# Comparison of Game Engines for Serious Games

Sanja Pavkov, Ivona Franković, Nataša Hoić-Božić
Department of Informatics, University of Rijeka, Rijeka, Croatia
spavkov@student.uniri.hr
ifrankovic@inf.uniri.hr
natasah@inf.uniri.hr

***Abstract – Serious educational games are specially designed computer games which are used in an educational setting, in other words, they are interactive competitive lessons with defined learning outcomes which allow students to have fun during learning. The importance of serious games in contemporary educational practice is increasing. Applying serious games in teaching, students facilitate the learning process, adopt new skills and abilities, show more interest in learning, are more focused and more active in a class, and better understand and apply lessons learned. The complexity of serious games requires large efforts for their development. For the development of serious games teachers mostly use commercial game engines. One of the important parts of game development is a selection of appropriate development tool. Due to the range of available tools the choice of platforms for serious games is a challenge, whose selection often has considerably different goals and technical requirements depending on context and usage. The aim of this paper is to propose criteria which should be considered before selecting a game engine for serious games and shows results of a comparison of evaluated most popular game engines. Some recommendations for teachers about choosing the most suitable game engines for serious games development are also presented.***

***Keywords: serious games, game engine, developing serious games***

## I. INTRODUCTION

The 21st century is marked by the use of information technology like Internet, computers, laptops, tablets, and other smart devices to almost all areas of human activity. We are surrounded by technology from all sides, from the jobs, government and public administration, schools, universities, hospitals, banks, through to our homes. Since the computer has become an indispensable part of our everyday life, its use in education is increasing and more important. Evaluation and monitoring of the impact of computer-assisted learning have shown that the combination of learning the traditional way and learning using computers is good and useful and provides a better understanding and memorized material. As a consequence, the need arose for developing new skills to prepare students for new jobs and technologies [1]. According to Prensky technology is important to live, survive and thrive in the 21st century. He points out that youth today communicate, search information and socialize differently than their predecessors [2]. Kids of the digital age are generations of students who are familiar with digital technology and using digital devices from an early age. Information and communication technology (ICT) has almost become a native language to them or a language with which they communicate, express themselves, understand, and perceive the world around them.

Today is necessary to have a new set of skills to succeed in learning, working and living. Those competencies are beyond ICT literacy and include communication, collaboration, social and cultural skills, creativity, critical thinking, problem-solving, productivity, flexibility, risk-taking, conflict management, and a sense of initiative. They should be developed by everyone, from primary school to lifelong students. For the acquisition of those skills, the effectiveness of adopting active learning methodologies is widely acknowledged; they involve activities that accentuate the development of learner skills. Active learning enhances students engagement in learning tasks and enables them to deal with new challenges, solve problems, and adapt to changes in technology and knowledge [3].

Serious games are becoming more and more popular form of active learning. Serious games are specially designed computer games which are used in an educational setting. They enable the learner to have some control of the game activity and engage in interaction [4]. The complexity of serious games requires large efforts for their development and needs a lot of planning. One of the important parts of game development is a selection of appropriate development tool or game engine. Teachers mostly use a commercial engine for serious games design and it should be chosen to provide developing the game with all necessary education goals set for students.

This paper proposes criteria which should be considered before selecting a game engine for serious games. The second part introduces the serious games, while the third part is about game engine. The fourth part describes criteria and provides comparison results of evaluated game engines. In the last, fifth part we draw the conclusion.

## II. SERIOUS GAMES

Serious games in teaching and education encourage students to participate in the learning process. It is important that these games are stimulating and motivating and that they develop students' creativity [4]. The game in education can be used in all stages of the teaching process, from the introductory part as a motivation for students to the evaluation and formative or summative assessment of students. Serious games allow students to complete tasks, acquire knowledge and reasoning of certain principles while having fun. They represent a simulation that includes the challenge of solving a particular task important for students [5]. Important elements that contribute to the educational values of the game are stimuli, fantasy, challenge, and curiosity.

Games in general can be classified into seven

categories: adventure, role-playing games, shooting games, simulation, puzzles, strategy and sports games [3]. Games in education are: quiz games, word games, puzzle games with branching scenarios, various simulations with the tasks, simulation environments, simulations that require personal answers toy and game playing role [5].

Serious Games are primarily used for learning and adopting the learning content in a fun way aiming to increase student interest and motivation. During the development of the game, it is necessary to pay attention that all the learning outcomes are fulfilled. Every serious game has to have well-defined learning objectives and promote the development of important skills in order to increase cognitive and intellectual abilities of students.

The process of serious games creation is based on the methodology SADDIE which include **S**pecification of the game, **A**nalysis, **D**esign, **D**evelopment, **I**mplementation, and **E**valuation [3]. Creating a didactic computer game requires choosing an appropriate tool or game engine during the analysis phase. There are a number of different types of programming tools and several different alternative systems for creating computer games, such as programming languages, multimedia development environments, environments designed specifically for game development, development tools for e-learning software and the modding environment (some gaming environments come with an additional creation engine that allows the development of extensions to the game such as new storylines known as "modding") [6].

## III.GAME ENGINES

Today there are many types of software tools for developing computer games, so it is a challenging task to choose the appropriate game engine for educational games. There are many dilemmas which tool to choose because they all have strengths as well as problems. Some of them are fast for development, some may have performance issues, some require a programming skills, and some come with a developer-friendly interface [7].

For the purposes of this study, we evaluated five different game engines, these are respectively: Adventure Game Studio, Construct 2, e-Adventure, GameMaker: Studio, Phaser Editor. Each of these tools is downloaded, installed on the computer and used for game development in order to choose those engines in which 2D adventure game can be realized. Below is an overview of the examined tools for creating computer games with general information and a brief description of each of them.

### *A. Adventure Game studio*

Adventure Game Studio (AGS) engine is a Windows-based integrated development environment (IDE 2) which allows users to create 2D "point and click" adventure games. It primary target group is not educators, although can potentially be used to create educational games. Among the biggest disadvantages to the development of educational games using this engine are: his games cannot easily be integrated with Learning Management Systems (LMS) or inserted in a SCORM package and does not have educational features to allow assessment of student performance [8] [9]. On the other hand, Adventure Game Studio supports high-resolution graphics, has a fully integrated audio and video. Games created with it can be run on multiple operating systems, supports the use of thousands of sprite's (sprite is a two-dimensional bitmap graphic that is integrated into a larger scene, it can either be a static image or an animated graphic that plays a specific role), creating hundreds of rooms, an unlimited number of characters, dialogue, GUI and other interface elements [10].

### *B. Construct 2*

Programming tool Construct 2 is HTML5 tool designed specifically for 2D games and provides game development without coding to all users regardless of previous knowledge of programming. Currently is available installation on a Windows platform only but games created with it can run on a very large number of different platforms which is a big advantage of this tool. It offers a simple Drag-and-Drop (DnD) interface, is easy and intuitive to use, and it is intended primarily for people who have no prior experience with programming [11].

### *C. e-Adventure*

The e-Adventure engine has emerged within the research project at the University of Madrid (Universidad Complutense de Madrid) in Spain with an aim to facilitating the integration of educational and simulation games in the educational [12]. The tool is specially designed to create a Point & Click adventure type of educational games [13]. The three main objectives of development e-Adventure tool are: reducing the development cost for serious games, the inclusion of educational specific features in a game engine and games integration with existing educational materials in virtual learning environments

The platform is composed of two applications - a game authoring editor (for creating the educational games) and a game engine (for execution of the games). The editor is completely instructor oriented, it does not require any technical background or programming skills to be used [14]. Through a simple and intuitive graphical interface tool, it enables the creation of games by creating scenes, objects, characters, dialogue and conditions that define the logic of the action in the game. The e-Adventure supports the creation of two different types of adventure games: third-person and first-person games [13].

Using this engine, teachers can create their own educational video games or adapt some existing popular games to add educational value. According to that, e-Adventure provides educational features that other popular game engines usually don't. The two most important educational features of this tool are the assessment reports and the integration with popular LMS (e.g. Moodle) [15].

As an example of an application in education, the e-Adventure platform is using for making educational computer games at the Faculty of Education, University of Ljubljana [3] and at the Department of Informatics, University of Rijeka.

### *D. GameMaker: Studio*

GameMaker: Studio is a development tool for creating a 2D video game. It is also possible to develop 3D video games, but the documentation notifies about the possibility of having several problems with depths, views and other attributes [16]. This engine is used by teachers in several schools to create digital games that suit their curriculum or to improve pupils' programming skills. This engine enables development of games without programming skills. It offers an intuitive and easy Drag-and-Drop (DnD) interface action icon that provides a very quick start development of its own games. Actions are used to define

how, when and where we want to something happened during the game. It is possible to load and create images, sounds and other graphical elements, and to add them to the objects in the scene. While creating actions, GameMaker: Studio in the background creates code of those actions which this tools make appropriate for users with and without programming experience [17]. GameMaker: Studio allows to create games for various platforms, including Windows, MAC OS, Web, Android and iOS [18]. The feature of this engine includes a sprite editor, timelines, paths, room editor, a custom programming language GameMaker Language (GML) and connection with external databases [19].

### *E. Phaser editor*

Phaser Editor is an integrated development environment for creating HTML5 2D games. For a development of games in this tool, it is necessary to know programming. It offers a lot of examples, different Physics libraries and some plugins [20]. Programming is not required only for adding objects to the scene since it is done using a simple drag-and-drop option. However, everything else must be programmed by a user therefore knowledge of JavaScript programming language is required. [21].

## IV. CRITERIA, TABLES AND COMPARISON RESULTS OF SELECTED ENGINES

Selected five tools that are primarily designed for making 2D games were compared in five criteria which are important for the development of the game. For the purposes of this research, we developed our own set of criteria and divided them into five groups:

- basic features of the engine including price
- support, flexibility, interoperability and usability
- engine system requirements and installation
- functionality and the ability to export
- multimedia support and working environment.

Among them, we considered as the most important criteria: functionality, basic features of the engine including price, and flexibility. For each of these groups a table with detailed engine comparison is presented. We also introduce short descriptions of the comparison results for other criteria.

### *A. Basic features of the engine including price*

The first criterion is split into two groups - the basic features and tools price. Basic features of the engine give basic information about every of them like the first version, the current version, is the tool open source or not etc., while price describes whether the engine is free or a commercial tool.

The comparison of selected engines based on the first criteria is given in Table 3.1.

After the comparison according to the criteria of the basic engine features and the price, considering the basic features, we can see that the oldest among them is GameMaker: Studio and the newest is Phaser Editor released in 2015. Adventure Game Studio, e-Adventure and Phaser Editor are open source engines, and Construct 2 and GameMaker: Studio is not. The programming language of the compared engines is relatively similar - Java, JavaScript, C #, C ++, except for GameMaker: Studio, which is based on GML, his own programming language. Considering the criterion of price, Adventure Game Studio and e-Adventure are totally free tools and Construct 2, GameMaker: Studio and Phaser Editor are commercial tools, although they have free demo versions.

### *B. Support, flexibility, interoperability and usability*

Support criterion includes categories that describe customer support, a community of users and documentation. Flexibility criterion is about how much the engine is adapted to some special situations. Interoperability includes the ability to use ASCII and UTF-8 characters in the compared engines. Usability implies how easily and quickly the engine can be learned and how simply is to develop a game.

The comparison of engines based on the second criteria is given in Table 3.2.

After the comparison according to the criteria of support, mostly all engines offer support for customers, complete documentation tools, access to finished examples of games and video tutorials for its community of users, access to forums, blogs and social networks. With regard to the criterion of flexibility, all compared engines are accessible to the wider community of users.

Creating multiplayer games allow only Construct 2, GameMaker: Studio and Phaser Editor. Of all benchmarked engines, only e-Adventure programming tool has multi-language support and, support for SCROM 1.2 (**S**harable **C**ontent **O**bject Reference **M**odel – a collection of standards and specifications for web-based e-learning) reference model and the possibility of integration with the LMS systems. Also, is the only tool among compared that is tailored to work with people with special needs.

According to the criterion of interoperability, all compared engines support the ASCII character set, while UTF-8 character set only Adventure Game Studio does not support. Regarding the criterion of usability, e-Adventure, GameMaker: Studio and Construct 2 enable creating games without programming. GameMaker: Studio has an advanced option that enabled programming in a way that allows users can create more game options. Adventure Game Studio requires programming scripts that define certain actions in the game.

### *C. Engine system requirements and installation*

Engine system requirements include which operating system is required for installation and some properties related to hardware. Examination shows that most of the comparison tool requires minimum Windows XP, Windows Vista or newer operating system. Considering the compared data of selected engines, the least system requirements for installation has Adventure Game Studio. Installation criterion describes how easy engine installation is, and testing has shown that all engines are simple to install and have appropriate installation instructions.

TABLE 3.1. COMPARISON BASED ON THE BASIC FEATURES OF THE ENGINE AND PRICE CRITERIA

| CRITERIA | CATEGORY | Adventure Game Studio | Construct 2 | e-Adventure | GameMaker: Studio | Phaser Editor |
|---|---|---|---|---|---|---|
| BASIC FEATURES OF THE ENGINE | *First version and release year* | Adventure Creator version 1, 1997. | Construct 2 r45, 2011. | e-Adventure 0.1 b, 2009. | Animo, 1991. | Phaser Editor 15.11 RC, 2015. |
| | *Current version and release year* | Adventure Game Studio 3.4.0. 2016. | Construct 2 r239,2016. | e-Advenure 1.5., 2012. | GameMaker: Studio 1.4.1763, 2016. | Phaser Editor 1.2.1., 2016. |
| | *Open source* | Yes | No | Yes | No | Yes |
| | *Programming language* | Scripting language similar to Java and C# | C++, JavaScript | Java | GML programming language | JavaScript |
| PRICE | *Free engine* | Yes | Free version, but without all the features | Yes | Free version, but without all the features | Available a free version for 15 days without all the features |
| | *Commercial version* | No | Personal and Business version of the engine | No | Professional and Master Collection | 1 year, 2 year or Lifetime License |

### *D. Functionality and the ability to export*

Functionality analysis the features that are specific for game development, while ability to export lists possible ways of exporting games for different platforms. The comparison of selected engines based on the functionality criteria are given in Table 3.3 After the comparison according to the criteria of functionality, we found out that all tools support the development of 2D computer games quite good. eAdventure allows making first-person games where the main character is not present, and third-person where the game is playing with the main character (avatar), Adventure Game Studio allows creating games in the third person, while other engines do not have limited modes for games creation. In the Adventure Game Studio and eAdventure navigation through the game is with the mouse pointer, and in other tools keyboard controls can also be used. Construct 2 and GameMaker: Studio have built-in Box2D Simulator Editor by which elements like friction and gravity are define. Adventure Game Studio and eAdventure engines have built-in dialogue editors, while GameMaker: Studio has a separate external editor that saves dialogues which are implemented in the project in .txt files. In the eAdventure dialogue can be displayed normally, as a whisper or as a thought, and dialogue text can be converted to speech. It provides the ability to change the main character, NPC (a non-player character/non-person character/non-playable character is any character that is not controlled by a player) or items appearance in scenes. These changes are flags conditioned. It has a built-in book subject that facilitates the information entry or giving instructions to a player. Built-in item inventory has Adventure Game Studio and eAdventure, while in others are necessary to program the inventory. In Adventure Game Studio and eAdventure are necessary to set up the main character to every scene, regardless of whether it is needed there or not. Out of all tools, only eAdventure offers game statistics view. Assessment and evaluation through a various question can be made in Adventure Game Studio, eAdventure and GameMaker: Studio, while in other tools this feature is not optimized in the best way. From all compared tool, the best option to create multiple-choice questions and assessment provides eAdventure. Regard to the criterion ability to export the game the most options for exporting offers Construct 2, and at least Phaser Editor engine.

Regarding the ability to export the game the most options for exporting offers Construct 2, and at least Phaser Editor engine, but we should consider that it's created in 2015. and is still in development phase. All engines can export games for Windows and Linux platforms. eAdventure and Phaser Editor can't export for Android, iOS nor consoles' platforms and Phaser can't export for Mac OS X. Only eAdventure offers the ability to export g games as learning object where the project is saved as educational content.

### *E. Multimedia support and working environment*

Criterion multimedia support is related to which multimedia elements engine supports, while engines working environment analyzes if an engine is user-friendly. After the comparison according to multimedia criterion, it seems that all compared engines allow the inclusion of multimedia elements like images, sound, video and animation. Compared engines mainly support common formats for image, sound, video and animation, while GameMaker: Studio supports vector graphics. The e-Adventure can create animations of the main characters and NPC only for standing, talking, walking and the use of objects, while other tools can create animations on request. Considering the category of engine working environment all compared engines have an intuitive graphical interface with Drag-and-Drop options that can be adjusted if necessary. Also, all of them allow testing of the project as well as the detection of errors that occur when creating games.

### *F. Review of work with selected software tools for comparison*

The research involved five engines for creating computer games which are primarily designed for making 2D games. For this study, we have developed our criteria and according to them compared all selected engines. In every engine, we made a basic version of the game, while in GameMaker: Studio a full version of the game was developed. For each compared tool we've created a new project in order to examine the functionality and features of the engine. We adapted designed game scenario to the possibilities of particular engine while some elements could not be realized in each of them. Further, in GameMaker: Studio, Construct 2 and Phaser Editor free engine versions was not possible to use

TABLE 3.2 COMPARISON BASED ON THE FLEXIBILITY AND SUPPORT CRITERIA

| CRITERIA | CATEGORY | Adventure Game Studio | Construct 2 | e-Adventure | GameMaker: Studio | Phaser Editor |
|---|---|---|---|---|---|---|
| SUPPORT | *Customer support* | Yes | Yes | Yes | Yes | Yes |
| | *Complete documentation* | Yes | Yes | Yes | Yes | Yes |
| | *Community* | Yes | Yes | Yes | Yes | Yes |
| | *Forum/Blog* | Yes | Yes | Yes/No | Yes | No |
| | *Facebook/Twitter* | Yes/No | Yes | No | Yes | Yes |
| FLEXIBILITY | *Adapted to work with people with special needs* | No | No | Yes | No | No |
| | *Accessible to the wider community* | Yes | Yes | Yes | Yes | Yes |
| | *Integration with LMS systems* | No | No | Yes | No | No |
| | *SCORM 1.2* | No | No | Yes | No | No |
| | *Multilingual engine* | No; English only | No; English only | Yes; | No; English only | No; English only |
| | *Multiplayer* | No | Yes | No | Yes | Yes |

all the features and functionalities. After the comparison, GameMaker: Studio was chosen for the development of the full game "Explore the solar system". since it met all the required criteria. We decided to use it because it's free engine for developing 2D adventures games for PCs. It has an intuitive graphical interface and complete engine, and it's relatively easy to create game with this engine. It has very good user community, offers various examples, tutorials, manuals, forums, developers personal experience.

The serious game "Explore the solar system" is intended for 5th grade students and is associated with teaching unit Earth in space in Geography course. The game can be used at the beginning of the lesson as an introduction in order to provide students' motivation for learning. Also, it can provide the repetition about the Solar system in a funnier and more interesting way for students. The game type that we developed is a 2D point and click adventure.

## V.CONCLUSION

One aspect of using ICT in the education is game-based learning and using serious computer games. Implementing games encourage students to participate and develop their creativity. They are primarily used for successful material completion in a more interesting and fun way. Serious games represent an opportunity for enhancing education from primary school through to lifelong learning.

The creation of a serious game is a complex project, and one of the most challenging task is to choose the appropriate game engine. Within this study, we examined five tools for game development. The comparison showed that each tool has certain advantages and disadvantages. Customers choose development platform based on engine characteristics, the experience they have in game development and their needs. This paper proposed criteria that will allow the developer to select the engine based on these characteristics.

For game development, we chose GameMaker: Studio since it met all the criteria set. It's very affordable engine with intuitive graphical interface and simple mode which facilitate game development.

GameMaker: Studio is suitable for beginners in game development, or for those who do not have much programming experience because in a very short period they can make a simple game.

TABLE 3.3 COMPARISON BASED ON THE FUNCTIONALITY CRITERIA

| CATEGORY | Adventure Game Studio | Construct 2 | eAdventure | GameMaker: Studio | Phaser Editor |
|---|---|---|---|---|---|
| *Game types* | *Point & Click* adventures | Different HTML5 2D games | *Point & Click* adventures | Different 2D games | Different HTML5 2D games |
| *Creating a game in a certain mode* | Yes, in the third person | Not limited | Yes, in the first or third person | Not limited | Not limited |
| *Including finished scripts in tools* | Yes | Yes | No | Yes | Yes |
| *Navigation through the game* | Mouse pointer | Mouse pointer or keyboard | Mouse pointer | Mouse pointer or keyboard | Mouse pointer or keyboard |
| *Programming code editor* | Yes | Yes | No | Yes | Yes |
| *Box 2D Simulator Editor* | No | Yes | No | Yes | No |
| *Game logic* | Scripts defined | Defined with certain events and actions | Defined with flags and variables | Programming | Programming |
| *Behavior of the characters and games objects* | Yes | Yes | Yes | Yes | Yes |
| *Dialogue editor* | Yes | No | Yes | External editor | No |
| *Embedded options for defining the movement of the main character and the NPC* | Yes | Yes | Yes | Yes | No |
| *Turning another elements look (the main character, NPCs, items)* | Yes | No | Yes | No | No |
| *Input the instructions* | No | No | Yes | No | No |
| *Items inventory* | Yes | No | Yes | No | No |
| *Creating icon object for games inventory* | No | No | Yes | No | No |
| *Embedded options to save, load and exit the game* | Yes | No | Yes | No | No |
| *Timer option* | Yes | No | Yes | Yes | No |
| *The random appearance of objects* | No | Yes | No | Yes | Yes |
| *The game statistics view option* | No | No | Yes | No | No |
| *Creating multiple-choice questions* | Yes | No | Yes | Yes | No |
| *Examination and assessment through various questions* | Possible to make | Possible, but not the best optimization | Well optimized | Possible to make | Possible, but not the best optimization |


## ACKNOWLEDGMENT

The research has been conducted under the project "E-learning Recommender System" (reference number 13.13.1.3.05) supported by University of Rijeka (Croatia).


## REFERENCES


[1] G. J. Hwang, L. H. Yang, and S. Y. Wang, "A concept map-embedded educational computer game for improving students' learning performance in natural science courses," *Comput. Educ.*, vol. 69, no. August, pp. 121–130, 2013.

[2] M. Prensky, "Digital Natives, Digital Immigrants," *Horiz.*, vol. 9, no. 5, pp. 1–6, 2001.

[3] J. Rugelj, "Serious computer games in computer science education," *EAI Endorsed Trans. Game-Based Learn.*, vol. 2, no. 6, p. 150613, 2015.

[4] S. Erhel and E. Jamet, "Digital game-based learning: Impact of instructions and feedback on motivation and learning effectiveness," *Comput. Educ.*, vol. 67, no. March 2016, pp. 156–167, 2013.

[5] W. Horton, "Games and simulations," in *E-Learning by Design*, vol. 45, no. 5, 2012, pp. 323–398.

[6] N. Whitton, *Learning with digital games*. 2009.

[7] C. Peng, "Introductory game development course: A mix of programming and art," *Proc. - 2015 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2015*, pp. 271–276, 2016.

[8] E. J. Marchiori, J. Torrente, Á. Del Blanco, P. Moreno-Ger, P. Sancho, and B. Fernández-Manjón, "A narrative metaphor to facilitate educational game authoring," *Comput. Educ.*, vol. 58, no. 1, pp. 590–599, 2012.

[9] C. D'Apice, C. Grieco, R. Piscopo, and L. Liscio, "Design of an Educational Adventure Game to teach computer Security in the working environment," *21st Int. Conf. Distrib. Multimed. Syst. DMS 2015 A4 - Knowl. Syst. Inst. Grad. Sch. KSI Res. Inc.*, pp. 179–185, 2015.

[10] S. Machines, »Adventure Game Studio - Members List,« Simple Machines, 2013. [Online]. April 21, 2017. Available: http://www.adventuregamestudio.co.uk/forums/index.php?action=mlist.

[11] SCIRRA, "Construct 2." 2016.

[12] J. Torrente, P. Moreno-Ger, B. Fernández-Manjón, and Á. del Blanco, "Game-Like Simulations for Online Adaptive Learning: A Case Study," *Int. Conf. Technol. E-Learning Digit. Entertain.*, pp. 162–173, 2009.

[13] J. Torrente, Á. Del Blanco, and ... E. M., "<E-Aventure> Introducing Educational Games in the Learning Process," *Educ. ..., 2010 -. ieeexplore ieee org.*, 2009.

[14] J. Torrente, Á. del Blanco, P. Moreno-Ger, I. Martínez-Ortiz, and B. Fernández-Manjón, "Implementing accessibility in educational videogames with <e-Adventure>," *Proc. first ACM Int. Work. Multimed. Technol. distance Learn.*, pp. 57–66, 2009.

[15] Á. Blanc, J. Torrente, E. J. Marchiori, I. Martínez-ortiz, P. Moreno-ger, and B. Fernández-Manjón, "Easing Assessment of Game-based Learning with <e-Adventure> and LAMS," *Contract*, pp. 25–30, 2010.

[16] D. Bigas Ortega, "Machine Learning Applied to Pac-Man Final Report," 2015.

[17] YoYo Games Ltd., "GameMaker_ Studio _ Documentation." 2015.

[18] B. Froz, A. Busson, R. Martins, and A. Oliveira, "Comparison between two Development Platforms of Games to E- learning Courses," *Int. Conf. Futur. Educ. 3rd*, 2013.

[19] E. Johansson and T. Andersson, "A closer look and comparison of cross-platform development environment for smartphones," 2014.

[20] A. Fornaris, "Phaser Editor _ HTML5 2D Games IDE." .

[21] A. M. de Miguel, "HTML5 2D videogame programming," 2016.