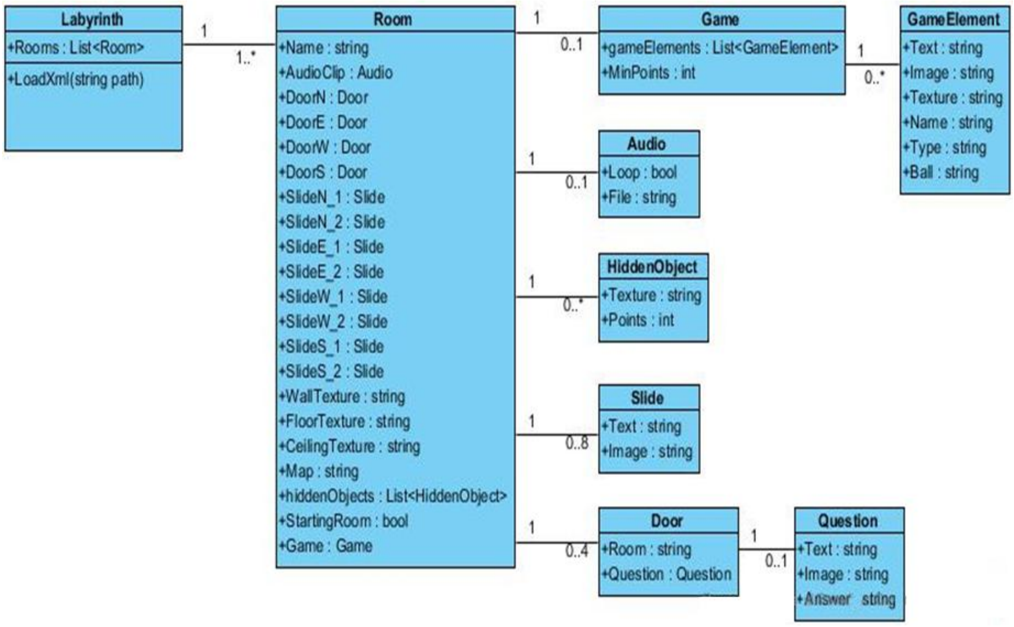

Fig. 3. Maze Builder data model

Fig. 3 depicts the main classes of the Maze Builder data model. It comprises classes used for XML serialization. The root class is Labyrinth and it corresponds to the whole maze. It contains a collection of rooms. It has one method LoadXml, which uses the XmlSerializer class to read the XML. The

[11] Unity3D uses the term prefab for a type of asset being a reusable game object stored in project view, which can be inserted multiple times into any game scene.

Room class is used for instantiating each room of the maze and may contain a name, textures, a map, an audio to be played while the player is in the room, up to four doors, up to eight slides, zero or many hidden objects (bringing points when discovered and clicked by the player), and a didactic task (game) to be solved. The Game class represents a puzzle game with balls and has as attributes a collection of game elements, a minimal number of points needed to activate the game, and a collection of GameElement objects being either balls, circles or rings. The puzzle game consists in rolling the balls to their proper positions over given circles on the floor or inside specific rings. The player should read a description over each ball and find where to roll the ball to, i. e., to a given area (circle) on the map or to an appropriate ring. When the target is matched, the ball changes its color and becomes frozen at that position. After the puzzle (i. e., didactic task) is solved, the game manager shows a question for unlocking the door when the player goes close to that door. The door will be unlocked by giving the right answer to that question. If the answer is wrong, then the player should read carefully the slides in that room and try again.

3.3. Platform Testing. The platform construction included full functional and quality testing stages. The functional testing was conducted by several test scenarios implemented in a separate way and, next, altogether by the sample XML document describing the creation of a two-room maze with sample slides, textures, audio, and didactic tasks of all the types. The XML document was imported and the maze game was generated and played as expected.

Next, quality tests were performed according to predefined test scenarios. The Maze Builder platform was tested with input resources such as various image/audio formats. Especially for the performance tests, a content-rich 6-room maze was designed. Its XML description was used for a performance testing of the Maze Builder on a developer laptop under Windows 10. The following time delays were recorded for the different activities:

- importing of Maze Builder Unity Package—4 minutes;
- importing of 4 audio files with total volume 28MB—176 seconds;
- importing of 86 images with total volume of 212MB—65 seconds;

- importing of an XML document describing a 6-room maze and generation of the maze—20 seconds;
- building a desktop standalone (executable)—3 minutes 20 seconds;
- building a WebGL game distribution—12 minutes.

Fig. 4 presents the times needed to perform the activities using Maze Builder, together with the times needed for arranging the 3D assets within the generated game. The figure allows comparing timings for user actions for arranging assets in a 6-room maze with those for machine activity (e. g., compiling and generating code). Obviously, the time for arranging assets (circa four hours) is much longer than those for machine activity.

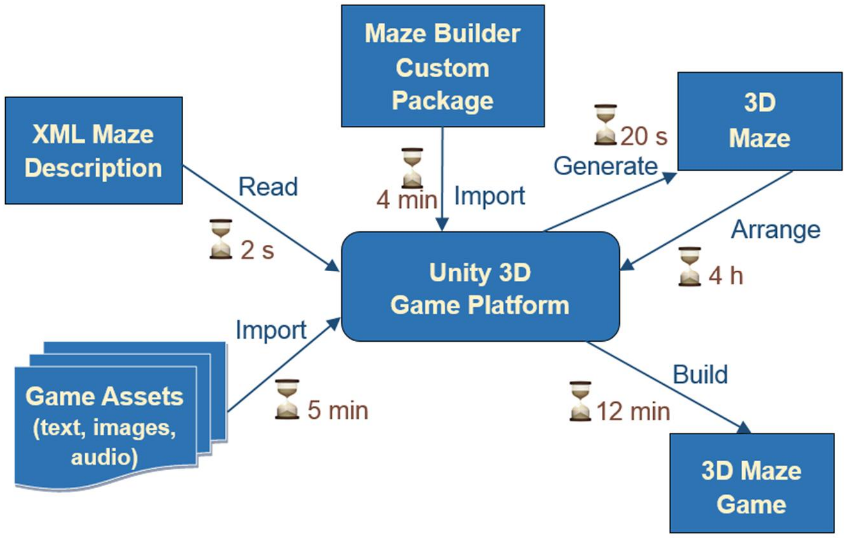

Fig. 4. Timing of the game creation process for a 6-rooms maze using the Maze Builder platform

Beside high performance, quality tests proved the expected portability, high extensibility, and easy maintenance of the Maze Builder platform. For validating platform usability, a separate case study was conducted.

4. Case Study. The usability of the Maze Builder solution was validated by creating a real educational game rich in didactic content and tasks.

The case study followed a scenario for measuring how easy and effectively educators can use the platform to generate educational games. An additional case study was conducted for assessing the playability of the generated educational games [15].

4.1. Case Study Design. In order to test and validate the usability of the platform, we created an experimental 3D video game supporting education in Bulgarian ethnography [15]. The game is focused on carpet manufacturing in Bulgaria since the 17th century to modern days. The maze of the game consists of six rooms interconnected by doors as represented in the maze graph shown in Fig. 5. Though the fact that each arc of the graph is bidirectional after its door is unlocked, the maze graph represents the directions of the initial crawl of the maze, i. e., while unlocking the doors. After the doors are unlocked, the player can open or close them from any side of the wall, therefore he/she can crawl the maze in both directions for each door. The maze has two cycles:

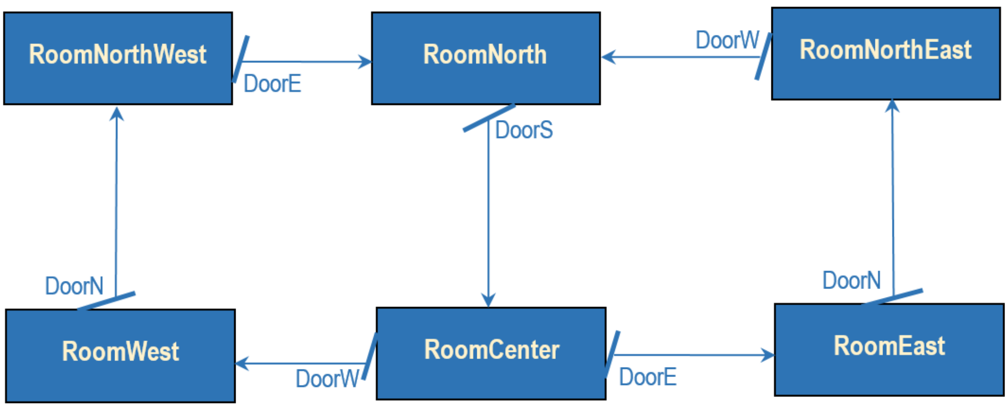

Fig. 5. Graph of the maze

- Cycle A, explaining the fabrication of carpets in the town of Chiprovtsi, Bulgaria: RoomCenter → RoomWest → RoomNorthWest → RoomNorth → RoomCenter;
- Cycle B, presenting the fabrication of carpets in Kotel: RoomCenter → RoomEast → RoomNorthEast → RoomNorth → RoomCenter.

All six rooms except the central room contain a collection of hidden objects as well as didactic tasks presented as ball rolling mini-games as explained in [15]. Fig. 6 provides a partial view of the XML description of the 3D maze for the development of the carpet handicraft in Bulgaria. The text marked in blue on the right-hand side of the figure presents the question for unlocking the east door DoorE of the room.

A screenshot of the generated game about the carpet handicraft in Bulgaria is given in Fig. 7. The game is playable in a Web browser with an installed Unity plugin at `http://adaptimes.eu/carpetgame/`, or as a downloadable desktop game. The player has to navigate the maze through the w-a-s-d keys or the arrows and use the mouse to rotate. He/she has to find the hidden objects and answer the question near the door to unlock it by approaching the question and placing the cursor in front of it. If there are balls in the room, they should be rolled to their respective positions before unlocking the doors.

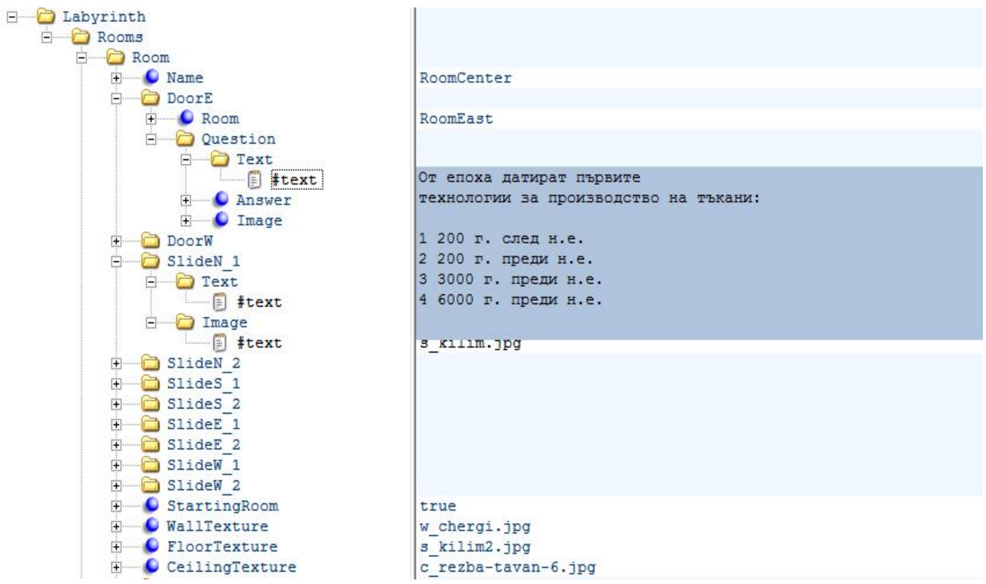

Fig. 6. XML description of the 3D maze for the development of carpet handicraft in Bulgaria

Fig. 7. A screenshot of the generated game about the carpet handicraft in Bulgaria

For assessing the platform usability of Maze Builder, a special questionnaire was developed. It is given in an Appendix at the end of the article and consists of two sections:

A. the relevance of the educational video maze games;
B. usability of the platform for generating educational video maze games.

The questionnaire applies a 5-level Likert scale (Definitely yes—5, Rather yes—4, Cannot judge—3, Rather no—2, Definitely no—1). It was administered to educators to study usability issues of Maze Builder. Another questionnaire was created for assessing the playability of the generated maze game and was administered to the players after the end of the game session [15]; however, it is out of the scope of this article.

4.2. Procedure and Participants. The practical experiment for assessing the platform usability followed a six-step procedure:

- The participating educators saw a live demonstration of generation of a maze game and arrangements of didactic objects embedded into it.
- The educators watched a video about the maze-creation process available at `https://www.youtube.com/watch?v=3IBqYooKwQg` (2:39 min.).

- The creation process of educational games by means of the Maze Builder was demonstrated individually to every participant in a session of circa 10 minutes.
- The educators used the prepared game assets and XML document (Fig. 6) to generate their 3D maze for the development of carpet handicraft in Bulgaria. They were free to change the XML game description and/or the game assets in order to see the effect of their modification. There was no time limit for this session.
- The participants were free to play the generated game either in the Unity3D editor or as a desktop game, without any time limit.
- All educators were asked to fill in the 12-item questionnaire about the usability of the Maze Builder platform (see the Appendix).

17 educators of high schools and universities took part of the study. They were aged between 26 and 58 years (M=39). Thirteen of them were women and four were men. All participated entirely anonymously and voluntarily.

4.3. Results. This section presents the most interesting results found by means of the questionnaire used for the study of platform usability.

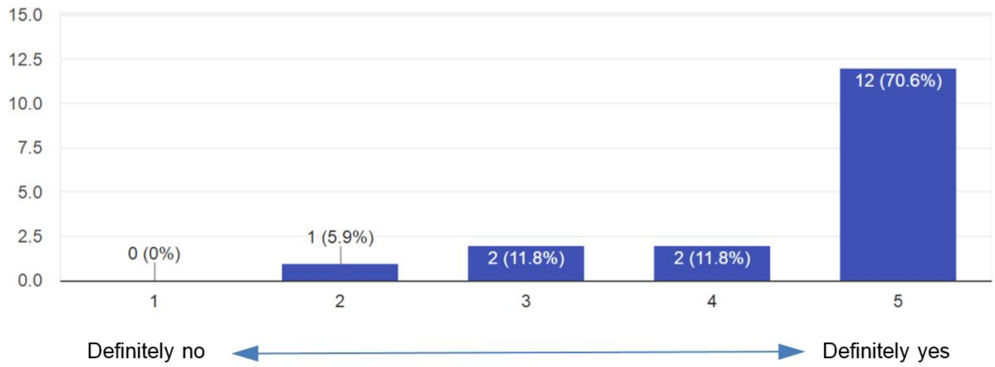

Fig. 8. Answers to the question "Do you think that educational video maze games are an effective tool facilitating education?"

Fig. 8 presents graphically the answers to the question "Do you think that educational video maze games are an effective tool facilitating education?"

(the answers "Definitely no" are coded as 1, "Definitely yes" as 5). More than 70% of the participants were convinced that such maze games can facilitate the educational process in an effective way. Next, Fig. 9 reveals the answers to the question about the idea to use text in XML format for describing video maze games. The results found show that more than a half of the educators are definitely convinced XML is very good for maze descriptions; moreover, there are no educators claiming the opposite.

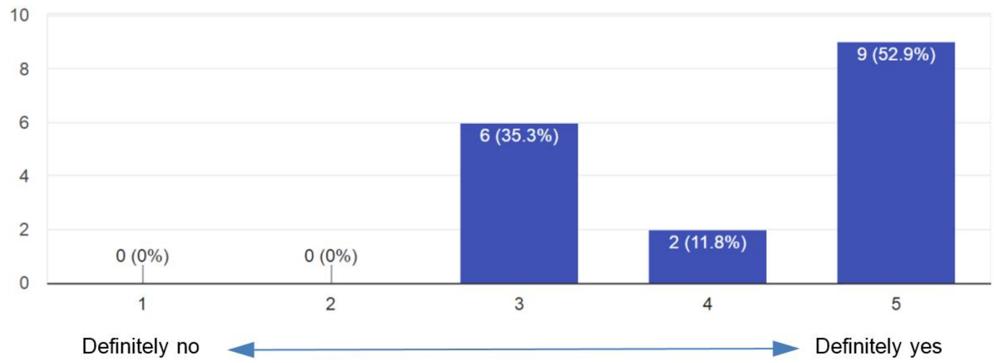

Fig. 9. Answers to the question "The idea to use text in XML format for describing video maze games is very good and has future."

The majority of the participating volunteers found the Maze Builder platform for generating educational video maze games easy and pleasant to use (Fig. 10). Only one of them claimed the opposite and 23.5% reported "Cannot judge". The question "Declarative description of a game in XML format is easy and takes much less time than gathering and assembling didactic materials for the same (texts, graphics, and audio)." is a little bit provocative as far as the participants were not asked to collect and assemble any didactic materials—they received all the assets ready for import (Fig. 11). Nevertheless, the great majority of them find that declarative XML game description takes much less time and effort than the preparation of game assets, which was practically confirmed by the experimental test results (Fig. 4).

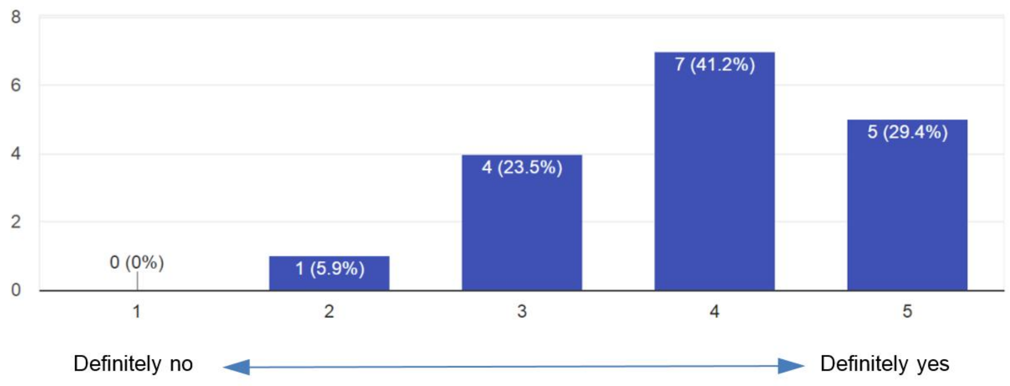

Fig. 10. Answers to the question "The platform for generating educational video maze games is easy and pleasant to use."

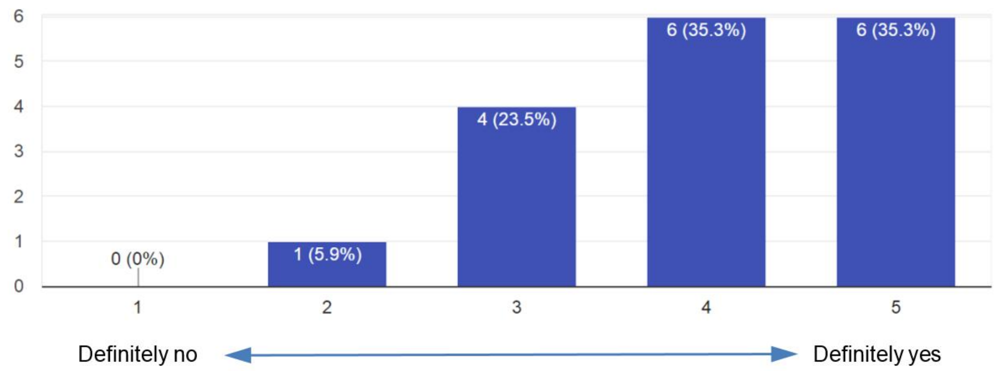

Fig. 11. Answers to the question "Declarative description of a game in XML format is easy and takes much less time than gathering and assembling didactic materials for the same (texts, graphics and audio)."

Beside the encouraging results reported so far, we also got some skeptical feedback. It comes with the fact that the educators lack skills in working with visual editors such as that of Unity3D. This drawback makes more than half of the educators participating in the study rather suspicious in claiming that non-IT specialists would be able to manually move game objects like balls, circles, and rings to the appropriate places in the Unity3D editor

(Fig. 12). Therefore, educators without any experience in Unity3D should receive an introductory course in its graphics editor.

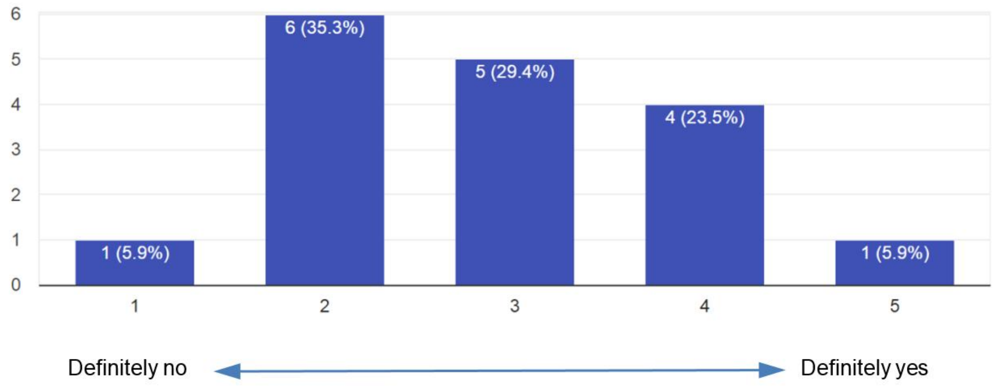

Fig. 12. Answers to the question "After generation of the maze, the learning objects are automatically positioned in the rooms by generator and have to be manually moved to the appropriate places in the Unity 3D editor. Do you think that non-IT specialists could move them and position them where they want in the Unity 3D editor?"

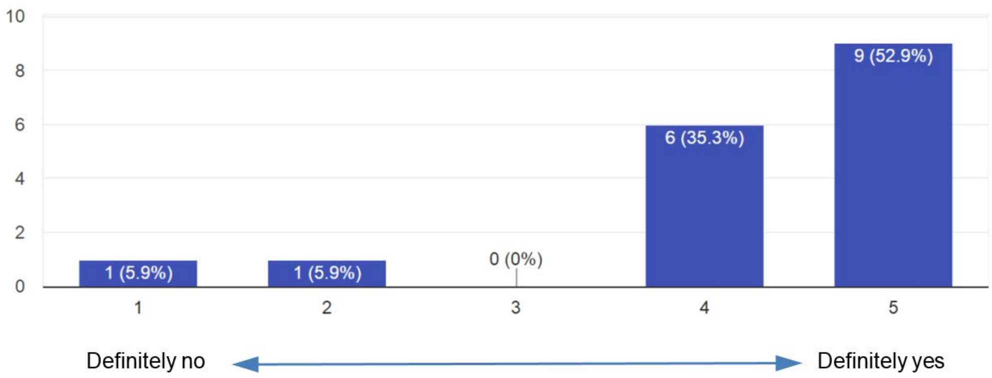

Fig. 13. Answers to the question "The utilized learning tasks such as moving objects to certain positions or next to other objects, answering test questions for unlocking doors to other rooms and finding hidden objects, are very appropriate for educational games and significantly facilitate education."

Finally, Fig. 13 provides the feedback about the appropriateness of the learning tasks employed, such as moving objects to certain positions or next to other objects, answering test questions for unlocking doors to other rooms and finding hidden objects. We were pleased to find out that almost 90% of the educators like these tasks; as well, they regard them as very appropriate for educational games and significantly facilitating the learning process. Of course, these didactic tasks can be extended by additional types of tasks, e. g., by 2D puzzles.

5. Conclusions. As stated in the introduction, serious games for education come at a relatively high production cost partially because such games have to be constructed on demand and be customized according to the learning objectives of a given curriculum [17]. This technological barrier, together with the other obstacles to serious games identified by the GALA consortium [16] and Shapiro [18], can be eliminated by offering free platforms for an automated creation of educational video games on behalf of non-IT specialists such as teachers, instructors, and pedagogues. In order to provide good performance and high usability, such platform should rely on free, powerful and portable game engines.

The article presented a novel, open software platform for rapid and straightforward construction and modification of educational video maze games—Maze Builder—built on top of one of the most popular free game engines—Unity 3D. It offers a graphical desktop interface embedded into the Unity 3D menu, which allows an import of maze assets and automatic generation of the game from an XML description of the maze together with didactic elements, educational contents, and their visual representation. The results obtained from the conducted case study revealed a high usability of the Maze Builder platform.

As a direction of our future research and practical work, we are going to address three important groups of issues, as follows:

- Development of various game assets for an enhanced player experience—3D prefab objects (e. g., torches instead of lamps, stony doors instead of wooden doors, etc.), sounds for events like unlock/open/close a door, text decorations specified in the XML document, scripts for a flexible behavior control—e. g., for controlling the visibility of texts over pictures, an interactive map showing the current player location in the maze, and others.

- Support of more educational features—e. g., puzzles or assessment tests for unlocking a door, more types of embedded mini-games (e. g., sorting and distributing hidden objects by placing them on specific places), logging of gameplay of an individual learner with contents of the log files controlled by the maze XML definition (like which indicators to log, how often, etc.).
- Builds not only of desktop and Web-based games but of games for mobile devices as well.

Next to the platform development, we plan using the platform for other case studies, with a greater number of participants. Such case studies will address other maze games in various teaching domains. The mazes might apply other applications of the ball games—e. g., a game ordering the balls in a line. As well, platform users should be able to save generated and modified game back to an XML document—it is possible to manually arrange the objects in the generated maze (for example hidden objects or balls, circles and rings for the mini-games), to position them elsewhere, or to manually add prefabs (like whole rooms for example). If the game is regenerated from the original XML document, the moved objects would be placed back to their default positions and the manually added ones would not be recreated. It would be a nice feature to save the customized maze game back to an XML document so that the next import of the XML would generate the customized game. In general, all the new functionalities, which we will add to the future versions of the Maze Builder platform, will firstly be studied by quantitative surveys conducted with both students and their teachers, within the scope of the APOGEE research project [24].

Acknowledgments. The research leading to these results has received funding from the APOGEE project funded by the Bulgarian National Science Fund, Grant Agreement No DN12/7/2017.

REFERENCES

[1] Adams E. Fundamentals of Game Design. 2nd Edition, New Riders, 2010.

[2] Lee T. Designing game narrative. Hitbox Team, 24 October 2013, http://hitboxteam.com/designing-game-narrative, 29 October 2018.

[3] Wolf M. J. P. Building Imaginary Worlds: The Theory and History of Subcreation. Taylor & Francis, 2012.

[4] Essential Facts About the Computer and Video Game Industry. ESA, 2015. http://www.theesa.com/wp-content/uploads/2015/04/ESA-Essential-Facts-2015.pdf, 29 October 2018.

[5] Abt C. C. Serious games. University press of America, 1987.

[6] Schmidt R., K. Emmerich, B. Schmidt. Applied games–in search of a new definition. In: Proceedings of the 14th International Conference ICEC 2015, Trondheim, Norway, 2015. *LNCS*, **9353** (2015), 100–111.

[7] Egenfeldt-Nielsen S., J. H. Smith, S. P. Tosca. Understanding Video Games: The Essential Introduction. Routledge, 2008.

[8] Michael D. R., S. Chen. Serious Games: Games That Educate, Train, and Inform. Thomson Course Technology, 2006.

[9] Sawyer B., P. Smith. Serious games taxonomy. Presented at Serious Games Summit, 2008.

[10] Susi T., M. Johannesson, P. Backlund. Serious Games—An Overview. Technical Report HS-IKI-TR-07-001, School of Humanities and Informatics, Univ. of Skövde, Sweden, 2007.

[11] Spence I., J. Feng. Video games and spatial cognition. *Review of General Psychology*, **14** (2010), No 2, 92–104.

[12] Shaffer D. W., K. R. Squire, R. Halverson, J. P. Gee. Video games and the future of learning. *Phi Delta Kappan*, **87** (2005), No 2, 105–111.

[13] Plass J. L., B. D. Homer, C. K. Kinzer. Foundations of game-based learning. *Educational Psychologist*, **50** (2015), No 4, 258–283.

[14] Bontchev B., D. Paneva-Marinova, L. Draganov. Educational Video Games for Bulgarian Orthodox Iconography. In: Proceedings of 9th Annual International Conference of Education, Research and Innovation, Seville, Spain, 2016, 1679–1688. DOI: 10.21125/iceri.2016.1374.

[15] Bontchev B., R. Panayotova. Generation of Educational 3D Maze Games for Carpet Handicraft in Bulgaria. *Digital Presentation and Preservation of Cultural and Scientific Heritage*, **7** (2017), 41–52.

[16] GALA. GALA Roadmap. Deliverable D1.6, Ver. 2, November 2011.

[17] Bontchev B. Customizable 3D video games as educational software. In: Proceedings of 7th International Conference on Education and New Learning Technologies EDULEARN'15, Barcelona, Spain, 2015, 6943–6950.

[18] Shapiro J. Games in the Classroom: Overcoming the Obstacles. KQED, 2014. `https://ww2.kqed.org/mindshift/2014/09/12/games-in-the-classroom-overcoming-the-obstacles/`, 29 October 2018.

[19] Bontchev B. P., D. Vassileva. Affect-based adaptation of an applied video game for educational purposes. *Interactive Technology and Smart Education*, **14** (2017), No 1, 31–49. DOI: 10.1108/ITSE-07-2016-0023.

[20] Bontchev B. Deliverable D10: Evaluation of the second field trials and final analysis of adaptivity effectiveness. ADAPTIMES: adaptive player-centric serious video games, 7th Framework Programme, 2016. `http://adaptimes.eu/pdf/ADAPTIMES-D10-ver1.0.pdf`, 29 October 2018.

[21] Kelly C. Programming 2D Games, CRC Press, 2012.

[22] Leung C., A. Salga. Enabling WebGL. In: Proceedings of the 19th International Conference on World wide web, Raleigh, North Carolina, USA, 2010, ACM, 1369–1370. DOI: 10.1145/1772690.1772933.

[23] Unity, Source 2, Unreal Engine 4, or CryENGINE—Which Game Engine Should I Choose? Pluralsight, 5 March 2015. `https://www.pluralsight.com/blog/film-games/unity-udk-cryengine-game-engine-choose`, 29 October 2018.

[24] APOGEE Web site. `http://www.apogee.online/index-en.html`, 29 October 2018.

[25] Unity User Manual. Version 2017.1. `https://docs.unity3d.com/2017.1/Documentation/Manual/UnityManual.html`, 29 October 2018.

Appendix: Questionnaire used for the study of platform usability

A. Relevance of the educational video maze games

1. Do you think that educational video maze games are an effective tool facilitating education?

2. Which age groups are educational video maze games appropriate for?

- from 3 to 6 years
- from 7 to 11 years
- from 12 to 18 years
- from 19 to 25 years
- above 25 years

3. Which subject areas are most suitable for educational video maze games?

- Natural sciences (physics, chemistry, biology)
- Social sciences (economics, law, archeology)
- Applied sciences (informatics, energy, agronomy)
- Formal sciences (mathematical analysis, algebra, geometry)

4. Educational video maze games for a particular subject (discipline) of learning (type checkbox question):

- should provide content from the curriculum for the age of the learner

- should provide content that complements the curriculum for the age of the learner
- should provide content not directly related the curriculum for the age of the learner

5. Educational video maze games are best played (type checkbox question):

- during the learning itself in the classroom
- after the learning in the classroom under the supervision of the teacher
- outside of the learning in the classroom without the supervision of the teacher

B. Usability of the platform for generating educational video maze games

6. The idea to use text in XML format for describing video maze games is very good and has future.

7. The platform for generating educational video maze games is easy and pleasant to use.

8. Declarative description of a game in XML format is easy and takes much less time than gathering and assembling didactic materials for the same (texts, graphics an audio).

9. Declarative description of a game in XML format should be assisted by a specialized text editor (based on XML schema) and a validator of this XML schema.

10. After generation of the maze, the learning objects are automatically positioned in the rooms by the generator and have to be manually moved to the appropriate places in the Unity 3D editor. Do you think that non-IT specialists could move them and position them where they want in the Unity 3D editor?

11. The utilized learning tasks such as moving objects to certain positions or next to other objects, answering test questions for unlocking doors to other rooms and finding hidden objects, are very appropriate for educational games and significantly facilitate education.

12. What other learning tasks would suggest for embedding in the rooms of the maze? (Free text answer)

Boyan Bontchev
Department of Software Technologies
Faculty of Mathematics and Informatics
St. Kliment Ohridski University of Sofia
5, J. Bourchier Blvd
1164 Sofia, Bulgaria
e-mail: `bbontchev@fmi.uni-sofia.bg`

Radina Panayotova
Club 45 Ltd.
Sofia, Bulgaria
e-mail: `radinai@gmail.com`

Received October 20, 2017
Final Accepted May 15, 2018