



**Francisco Duarte Simões Gonçalves**

Licenciado em Ciências da Engenharia Electrotécnica e de Computadores

# Game Wizard

Dissertação para Obtenção do Grau de Mestre em  
Engenharia Electrotécnica e de Computadores

Orientador: Dr. Tiago Oliveira Machado de Figueiredo Cardoso,  
Professor Auxiliar,  
Faculdade de Ciências e Tecnologia,  
Universidade Nova de Lisboa



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA

Março, 2017



Game Wizard

Copyright © Francisco Duarte Simões Gonçalves, Faculdade de Ciências e Tecnologia,  
Universidade Nova de Lisboa.

The Faculty of Science and Technology and the New University of Lisbon are entitled, perpetual and without geographical boundaries, to archive and publish this dissertation through printed copies reproduced on paper or digital form, or by any other means known or hereafter be invented, and divulge it through scientific repositories and admit its copy and distribution for educational purposes or research, not commercial, as long as credit is given to the author and publisher.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



**“Do not get obsolete like an old technology, keep innovating yourself” –  
Sukant Ratnakar**



# Agradecimentos

Gostaria de começar por agradecer à minha família por todo o apoio dado ao longo destes anos, sem os quais este percurso não teria sido possível.

Agradecer ao meu orientador, Professor Tiago Cardoso, não só pela orientação e dedicação prestada ao longo desta dissertação, como também pelas oportunidades e ideias que me disponibilizou.

Devo também um agradecimento pelo apoio e divulgação fornecido pela Organização Pais-em-Rede, sem a qual parte da componente prática da presente dissertação não teria tido a mesma exposição.

A todos os professores do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa que de uma maneira ou outra sempre se mostraram disponíveis para qualquer esclarecimento ou divulgação de conhecimento.

Agradeço aos meus colegas e amigos que tanto me ajudaram e incentivaram ao longo destes anos, sem os quais o alcançar desta etapa não teria sido possível. Por fim, um especial obrigado à minha namorada Sara por todo o apoio e paciência que sempre demonstrou.





# Abstract

Currently there is a growing demand for educational games and a consequent investment in this area. These games are mainly used to spread knowledge in an appealing and motivating way, in an attempt to reach different types of public.

This demand, coupled with the need to adapt and customize the final product regarding the users' ideas is limited, since the creation of new solutions that meet these needs is a complex process due to the specific knowledge required for its achievement.

The present study emerged from these facts.

As a main objective of this research work and in order to overcome the previously presented problems, a solution will be proposed to simplify the process of creating these games. For this purpose, content loading and manipulation mechanisms were used, making it a solution available to all, regardless of the level of specific technical knowledge, usually necessary for its use. As a possible solution, a game creation platform and an Android application will be developed.

The game creation platform will consist of a web application for uploading content (images, sounds, questions and answers) to the server and database, the content will later be imported and used in the Android application.

The Android application will consist of a mechanism that includes a set of modules that will be filled with the imported content, thus creating custom games and making them available to the player.

In the end, this solution will allow the creation of a simple and intuitive product, easy to use by anyone and that can generate games by filling in the necessary components such as images, sounds and questions.

**Keywords:** Game Wizard, Game Engine, Educational Games



# Resumo

Actualmente existe uma crescente procura de jogos educativos e um conseqüente investimento nesta área. Estes jogos são sobretudo utilizados para divulgar conhecimento de um modo apelativo e motivante, numa tentativa de chegar a diferentes tipos de público.

Esta procura, em junção com a necessidade de adaptação e personalização do produto final face ao pretendido pelo utilizador vê-se limitada, uma vez que a criação de novas soluções que satisfaçam estas necessidades é um processo complexo devido aos conhecimentos muito específicos necessários para a sua realização.

O presente estudo surgiu da constatação destes factos.

Como objectivo principal deste trabalho e de modo a colmatar a problemática apresentada anteriormente, irá ser proposta uma solução com o intuito de simplificar o processo de criação destes jogos. Como solução possível, irá ser desenvolvida uma plataforma de criação de jogos e uma aplicação Android na qual os mesmos poderão ser jogados.

A plataforma de criação de jogos irá consistir numa aplicação web para carregamento do conteúdo (imagens, sons, perguntas e respostas) para o servidor e base de dados, conteúdo esse que será posteriormente importado e utilizado na aplicação Android.

A aplicação Android irá consistir num mecanismo que engloba um conjunto de módulos que serão preenchidos com o conteúdo importado, criando assim jogos personalizados e tornando-os acessíveis ao jogador.

No final, esta solução irá permitir a criação de um produto simples e intuitivo, de fácil manejo por qualquer pessoa e que poderá gerar jogos através do preenchimento das componentes necessárias, como imagens, sons e perguntas.

**Palavras-chave:** *Game Wizard*, Motor de Jogo, Jogos Educativos



# Contents Index

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	MOTIVATION	1
1.2	MAIN OBJECTIVE	2
1.3	SECONDARY OBJECTIVE	3
1.4	DISSERTATION STRUCTURE	3
<b>2</b>	<b>STATE OF THE ART</b>	<b>5</b>
2.1	VIDEO GAMES	5
2.1.1	<i>Serious Games</i>	6
2.2	GAME ENGINES	14
<b>3</b>	<b>PROPOSAL</b>	<b>17</b>
3.1	DEVELOPER DIMENSION	18
3.1.1	<i>Developer Architecture</i>	18
3.2	PLAYER DIMENSION	20
3.3	GLOBAL DIMENSION	20
<b>4</b>	<b>IMPLEMENTATION OF THE PROTOTYPE</b>	<b>23</b>
4.1	DEVELOPER IMPLEMENTATION	23
4.1.1	<i>GUI</i>	23
4.1.2	<i>Back-end</i>	32
4.2	PLAYER IMPLEMENTATION	35
4.2.1	<i>GUI</i>	35
4.2.2	<i>Back-end</i>	44
4.3	INTERCONNECTION BETWEEN DEVELOPER AND PLAYER	49
4.4	DATABASE AND SERVER	49
4.4.1	<i>Structure</i>	50
4.5	GAME ENGINE	54
<b>5</b>	<b>VALIDATION</b>	<b>57</b>
<b>6</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>59</b>
6.1	CONCLUSION	59
6.2	FUTURE WORK	60

<b>7 REFERENCES .....</b>	<b>61</b>
<b>APPENDIX A - DEVELOPER MANUAL .....</b>	<b>63</b>

# Figures Index

FIGURE 2.1 - AMERICA'S ARMY GAMEPLAY ("AMERICA'S ARMY" 2016) .....	7
FIGURE 2.2 - AGE OF MYTHOLOGY GAMEPLAY ("AGE OF MYTHOLOGY HEAVEN: A GUIDE TO POSEIDON" 2016) .....	8
FIGURE 2.3 - CSI WEB ADVENTURES GAMEPLAY ("CSI: NEW YORK WALKTHROUGH, GUIDE" 2016) .....	9
FIGURE 2.4 - SIMS 2 GAMEPLAY ("THE SIMS 2 SCREENSHOTS FOR WINDOWS - MOBYGAMES" 2016) .....	10
FIGURE 2.5 - SIMCITY GAMEPLAY ("SIMCITY 4" 2016) .....	11
FIGURE 2.6 - PULSE GAMEPLAY (SQUIRE 2008A) .....	12
FIGURE 3.1 - DIAGRAM OF A NEW GAME CREATION .....	19
FIGURE 3.2 - SEQUENCE DIAGRAM .....	21
FIGURE 4.1 - LOGIN SCREEN .....	24
FIGURE 4.2 - REGISTER SCREEN.....	25
FIGURE 4.3 - NEW GAME SCREEN.....	26
FIGURE 4.4 - INSERT GENERAL GAME CONTENT SCREEN .....	26
FIGURE 4.5 - SELECT GAME MODE SCREEN .....	27
FIGURE 4.6 - INSERT NAME - GAME MODE SCREEN.....	28
FIGURE 4.7 - EQUAL PARES SMALL - GAME MODE SCREEN.....	29
FIGURE 4.8 - EQUAL PARES - GAME MODE SCREEN.....	29
FIGURE 4.9 - LINES - GAME MODE SCREEN .....	30
FIGURE 4.10 - MULTIPLE CHOICE - GAME MODE SCREEN.....	31
FIGURE 4.11 - PUZZLE SMALL - GAME MODE SCREEN .....	31
FIGURE 4.12 - PUZZLE - GAME MODE SCREEN .....	32
FIGURE 4.13 - FINAL SCREEN.....	32
FIGURE 4.14 - WEBSITE SOURCE CODE - INSERT NAME SCRIPT .....	34
FIGURE 4.15 - APPLICATION INTRO SCREEN .....	35
FIGURE 4.16 - INSERT PLAYER NAME SCREEN .....	36
FIGURE 4.17 - SELECT GAME SCREEN.....	37
FIGURE 4.18 - UPDATE SCREEN.....	37
FIGURE 4.19 - PLAY/SETTINGS SCREEN .....	38
FIGURE 4.20 - SETTINGS SCREEN .....	39
FIGURE 4.21 - INSERT NAME - GAME MODE .....	39
FIGURE 4.22 - LOGOS TRANSITION SCREEN .....	40
FIGURE 4.23 - EQUAL PARES SMALL - GAME MODE .....	40
FIGURE 4.24 - EQUAL PARES - GAME MODE .....	41
FIGURE 4.25 - LINES - GAME MODE.....	42
FIGURE 4.26 - MULTIPLE CHOICE - GAME MODE.....	42

FIGURE 4.27 - PUZZLE SMALL - GAME MODE .....	43
FIGURE 4.28 - PUZZLE - GAME MODE .....	43
FIGURE 4.29 - END SCREEN.....	44
FIGURE 4.30 - END GAME SOURCE CODE.....	44
FIGURE 4.31 - SOUND MANAGEMENT SOURCE CODE .....	45
FIGURE 4.32 - NEXT SCENE MANAGEMENT SOURCE CODE .....	46
FIGURE 4.33 - LOCAL DATA STORAGE.....	46
FIGURE 4.34 - ASSETBUNDLE MANAGEMENT SOURCE CODE .....	47
FIGURE 4.35 - ASSETBUNDLE CONTENT SOURCE CODE .....	47
FIGURE 4.36 - ASSETBUNDLE SOUND SOURCE CODE .....	48
FIGURE 4.37 - UPDATE APPLICATION SOURCE CODE .....	48
FIGURE 5.1 - APPDATA DATABASE TABLE .....	57



# Table Index

TABLE 2.1 - GAME COMPARISON .....	13
TABLE 2.2 - GAME ENGINE COMPARISON .....	15
TABLE 4.1 - APPDATA CONTENT TABLE.....	50
TABLE 4.2 - DEVELOPERINFO CONTENT TABLE .....	51
TABLE 4.3 - USERDATA CONTENT TABLE .....	52
TABLE 4.4 - INSERTNAMEGAME CONTENT TABLE .....	53
TABLE 4.5 - LINESGAME CONTENT TABLE .....	53
TABLE 4.6 - MULTICHOICEGAME CONTENT TABLE.....	54



# Acronyms

**2D** – Two dimensions

**3D** – Three dimensions

**APK** – Android Package

**FPS** – First Person Shooter

**GUI** – Graphical User Interface

**iOS** - iPhone Operating System

**OS** – Operating System

**PC** – Personal Computer

**STB** – Social Tech Booster



# 1 Introduction

This research work consists of the creation of a Game Wizard concept.

The main focus of this concept is to allow for an easy creation of games, based on sets of already existing components, like images, sounds, questions and answers that can be applied to different scenarios but still maintaining its simplicity of use for the developers, so that they won't need any specific programming or software knowledge.

In order to create a proper solution, partnerships with Social Tech Booster ("Social Tech Booster" 2016) and Pais-em-Rede ("Pais Em Rede," n.d.) were created, both organizations work with social causes and helped in the development and validation of this concept in a real environment.

In this first chapter a brief presentation of the motivation for this research work will be presented, as well as the main and secondary objectives regarding its development and lastly a short description of this document's organisation in terms of its contents.

## 1.1 Motivation

The social aspect of the dissertation's subject is by itself the biggest motivation, since it aims to contribute for the dissemination of the written stories about children with disabilities, contained in the books published by Pais-em-Rede, by making it possible for people without any programming knowledge to create new games with the books' content.

The considered market for the application is mainly for children, with the purpose of complementing the published books and thus lead the children to read them in order to be able to play the created games and helping them to realize that despite the fact that the children in the stories have disabilities, it doesn't mean they don't like to play and interact as much as any other child.

The final aspect is the vast diversity of features, programming languages, platforms, among others that also serve as a motivation for this research work, by allowing for further knowledge development in all the areas stated above.

## **1.2 Main Objective**

The main objective of this research work is the development of a semi-automatic game module generator and an application to be able to manage and run the game modules. These games will be selected by the developer from a predefined set of available game modes, by choosing their order, which modes they wish to have in the game module and by filling the necessary content for each of the selected game modes.

The developed solution has two associated dimensions:

The Developer logic consists on a website whose sole purpose is to allow for a simple way to allow any person (that is registered as a developer) to choose not only the game modes, but also to submit the necessary contents (images, sounds, questions, etc.) and fill the required fields to enable the creation of the game modules, these modules will later be used to fill the configurable components of the various requested game modes, as well as the sequence in which they will be played.

The Player logic consists on an Android OS application that will manage and run the several game modules based on the player's preferences, by allowing the selection of a game from a predefined list of possible games to play and downloading the game modules that the player chooses to play. Lastly it will collect some data and send it to the database with the players' information, like names, smartphone model, time played among other information to allow for an analysis of some of the game's specifications, to understand if changes need to be made regarding difficulty, incompatibilities, etc.

## 1.3 Secondary Objective

The secondary objective of this research work is to focus the development of the solution, so that that the game modules will be based on the books that will be published, that tell short stories of children with disabilities, using its characters, stories and other relevant content. It's intended primarily for children above six years of age, having an increasing difficulty as the game progresses, as stipulated by the Developer.

Since the application has to work with different books, containing different content, it must be implemented to be as generic as possible, allowing for a pre-set number of components to be changed, that will result in different games, but with the same foundations.

The purpose of focusing the development so that the games will be based on books, is to require for an interpretation of the books' content, making it an interactive and fun way of making children learn about the subject.

It is also relevant to denote that even though the development was made with a specific focus, it doesn't limit the solution to only this problematic, meaning that it can be used for other purposes that can fit in the specified frame of the solution.

## 1.4 Dissertation Structure

The document's structure is organized as follows:

- **Chapter 2 - State of the Art**

In this chapter a brief research about the concepts of Video Games, Serious Games and Game Engines is presented as well as some significant examples and comparisons of some of the referred concepts.

- **Chapter 3 – Proposal**

In this chapter the proposal for the solution to be developed will be presented, for both the Player and Developer parts. Some suggested architectures are also presented along with some creation sequences, these will be used to better develop the solution.

- **Chapter 4 - Implementation**

This chapter consists of a description of the solution's implementation, it will be divided in three main chapters and some subchapters, as presented below:

- Developer – GUI and Back-end;
- Player – GUI and Back-end;
- Interconnection between Developer and Player;
- Database and Server;
- Game Engine.

- **Chapter 5 - Validation**

In this chapter the dissertation's validation criteria is explained in a detailed manner, as well as a simple demonstration of some of the entries and how the data is stored in the database.

- **Chapter 6 – Conclusion and Future Work**

In this chapter, as the name leaves to understand, will be presented some of the conclusions taken from the solutions development as well as some suggestions for future work, like possible component alterations and adding new features to the final solution.



## 2 State of the art

This chapter presents some concepts considered relevant to the development of the research work's topic, as well as a comparative analysis of some existing development tools that are presently on the market, that are necessary for the development of the final solution.

It's important to note that the tools compared in this chapter represent a very small percentage of the ones existing currently on the market, and that these were chosen due to their specific characteristics.

### 2.1 Video Games

**“Video game: a mental contest, played with a computer according to certain rules for amusement, recreation, or winning a stake.”**

(Zyda 2005)

The concept of a game varies according to if a person grew up playing video games or not, being that this generation-gap is explained by the fact that people that have been exposed to video games their entire lives associate the concept of a game differently (Zyda 2005).

A video game has as content art, sounds, levels and maps (Zook and Riedl 2014) as well as a storyline, and its design is substantially complex, sometimes involving several different development teams that define its components in order to achieve a final product. An example of the different teams necessary for game development is to divide them into two parts: the art team, responsible for how the game looks and the programming team, responsible for the code implementation, interfaces, game engine changes, etc. (Zyda 2005).

### 2.1.1 Serious Games

**“Serious games are IT applications that combine aspects of tutoring, teaching, training, communications and information, with an entertainment element derived from videogames. By offering this combination, the programs aim to make practical, utilitarian content (serious) enjoyable (game).”** (Alvarez and Michaud 2008)

With the evolution of games and the impact that they have, the concept of serious games has become increasingly important, becoming the study focus of several areas such as psychology, computer science, sociology, among others. Being that the growing number of publications, conferences and companies dedicated to the subject is a great indicator of its increasing relevance (Breuer and Bente 2010).

Currently there are several cases of serious games used in training environments such as America's Army, developed to assist the American army in its training (Zyda 2005), and educational environments like Civilization in the field of history, CSI in the field of forensic science, The Sims 2 regarding social skills and SimCity in the field of engineering (Bellotti, Berta, and De Gloria 2010). There is a growing need to customize the gameplay to the player's interests, adapting certain characteristics of the game, such as the difficulty levels in order to improve their gaming experience and captivating the player (Bueno 2014).

There are however some differences regarding the way to categorize serious games according to their market, since they can be classified for example according to following categories: Military Games, Government Games, Educational Games, Corporate Games, Healthcare Games, Political Games Religious Games, Art Games (Susi, Johannesson, and Backlund 2007). They may also be classified in a different perspective encompassing however some similarities and dividing them into: Defence, Teaching and training, Advertising, Information and communications, Health, Culture, Activism (Alvarez and Michaud 2008), and there are still different ways to categorize them according to other authors (Breuer and Bente 2010).

## America's Army



Figure 2.1 - America's Army Gameplay ("America's Army" 2016)

America's Army can be considered as one of the most relevant games when mentioning serious games, the game was launched in 2002 and is still played today.

The game was developed with the purpose of simulating military trainings and missions using an online virtual environment and resorting to the concept of a first person shooter (FPS) developed as a training and recruiting tool for the American army. The game was so successful in training people that the players with the highest scores received a letter inviting them to enlist in the army.

Updates to the game are still coming out and it can be played in multiple platforms, such as PC, Xbox, PlayStation, among others (Zyda 2005).

## Age of Mythology



Figure 2.2 - Age of Mythology Gameplay (“Age of Mythology Heaven: A Guide to Poseidon” 2016)

The game Age of Mythology is based on the mythology present in different civilizations in world history, such as Greek, Roman and Egyptian.

This game can be divided in multiple components, since on one hand, there is the creation, construction and maintenance component for different cities, for which it’s necessary to collect resources (such as food, wood, divine protection, etc.) and on the other hand there is the need to fight, defend and expand the population by conquering the enemies land and destroying their civilizations. The player will also advance across the ages, evolving his armies, unlocking new types of buildings, weapons, ships, etc., and being challenged to fulfil certain objectives in order to overcome each stage.

This kind of games are very important culturally, in the sense that the players start to get an interest about historical facts and many times, voluntarily search for information regarding the subject of ancient civilizations (Breuer and Bente 2010).

## CSI: The Experience



Figure 2.3 – CSI Web Adventures Gameplay (“CSI: New York Walkthrough, Guide” 2016)

The CSI Web Adventures game is a set of interactive virtual games that contain cases of forensic analysis and small attention games in order to promote learning on the subject of forensic science.

In this game, the players’ objective is to collect evidence from the crime scene, interrogate suspects and analyse the clues to solve the crime with which he faces, or solve small games like looking for differences between similar pictures and game modes alike, with the objective of helping the player to improve his concentration skills. The various scenarios presented throughout the levels integrate some simple concepts of laboratory techniques such as DNA analysis, toxicology, etc., introducing these concepts in a simple and fun way.

This game was developed in partnership with relevant organisations such as the National Institute on Drug Abuse, the American Academy of Forensic Sciences and the National Science Foundation, so the scenarios and facts presented in the game are properly supported and are introduced by professionals with training in the area (Miller, Chang, and Hoyt 2010).

## Sims 2



Figure 2.4 - Sims 2 Gameplay (“The Sims 2 Screenshots for Windows - MobyGames” 2016)

The Sims 2 consists of a virtual environment, initially developed with the intention to allow the projection of housings, which was expanded to include the control of a family’s characters, where the player can define the actions that they will take.

It’s important to note that the interactions with the characters are usually taken based on a necessities index, where the player can check certain aspects associated with each character, such as their hunger, hygiene, energy, social component, among others. The player’s objective is to manage these aspects in order to comply with the requirements set out, by keeping their characters 'happy'. It’s also possible to control other components, such as where they work or if they go to school, manage their incomes, that can be used to purchase goods such as food, electricity and other non-essential goods such as expanding their housing, acquiring furniture, etc.

This type of game has a great social significance, since it presents the consequences of certain actions and choices made by the player, as well as introducing some responsibility concepts associated with daily activities (“The Sims 2 - EA Games” 2016).



## SimCity



Figure 2.5 - SimCity Gameplay (“SimCity 4” 2016)

SimCity is a simulation game where the player is responsible for the city planning as well as its management and other significant aspects.

In this game the player can define the layout of the various buildings, roads, etc., and also manage the monetary resources acquired through activities such as the exploitation of natural resources, tax collection, among others. It’s also possible to simulate the effects/consequences of certain actions and environmental disasters.

This game is very relevant because it allows the player to acquire a new understanding regarding the complexity of managing the various components associated with a city as well as the consequences of his decisions in a safe virtual environment (Bueno 2014).

## Pulse

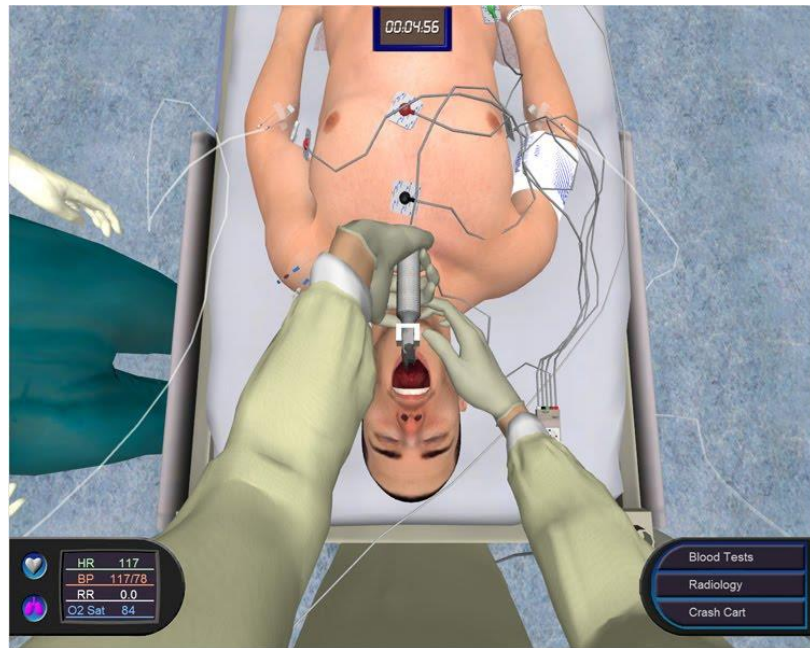


Figure 2.6 - Pulse Gameplay (Squire 2008a)

The Pulse is an interactive virtual game environment where health care professionals (civil and military) can expand their knowledge and technique, that includes some clinical skills considered relevant and that are possible to replicate in a virtual environment.

In this game the player works in a virtual hospital where he can perform diagnoses and treatment to his patients. Since the focus of the game was to develop a way of training the response to disasters such as combat and bioterrorism, the game modes are mainly focused in these actions.

Although it's a game, it's also a highly relevant learning and training tool for health care professionals, since it allows the simulation of specific clinical actions without risks.

This game was developed in partnership with Texas A & M University - Corpus Christi and funded by the Federal Grant meaning that the clinical practice simulations are duly substantiated (Squire 2008b).



The following table presents a brief comparison analysis between the games presented above, focusing in their categories and main educational benefits:

Table 2.1 - Game Comparison

	Category	Educational Aspect
America's Army	Military	Simulation of military training for the American army; Used as a recruitment tool;
Age of Mythology	Strategy	Strategy training by controlling civilizations to expand the empire; Increase the player's interest in history;
CSI: The Experience	Forensic Science	Learn concepts of forensic science such as DNA analysis, toxicology, etc.; Increase concentration skills;
Sims 2	Social	Learn concepts of responsibility and social interactions;
SimCity	City Planning	Acquire knowledge about the complexity of managing several components associated with city planning and management;
Pulse	Health	Simulation of patient diagnosis and treatment for healthcare professionals and the military;

## 2.2 Game Engines

**“the game’s engine refers to that collection of modules of simulation code that do not directly specify the game’s behaviour (game logic) or game’s environment (level data). The engine includes modules handling input, output (3D rendering, 2D drawing, sound), and generic physics/dynamics for game worlds.”** (Lewis, Michael; Jacobson 2002)

As games evolved along the years, so has the need to adapt the way that games are created, by improving the development techniques and making it more and more generic in order to make its adaptation to other games easier, improving future development. Thus it has become increasingly common to use reusable component modules, directly linked with the arising of the concept of Game Engines (Anderson et al. 2008).

Briefly a Game Engine consists of software that uses its own library and inputs provided by the developer, such as scripts and objects, to assist in the development of various games, that can be of different genres and dimensions (Bueno 2014).

It’s also important to consider the growing need to adapt the applications to different operating systems and devices (Android OS, iOS, PC, etc.), the need to use cross-platform game engines to develop new applications becomes a requirement, facilitating their adaptation to the various systems and equipment.

In Table 2.2, some relevant aspects for the selection of the Game Engine are presented, in order to choose the most appropriate for the present application.

Table 2.2 - Game Engine Comparison

	Marmalade	V-Play	Cocos2s-JS	Unity
Programming Language	C++, Lua, Objective-C, HTML5	C++	JavaScript	C#, Javascript
Main Target Platforms	Android OS BlackBerry iOS Mac OS X Microsoft Windows	Android OS iOS Mac OS X Microsoft Windows	Android OS iOS Mac OS X Microsoft Windows Windows Phone 8	Android OS BlackBerry 10 Gear VR iOS Mac OS X Microsoft Windows Oculus Rift PlayStation 4 Xbox Wii U Windows Phone
2D / 3D	2D & 3D	2D	2D	2D & 3D
Support	★★★★☆	★★★★★	★★★★☆	★★★★★
License	Free (w/ limitations)	Free (w/ limitations)	Free (Open-Source)	Free (w/ limitations)
Classification	★★★★☆	★★★★☆	★★★★☆	★★★★☆



## 3 Proposal

In this chapter, the solution's architecture will be presented, that will serve as a foundation for the development of both Player and Developer dimensions. The architecture was designed in order to be as simple as possible for both player and developer to use, allowing for a simple implementation of new game modules and simple comprehension of the gameplay.

This pre-planning is to be considered extremely relevant, since it will allow for a better understanding of the entire solution, also making it easier to find out flaws and correcting them before the actual implementation.

It's relevant to point out that an Android application was also created before the actual Wizard, to allow for a better comprehension of the necessary components, game modes, characteristics and structure to be used. This application has already been published together with one of Pais-em-Rede new books, entitled *Afonso e a Espada Mágica* and will serve as a basis for the development of the actual Game Wizard.

This study will also be used to determine the necessary changes to be made in order to make the application work offline, by structuring a way to save all necessary data locally on the device.

Lastly the explanation of the solution's structure will be separated in three parts in this chapter, Developer Dimension, Player Dimension and Global Dimension, since they coexist but are created in completely different ways.

## 3.1 Developer Dimension

The Developer will consist of any person (registered as such) that will create new game modules that will later be used in the Game Wizard, having in account that the game modules are made of sets of necessary images, sounds, questions and answers to create a new game.

The creation of new game modules will be performed via website, in which the user will register as a developer, login and then proceed to create a new game by filling up the necessary data for its creation and by selecting the game modes from a pre-existing list of available game modes, by whatever order (s)he chooses to have along the gameplay. This selection will result in different webpages being opened so as to fill the necessary information for each game mode.

The uploaded contents inserted by the developer will be stored in the server and database (the images and sounds will go to the server and the questions, answers, sequence and game information will be stores in the database).

Since the Game Wizard will be mainly focused on the development of game modules for published books, the fields will be filled with the books' content (images, questions, character names, etc.) nonetheless it can receive different inputs generating alternative types of games.

### 3.1.1 Developer Architecture

In Figure 3.1 a simple sequence is represented of the new game module generation, to facilitate the comprehension of the necessary steps to take.

In more detail, to create a new game module one needs to take the following steps:

- 1) Register as a Developer;
- 2) Login as a Developer;
- 3) Insert the name of the new game, this is what will be used to reference the inserted game in the game list presented in the Android application;
- 4) Insert the general game content, this content refers to the background music, right and wrong answer sounds and images;
- 5) Select the game modes, these game modes will be presented to the player by the same order as they are selected here. The developer as to choose from the predefined set of created game modes;

- 6) Fill the content for each of the selected game modes, these contents vary from mode to mode, meaning that their content will be significantly different;
- 7) After selecting all the desired game modes and filling them with the proper information, the developer as to confirm that the configuration is finished.

The data from each step will be uploaded along the way in order to create a smoother transition and faster development.

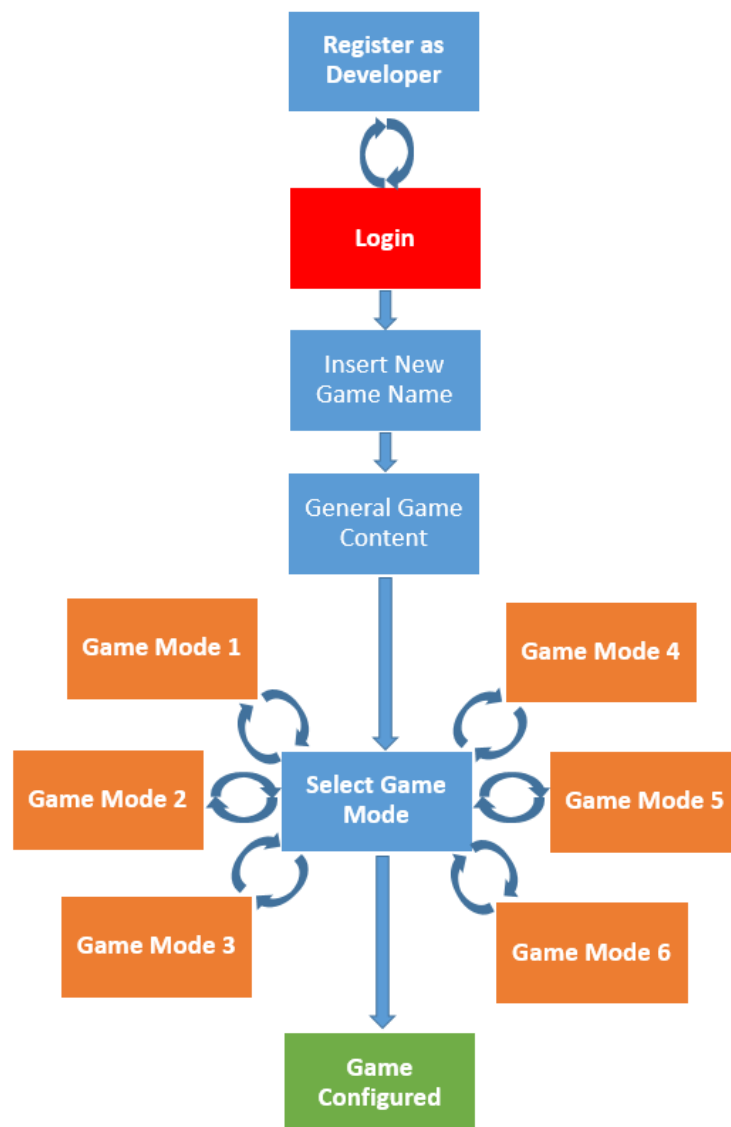


Figure 3.1 - Diagram of a New Game Creation

## 3.2 Player Dimension

Regarding the player dimension, there will be no need for configurations on the players' part, meaning that they will only need to install the Android application on their smartphone and select the game module that they want to play, the rest of the process will be managed automatically.

The way that this dimension will work is by downloading the necessary content for each of the requested game modules. After the corresponding content is downloaded it will be stored locally in the device so that the player won't have to download again all the necessary data, this process will be explained in depth in the chapter Implementation of the Prototype.

After all the content has been downloaded and stored, the game can start, meaning that it will call the game modes in the introduced order with the corresponding content.

Along the gameplay, some information will be collected from the user (if there is an internet connection) that will be stored in the database, this information will allow for a review of some of the game's characteristics, to analyse its divulgation, among other aspects.

## 3.3 Global Dimension

In Figure 2.1 there is a representation of the global sequence of the solution, this sequence includes both Developer and Player.

This sequence will consist of the steps taken in both dimension, having in account that the player cannot start a game without it being previously made available in the game modules list. Another relevant consideration is that despite the fact that there are only one Developer and two players in this representation, this number is not limited, since there can be more users on both sides, both creating and playing the respective chosen games.

Lastly, and since it is not represented in the diagram, the player also performs a sort of login, since they have to introduce their name (only on the first time playing) and this information



is sent to the database (if there is an internet connection). This step is not represented in the diagram because after the first gameplay, it is filled automatically without the need for the player's input.

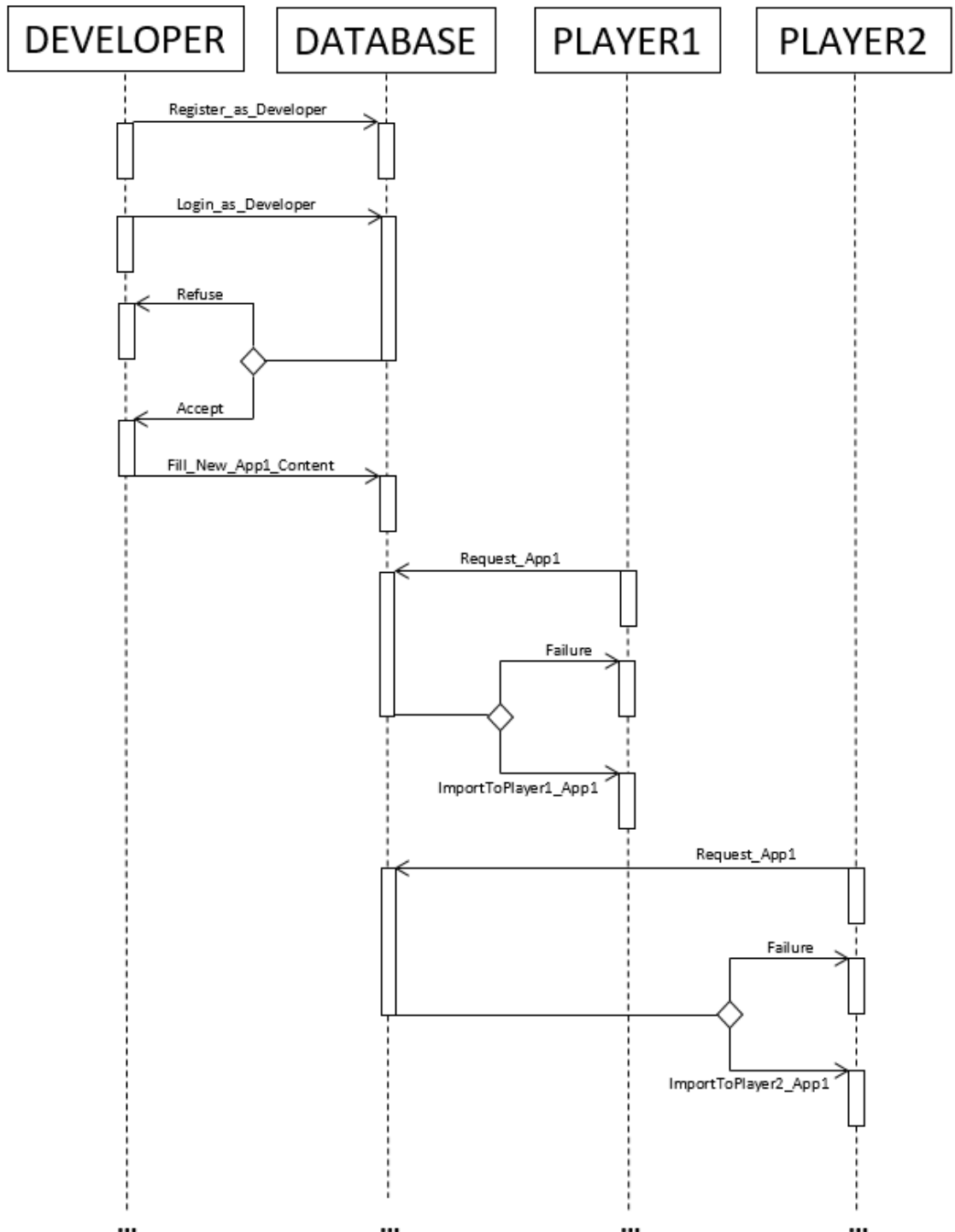


Figure 3.2 - Sequence Diagram



## 4 Implementation of the Prototype

In this chapter, the implementation of a prototype solution will be described, resorting to the content of one of the books from Pais-em-Rede, *Afonso e a Espada Mágica*, presenting both Player and Developer dimensions.

To properly explain the development of the entire solution, this chapter will also be divided in the Developer Dimension and Player Dimension, with a detailed explanation of how they interconnect in the end, each of the dimensions will be subdivided in two layers, the one seen by the Developer/Player (GUI) and the “background” layer that executes the operations chosen by the user (Back-end). Some images and source code parts will be used to explain in depth the way that all the parts come together, this explanation will be somehow more technical since there is a need to show how things work in the “backstage”.

In the end an introduction to the database and server organisation and chosen game engine will be presented to terminate the explanation of how the entire solution works, as well as the tools used for its development.

### 4.1 Developer Implementation

#### 4.1.1 GUI

Since the architecture and step sequence has already been explained previously in the Developer Dimension, this section will consist of the website layout to improve the comprehension of how the solution works in terms of creating new game modes.

The first figure will include both content window and background, the following images will only contain the windows since the background is kept thought all the website pages from start to finish being only an aesthetics component.

A more detailed description can be found in Appendix A – Developer Manual

that consists of a brief and simple manual to help the used to develop a game module, by following the rules and stipulations set out.

In this first screen (Figure 4.1) the user will be able to log in by inserting his/her email and password, this data will be compared with the users registered in the Developer database with the same email and password (md5 hash of the password) and if the data is correct it will proceed to the next step, if not, then an error will appear asking the user to reinsert the text in the fields correctly.

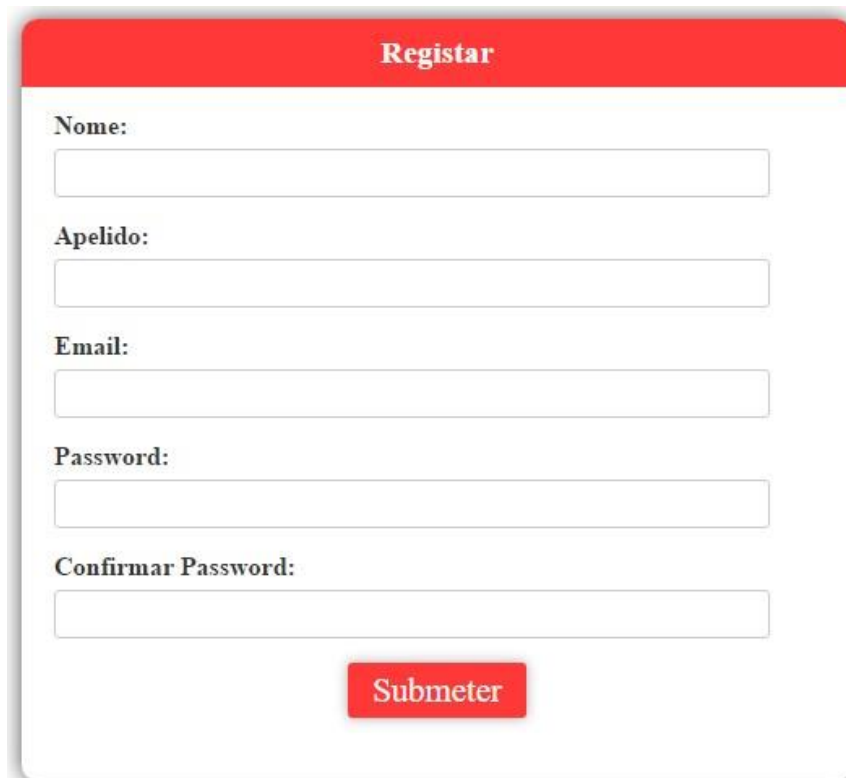
If the user hasn't registered yet, there is also the possibility to choose the option "Registrar" that will take the user to the registration page shown in Figure 4.2, after this page has been properly filled out, then the user can proceed to login with the data inserted in the register screen.



Figure 4.1 - Login Screen

In the screen in Figure 4.2, the user is able to register as a developer, by filling out the fields presented above. After the registration, the user can login and start creating new game modules using his/her login data.

It is important to refer that the games information will be uploaded to the database with the indication of the date, time and developer that created the specified game module to keep track of each one of the created games and each of the specifications.



The image shows a registration form titled "Registrar" in a red header. The form contains five input fields: "Nome:", "Apelido:", "Email:", "Password:", and "Confirmar Password:". Below the fields is a red "Submeter" button.

Figure 4.2 - Register Screen

The screen in Figure 4.3 will have the simple purpose of allowing the developer to introduce the name that will be used for the game module so the user can distinguish between the several game modules created and shown in the game list. This name will also be used as a control variable in multiple situations. After the name has been selected, a new folder will be created in the server to fill with the game module's content and the database will be updated with this new information.

**Novo jogo**

Nome do Jogo:

Submeter

Figure 4.3 - New Game Screen

The screen in Figure 4.4 will be used to fill the general content of the application, the general content consists of some components that are common to all game modules, these components correspond to the background music and right and wrong sounds and images (used to indicate to the user if the answer is right or wrong with the visual and hearing aid).

**Conteúdo geral do jogo**

**Música de fundo (.mp3):**  
Choose File No file chosen

**Som da resposta certa (.mp3):**  
Choose File No file chosen

**Som da resposta errada (.mp3):**  
Choose File No file chosen

**Imagem da resposta certa (.png):**  
Choose File No file chosen

**Imagem da resposta errada (.png):**  
Choose File No file chosen

Submeter

Figure 4.4 - Insert General Game Content Screen

The screen in Figure 4.5 will be the most frequent, since the game mode pages will loop back to this page in order for the user to select the next game mode. The game modes will have to be selected in the order that the developer wants the game to flow and this order will also be stores in the database.

Each of the selected modes has his own series of content to be filled, between questions, answers and images.

After all the game modes have been properly filled out and the developer is satisfied with the configuration, the option “Terminar” can be selected as a way to indicate that the configuration is over and the sequence can be saved.

If the developer happens to make a mistake, he/she can also choose the option “Apagar e recomeçar” to start over the configuration of the game. The option to start over was made in order to reduce the doubt of which configurations were still saved and which would have to be reconfigured. If this option is selected, the content will be erased from the database and the game module folder and respective content will be removed from de server.



Figure 4.5 - Select Game Mode Screen

Following will be all the predefined game modes that the developer will be able to choose from. These game modes have different levels of difficulty and can be configured to easier/harder in accordance with the inserted inputs, like the questions asked or the images inserted in case of the puzzle game mode.

The main focus of this section is to refer the way that the game module's will be created, for a better understanding of how the game modes will work, see the chapter Player Dimension in which the game modes will be explained in more detail.

All the game modes have a specific background image so that the gameplay will be more attractive to the player, this will be one of the specific contents to be filled in each one of the game modes.

In Figure 4.6, the required content for the game mode "Adivinhar a Palavra" will be uploaded, this content consists of a background image (like all game modes), a question, an answer to said question and five images to represent the lives, since each time the user fails to select a right answer, a different image will be used.



**Adivinhar a Palavra**

**Inserir imagem de fundo (.jpg):**  
Choose File No file chosen

**Pergunta a apresentar:**  
[Empty text box]

**Resposta correcta (1 Palavra - Até 10 caracteres):**  
[Empty text box]

**Inserir 5 Imagens das "vidas" (.png):**  
Choose Files No file chosen

**Submeter**

Figure 4.6 - Insert Name - Game Mode Screen

In Figure 4.7, the required content for the game mode "Pares 5x4" will be uploaded, this content will consist of 10 images that will be used in this game, all the images shall be selected at once to simplify the selection and the upload process.



Figure 4.7 - Equal Pares Small - Game Mode Screen

In Figure 4.8, the required content for the game mode “Pares 6x4” will have the same logic as “Pares 5x4” but with two more images, in order to increase the difficulty level. The user also has the option to upload only two images if has selected the game mode “Pares 5x4” since these images are shared between these two game modes in order to reduce the amount of content to be downloaded to the smartphone.

Figure 4.8 - Equal Pares - Game Mode Screen

In Figure 4.9, the required content for the game mode “Linhas cruzadas” will be a question and four images, these images consist of a main image (used as a reference), two wrong answers for the question asked a right answer image.

**Linhas cruzadas**

**Inserir imagem de fundo (.jpg):**  
Choose File No file chosen

**Pergunta a apresentar:**  
[Empty text input field]

**Inserir imagem principal (.png):**  
Choose File No file chosen

**Inserir 2 imagens das respostas erradas (.png):**  
Choose Files No file chosen

**Inserir imagem da resposta certa (.png):**  
Choose File No file chosen

**Submeter**

Figure 4.9 - Lines - Game Mode Screen

In Figure 4.10, the required content for the game mode “Escolha Múltipla” will be a question three answers (one right answer and two wrong ones), these will all be stored in the database.

These are some of the contents will be stores in the Android device locally so that the application will be able to work offline, without having to download the game module content again.

**Escolha múltipla**

Inserir imagem de fundo (.jpg):

Choose File No file chosen

Pergunta a apresentar:

Resposta correcta (1 Palavra - Até 10 caracteres):

Resposta errada (1 Palavra - Até 10 caracteres):

Resposta errada (1 Palavra - Até 10 caracteres):

Submeter

Figure 4.10 - Multiple Choice - Game Mode Screen

In Figure 4.11, the required content for the game mode “Puzzle 3x3” will be a simple image that will be used to create a puzzle.

**Puzzle 3x3**

Inserir imagem de fundo (.jpg):

Choose File No file chosen

Inserir imagem do puzzle(.jpg):

Choose File No file chosen

Submeter

Figure 4.11 - Puzzle Small - Game Mode Screen

In Figure 4.12, the required content for the game mode “Puzzle 4x4” will be the same as the previous game mode, with the difference that the result will be more complex, making the level harder to solve.



Figure 4.12 - Puzzle - Game Mode Screen

At last, Figure 4.13 serves as an indication that the game as been properly filled out and that the submitted information has been correctly updated to the server and database.



Figure 4.13 - Final Screen

## 4.1.2 Back-end

In this section, will be inserted the content from one of the source code files, the remaining files will not be inserted since in terms of code, the requested operations are similar on all screens, consisting of simple operations to get text inputs and inserting/updating them to the database and uploading images and sounds to the server.

The only source code files that have different but relevant functionalities that will not be presented are the Registration and Connection files. The Connection file content will not be showed due to security reasons since it has the required information to login in the database, the Registration file demonstrates some of the security precautions taken, like using md5 to create a hash of the password (in this case, a form of “cryptography” that changes the content so that it can’t be changed back), this way the password can’t be seen by anyone but still be securely stored, and using protection against SQL injection in the text fields (this practice is used in all scripts).

The files used in the website are coded in PHP (execution code), HTML (website components layout) and CSS (website components design).

```
<!DOCTYPE html>
<html >
<head>
<meta charset="UTF-8">
<title>Insert Name Content</title>
<link rel="stylesheet" href="http://stb.uninova.pt/projetos/Book_Story_Game/GameWizard/site_content/site_css/style.css">
</head>
<body>
<?php
session_start();
require('connect.php');

$DevUsername = $_SESSION['UserName'];
$GameName = $_SESSION['GameName'];
$AppID = $_SESSION['AppID'];

if(isset($_POST['Submit']) && !empty($DevUsername))
{
    $Question = mysqli_real_escape_string($db,$_POST['INQuestion']);
    $Answer = mysqli_real_escape_string($db,$_POST['INAnswer']);

    $AllFiles = True;
    $NumImgs = sizeof($_FILES['Img']['name']);

    if($NumImgs==5)
    {
        for($i=0;$i<5;$i++)
        {
            $path = $_FILES['Img']['name'][$i]; //To check if all extensions are correct
            $ext = pathinfo($path, PATHINFO_EXTENSION);

            if($_FILES['Img']['size'][$i]==0 || $ext!="png")
                $AllFiles = False;
        }

        $path = $_FILES['BackgroundImg']['name']; //To check if all extensions are correct
        $ext = pathinfo($path, PATHINFO_EXTENSION);

        if ($AllFiles==True && $_FILES['BackgroundImg']['size']>0 && $ext=="jpg" && !empty($Question) && !empty($Answer))
        {
            $FolderName = $_SESSION['DirectoryName'];

            move_uploaded_file($_FILES['BackgroundImg']['tmp_name'], $FolderName."BackgroundIN.jpg");

            $query = "INSERT INTO `fdgoncalves_InsertNameGame` (`AppID`,`GameName`,`HangQuestion`,`HangAnswer`)
VALUES (`$AppID`,`$GameName`,`$Question`,`$Answer`)";
$results = mysqli_query($db,$query);

            for($i=0;$i<5;$i++)
            {
                move_uploaded_file($_FILES['Img']['tmp_name'][$i], $FolderName."Hang".$i.".png");
            }
        }
    }
}
```

```

        $_SESSION['InsertNameSelected']=1;
        mysqli_close($db);
        header('Location: http://stb.uninova.pt/projetos/Book_Story_Game/GameWizard/site_content/site_php/GameModes.php');
    }
    else
        $ErrorFound = "Conteudo preenchido incorrectamente";
    }
    else
        $ErrorFound = "Numero incorrecto de ficheiros carregados";
}
?>

<form action="InsertNameCont.php" method="POST" enctype="multipart/form-data">
<header>Adivinhar a Palavra</header>
<label><b>Inserir imagem de fundo (.jpg):</b><span></span></label>
<input type="file" name="BackgroundImg"/>
<label><b>Pergunta a apresentar:</b><span></span></label><br>
<input type="text" name="INQuestion">
<label><b>Resposta correcta (1 Palavra - Até 10 caracteres):</b><span></span></label><br>
<input type="text" name="INAnswer">
<label><b>Inserir 5 Imagens das "vidas" (.png):</b><span></span></label>
<input type="file" name="Img[]" multiple />
<div class="centerAlign">
    <button type="submit" name="Submit">Submeter</button>
</div>
<label><?php echo $ErrorFound; ?></label>
</form>
</body>
</html>

```

Figure 4.14 - Website Source Code - Insert Name Script

In Figure 4.14 is the source code from the Insert Name Script, where many relevant information is presented, like the use of `$_SESSION` variables to store data between scripts to validate information, how the images are uploaded to the server by using the `move_uploaded_file()` function and the queries to insert data in the database.

This is the script behind Figure 4.6, where the screen to insert the required data for the game mode “Adivinhar a Palavra” appears. In here we can see how the buttons are laid out using the HTML form, how the CSS script is called to insert the button and webpage layout and the executed commands when the buttons are pressed using PHP.

## 4.2 Player Implementation

### 4.2.1 GUI

This section will consist of the Android application layout to improve the comprehension of how the solution works in terms of selecting and playing the game modules created by the developers.

The first figure (Figure 4.15) consists of a simple introduction screen to the application. To efficiently use the time that this screen is up, in the background some operations will be executed, among which the verification if the device has an internet connection and the gathering of the device's information to be stores, like the devices ID, locations, etc., this will be explained in more detail in chapter 4.4.1 - Structure.

The use of the GPS is not required, but the internet connection is requires the first time of each game module's selection since it is necessary to extract all the necessary data for the game to be played, all the necessary information will be stored in the device via the Assetbundle file that contains the images and sounds and via the PlayerPrefs variable provided by Unity that allow for the storage of strings and integers, which in this case will be used as control variables and to store the question and answer strings.



Figure 4.15 - Application Intro Screen

In Figure 4.16, the player is asked to insert his/her name, this will be the name used to store the player's information in the database.

The name can consist of one or more words and it will also be stored locally on the device, so the player only need to fill his/her name once and this screen will not show up again, unless it is by request, see Figure 4.20.



Figure 4.16 - Insert Player Name Screen

In Figure 4.17, is shown the screen where the player can select which game module he/she wants to play from a predefined list of game modules, this information is also stored in a string locally so the player can choose which game to play even when de device is offline. For more information on this game modules' list see chapter 4.4.1 Structure.





Figure 4.17 - Select Game Screen

In Figure 4.18, the update message is demonstrated, this message appears after the application's version is checked. If the application version isn't the same as the string in the Version column in the database, the update window will appear asking if the user wants to update the application.

Since the application isn't in the Google Play Store ("Google Play," n.d.), if the user chooses to update, it needs to be reinstalled with a more recent version, so it will find the URL column in the database to get the download URL and will open it in the device's browser.

The assetbundle version is also verified, but since it can update without the need for the application to be official, it is made automatically if there is an internet connection.



Figure 4.18 - Update Screen

In Figure 4.19, the player is presented with two options, either to start playing the game, or to access the settings.

In this screen is also when the game module's background music will start being heard, which can be muted using the button in the upper right corner.



Figure 4.19 - Play/Settings Screen

If the player selects the option “Definições”, he/she will be sent to the screen in Figure 4.20, where three options will be presented:

- “Quero tentar bater o recorde” – this option will set the Timer variable to one and the Best Time will be stored online so other players can compete for the best score;
- “Mudar o nome” – this option will send the player back to the screen in Figure 4.16, where the player can change his/her name;
- “Voltar atrás” – this option will return to the previous menu without making any changes.



Figure 4.20 - Settings Screen

In Figure 4.21, the game mode “Adivinhar a Palavra” is shown, in this screen the player must insert the answer to the question shown. For each wrong letter inserted by the player, he/she will lose one life of the five given, these will decrement until the player has no lives left to play.

Each time the user selects a letter, this letter will be marked with a darker color to remind the player of the choices he/she has taken so far. It is also relevant to refer that if the answer word has repeated letter, they will all be filled at the same time, without any need to repeat them.

When the word has been completely filled, the game mode will proceed to the next one, to continue the game sequence.



Figure 4.21 - Insert Name - Game Mode

In Figure 4.22, a transition screen is shown, this screen will be used in every game mode transition and will consist of the entities that participated in the game’s development, as the name

leads to understand it will also be used as a transition element so the game doesn't change between game modes "abruptly".



Figure 4.22 - Logos Transition Screen

In Figure 4.23, the game mode "Pares 5x4" is shown, this game mode will consist of using the ten images uploaded by the developer and duplicating them, so the user will have to find the equal images.

If the user selects two equal images, they will stay turned showing the images, if not, the incognito image will re-appear.

It is relevant to refer that all images will be randomly positioned each time this game mode starts, this way despite the fact that the images are the same, their position will always change.



Figure 4.23 - Equal Pares Small - Game Mode

In Figure 4.24, the game mode “Pares 6x4”, the gameplay will be similar to the one in “Pares 5x4”, the only difference is that there will be more images.

Since the images are shared between both game modes, if the developer chooses to insert these two, he/she will only need to add the remaining two images instead of inserting twelve images.



Figure 4.24 - Equal Pares - Game Mode

In Figure 4.25, the game mode “Linhas Cruzadas” is shown, this gameplay will consist of a question that will be answered using a graphical element (image). To make the game mode more difficult, there will be three possible answers and the player must see which line connects to the right answer and proceed to select the letter to which it corresponds (A, B or C).

This game mode will have two changing elements that will be positioned and chosen randomly, the lines will be selected from a set of three possible line combinations and the answers will also be placed randomly, this way the game mode will change significantly each time the player enters the gameplay.



Figure 4.25 - Lines - Game Mode

In Figure 4.26, the game mode “Escolha Múltipla” is presented, as the name suggests, this game will consist of a question and three possible answer in which the player must select the right one.

This game mode will also have its answers positioned randomly so the gameplay will change each time the user plays.



Figure 4.26 - Multiple Choice - Game Mode

In Figure 4.27, the game mode “Puzzle 3x3” is presented, this game will consist of a randomly placed set of images that are segments of a main image, the player will have to place all pieces in order so that the main image will be shown in the end, the difficult part of the gameplay will be that the player can only move the pieces that are placed in the right, left, top or bottom of the black square, making it so that not all pieces can be moved.



To help the player, he can press the button “?” that will show or hide the main image that should be “built”.



Figure 4.27 - Puzzle Small - Game Mode

In Figure 4.28, the game mode “Puzzle 4x4” is presented, this game will be similar to “Puzzle 3x3” with the exception that the difficulty will be higher due to the number of segmentations of the main image.



Figure 4.28 - Puzzle - Game Mode

In Figure 4.29, is shown the final screen of the game module, if the player selected the option to try to beat the record it will show a different message, either showing a congratulation message if the player beats the record, or showing the name of the player and time of the best person keeping the best score. In both situations the player’s play time is shown.



Figure 4.29 - End Screen

## 4.2.2 Back-end

Following will be some relevant parts of the C# scripts used in the application's development.

Firstly, some code that will be used in several scripts (mainly the game mode scripts), this code is similar in all the scripts in which it's used. In Figure 4.30, the code for storing the players' game data is presented, an URL link is created containing the information to insert in the database.

After all the information is filled, the application will close by using the `Application.Quit()` function.

```

public IEnumerator EndGame() //Send user data do database
{
    if(IntroGWScript.HasInternet)
    {
        string tempString="";
        string FinalURL;
        IntroGWScript.timeSpan = TimeSpan.FromSeconds(Time.time);
        IntroGWScript.GameTime = string.Format("{0:00}:{1:00}",IntroGWScript.timeSpan.Minutes, IntroGWScript.timeSpan.Seconds);
        tempString = WWW.EscapeURL(IntroGWScript.GameTime);
        IntroGWScript.GameTime = tempString;

        FinalURL=IntroGWScript.url_InsertGameData + "?" +
            "IDEntry=" + IntroScript.CurrentIDEntry +
            "&GameTime=" + IntroGWScript.GameTime +
            "&LastModePlayed=" + SceneManager.GetActiveScene().name +
            "&BestTime=" + IntroGWScript.BestTime;

        WWW www_InsertGameData = new WWW(FinalURL);
        yield return www_InsertGameData;
    }
    Application.Quit();
}

```

Figure 4.30 - End Game Source Code



In Figure 4.31, the sound management code is presented, this will allow the switch between sound/mute and music/mute, it will also switch the images to reflect the current state of the configuration.

```
public void Mute() //Change Music icon On/Off
{
    if (SoundManager.MusicState)
        MusicButton.image.sprite = NoMusicSprite;
    else
        MusicButton.image.sprite = MusicSprite;
    SoundManager.instance.MuteMusic();
}

public void MuteSound() //Change Sound icon On/Off
{
    if (SoundManager.SoundState)
        SoundButton.image.sprite = NoSoundSprite;
    else
        SoundButton.image.sprite = SoundSprite;
    SoundManager.SoundState = !SoundManager.SoundState;
}
```

Figure 4.31 - Sound Management Source Code

Another relevant code segment is the one responsible for the management of the game modes' sequence, to check the correct order in which the game modes should be called, a string is imported from the database (later stored locally) containing a letter sequence, each letter is corresponds to a specific game mode. By incrementing the position each time the function is

called, it will see the corresponding char in the string's position and call the SceneManager.LoadScene() function.

```
void NextScene ()
{
    IntroScript.SceneNum++;
    switch(IntroScript.GameSequence[IntroScript.SceneNum])
    {
        case 'A': SceneManager.LoadScene("InsertName"); break;
        case 'B': SceneManager.LoadScene("EqualParesSmall"); break;
        case 'C': SceneManager.LoadScene("EqualPares"); break;
        case 'D': SceneManager.LoadScene("Labyrinth"); break;
        case 'E': SceneManager.LoadScene("MultiChoice"); break;
        case 'F': SceneManager.LoadScene("PuzzleSmall"); break;
        case 'G': SceneManager.LoadScene("Puzzle"); break;
        case 'X': SceneManager.LoadScene("EndScene"); break;
    }
}
```

Figure 4.32 - Next Scene Management Source Code

As it was referred before, some of the game's content is stores locally so the application can work offline, the way it can be stored is by using Unity's PlayerPrefs as shown in Figure 4.33. The necessary content to store are some game control variables and the questions and answers required for some of the game modes.

```
Question = PlayerPrefs.GetString("MCQuestion_"+ChooseGameScript.SelectedGameNum.ToString());
Answer0 = PlayerPrefs.GetString("MCAnswer1_"+ChooseGameScript.SelectedGameNum.ToString());
Answer1 = PlayerPrefs.GetString("MCAnswer2_"+ChooseGameScript.SelectedGameNum.ToString());
Answer2 = PlayerPrefs.GetString("MCAnswer3_"+ChooseGameScript.SelectedGameNum.ToString());
```

Figure 4.33 - Local Data Storage

In Figure 4.34 is presented the way the assetbundle is managed, by using the WWW.LoadFromCacheOrDownload() function, this will check if the assetbundle file has been downloaded and is updated (by getting the AssetBundle Version from the database), if it has not

been downloaded or if it not updated, then the file will be downloaded, this process is made automatically.

```
string FinalBundleURL = IntroGWScript.BundleURL+"_"+ChooseGameScript.SelectedGameNum;

if (Caching.IsVersionCached(FinalBundleURL, BundleVersion))
    textPos.text="Versão actualizada: V"+BundleVersion;
else
    textPos.text="Versão desactualizada: V"+BundleVersion;

while(!Caching.ready)
    yield return null;

WWW Bundlewww = WWW.LoadFromCacheOrDownload(FinalBundleURL, BundleVersion);
yield return Bundlewww;

AssetBundle bundle = Bundlewww.assetBundle;
```

Figure 4.34 - AssetBundle Management Source Code

In Figure 4.35, the way the assetbundle file is managed is shown, for one of the game modes. Since the assetbundle file contains all the images and sounds, the content needs to be extracted before being used so an asset request is made for each of the game contents.

```
if(GameSequence.Contains("D")) //Get Lines Content
{
    string LinesQuestion;

    FinalURL=IntroGWScript.url_GetLinesData+"?"+"AppID="+ChooseGameScript.SelectedGameNum;
    WWW GetLinesData = new WWW(FinalURL);
    yield return GetLinesData;

    LinesQuestion=GetLinesData.text;
    PlayerPrefs.SetString("LinesQuestion_"+ChooseGameScript.SelectedGameNum.ToString(),LinesQuestion);
    GetLinesData.Dispose();

    for(i=0; i<LinesStrings.Length;i++)
    {
        request = bundle.LoadAssetAsync(LinesStrings[i], typeof(Texture2D));
        yield return request;
        texture = request.asset as Texture2D;
        LinesList[i] = texture;
    }
}
```

Figure 4.35 - AssetBundle Content Source Code

The method for extracting the games' images was shown in Figure 4.35, in Figure 4.36 is demonstrated how to extract and associate the sound files required for the game module. The audio file is then associated to an AudioClip file to be used posteriorly.

```
for(i=0; i<SoundName.Length;i++) //Download AudioClips
{
    request = bundle.LoadAssetAsync(SoundName[i],typeof(AudioClip)); //as Texture2D;
    yield return request;
    switch(i)
    {
        case 0: Music = request.asset as AudioClip; break;
        case 1: RightSound = request.asset as AudioClip; break;
        case 2: WrongSound = request.asset as AudioClip; break;
    }
}
```

Figure 4.36 - AssetBundle Sound Source Code

In the last figure (Figure 4.37), the update script code is presented, if the application is not updated (compared with the database version) then a window will show where the user can choose to update the application or not, if he/she chooses to update the application, then it will be closed, and the device's browser will open with the update URL.

```
private IEnumerator UpdateAppVersion() //Opens the browser with the update link (download new version)
{
    FinalURL=IntroGWScript.url_GetUpdateURL+"?"+"AppID="+ChooseGameScript.SelectedGameNum;
    WWW UpdateVersionURL = new WWW(FinalURL);
    yield return UpdateVersionURL;
    Application.OpenURL(UpdateVersionURL.text);
    UpdateVersionURL.Dispose();
    ExitGame();
}
```

Figure 4.37 - Update Application Source Code

### **4.3 Interconnection between Developer and Player**

It is relevant to refer the interconnection between developer and player since this step isn't made automatically.

As it was referred before, there is a sequence of steps necessary to take from the game's creation until it is ready to be played. The operation sequence consists of the following steps:

- 1) Game's creation by the developer;
- 2) The uploaded content is stored in the database (in the respective tables) and in the server (in a folder created for each game module);
- 3) Creation of the Assetbundle file – this is the step that has to be taken manually, the intention was to make it automatically, but Unity doesn't allow the creation of assetbundle files outside the Unity Editor, so this still needs to be made manually. This step is performed by using the code provided by Unity to increase the options presented in Unity 3D, after this option is enabled, the content of is uploaded to Unity manually and the option to create a new assetbundle is selected;
- 4) Upload of the assetbundle file to the respective folder in the server;
- 5) Game module is made available to the player and can now be downloaded and used in the game;
- 6) The player can play the game.

### **4.4 Database and Server**

As mentioned before, to manage all the required data storage and transfer, there was a need to use a database and a server, these were provided by the STB.

To use the database, the MySQL tool was used, due to its simplicity of use, provided tools and open source nature, using PHP as the scripting language to communicate between layers.

Since the server and database were both provided, there was no need for external tools to run the PHP scripts, still Filezilla was used to manage some of the uploads and downloads to the server due to the fact that it allows for a desktop application to manage said data.

## 4.4.1 Structure

When a game is created, the server will create a new folder to contain all the images and sounds required for the game module implementation, all this content will be named properly to serve as a standard, since the application will access the assetbundle and extract the necessary content and use it in the necessary component fields by its name (name and extension).

Regarding the database, it will be divided into six data tables, these tables will have the necessary content to identify the game module, its contents and some information about the players and developers.

The first table that will be mentioned refers to the AppData (Table 4.1), this table will store the basic description content of each of the created game modules, it will also be from the information here stored that the game list will be created for the application.

Table 4.1 - AppData Content Table

Column Name	Description
AppID	Contains the ID of the game module, this ID will be incremented automatically
GameName	Contains the name of the game module inserted by the developer
GameSequence	Contains the sequence in which the game modes will be called
CreatorUsername	Contains the username of the developer that created the game module
Version	Contains the version of the game module to control if the application is updated
AssetBundleVersion	Contains the version of the assetbundle used in the game module to check if it is updated
CreationDate	Contains the date and time of when the game module was created
URL	Contains the URL to download/update the application

Following is the table with the necessary information from the developer (Table 4.2), this table will be used to allow for the developers to login and create new game modules.

Table 4.2 - DeveloperInfo Content Table

Column Name	Description
DeveloperID	Contains the ID of the developer, this ID will be incremented automatically
FirstName	Contains the first name of the developer
LastName	Contains the last name of the developer
Email	Contains the email of the developer
Password	Contains the md5 hash of the password inserted by the developer
RegistrationDate	Contains the registration date and time of the developer

The UserData table (Table 4.3) will be used to store the information of each player every time they play the game (and if they have an internet connection, if not then it will skip this step).

The objective of this table is to keep information for further analysis of several topics, like the preferred game modules, average completing times, application propagation, amongst others.

Table 4.3 - UserData Content Table

Column Name	Description
IDEntry	Contains the ID of the entry, this ID will be incremented automatically
UserName	Contains the name of the player
AppID	Contains the ID of the game module selected by the player
GameName	Contains the name of the game module selected by the player
DeviceID	Contains de ID of the device in which the player is playing the game
DeviceModel	Contains the model of the device in which the player is playing the game
LocationLat	Contains the latitude of the player's location
LocationLong	Contains the longitude of the player's location
DateTime	Contains the date and time in which the player started to play
GameTime	Contains the time that the player spent playing
LastModePlayed	Contains the last mode that the player played, used to see in the player could complete the game or not
BestTime	Contains the players time if he chose to compete with other players
Timer	Contains 0 if the player chose not to compete with other players and 1 if he did

The table InsertNameGame (Table 4.4), contains the required data for the game mode “Adivinhar a Palavra”, that consists of a question and an answer inserted by the developer.



Table 4.4 - InsertNameGame Content Table

Column Name	Description
AppID	Contains the ID of the game module, to link the contents
GameName	Contains the game module's name
HangQuestion	Contains the game mode's question
HangAnswer	Contains the game mode's answer

The table LinesGame (Table 4.5), contains the required data for the game mode “Linhas Cruzadas”, that consists of a simple question inserted by the developer.

Table 4.5 - LinesGame Content Table

Column Name	Description
AppID	Contains the ID of the game module, to link the contents
GameName	Contains the game module's name
LinesQuestion	Contains the game mode's question

The table MultiChoiceGame (Table 4.6), contains the required data for the game mode “Escolha Múltipla”, that consists of a simple question and three possible answers (one right and two wrong) inserted by the developer.

Table 4.6 - MultiChoiceGame Content Table

Column Name	Description
AppID	Contains the ID of the game module, to link the contents
GameName	Contains the game module's name
MCQuestion	Contains the game mode's question
MCAnswer1	Contains the correct answer for the inserted question
MCAnswer2	Contains a wrong answer for the inserted question
MCAnswer3	Contains a wrong answer for the inserted question

The remaining game modes were not included in more tables since they don't contain data that needs to be stored in the database, they only require images and these should not be stored in a database so they were stored in the server in an appropriate folder that allows for easy access and manipulation.

## 4.5 Game Engine

The selected game engine used to develop the solution was Unity 3D ("Unity - Game Engine," n.d.), since it provided the required tools to develop the necessary components of the game.

Some other requirements were taken into account when comparing Unity to other available game engines like:

- The fact that the necessary requirements are all available in the free edition;
- The online support and documentation provided;
- The possibility of running the application via PC;
- The fact that it allows for the development of multiplatform applications;
- The ease of using the Unity Editor;

- The provided tool to create AssetBundles.

A comparison table between game engines can be found in Table 2.2.



## 5 Validation

The final solution was implemented and validated by performing tests to all its functionalities to find eventual bugs in the code or incorrectly associated components also to verify if any changes need to be made to improve the application's and website's performance.

This validation consisted in the creation of several game modules and running them in the application, this way the complete process can be verified and corrected in case of any detected problem. Security tests were also performed to verify the website's liability and protections were added to improve its structure.

Since the final solution was not yet made available to the public, its results are not significant in terms of analyzing the data. Still the base version that was created to study the necessary components, structure and functionalities was already published in a Pais-em-Rede divulgation of the book that was used as a case study for the application, the data collected from this application's divulgation can be used to validate the solution in the sense that it has more than a hundred game play submissions.

Some of the first game data entries are presented in Figure 5.1, this data is collected from the player's results and stored in the database for further analysis.

IDEntry	UserName	DeviceID	DeviceModel	LocationLat	LocationLong	DateTime	GameTime	LastModePlayed	BestTime	Timer
1	Teste	Teste	Teste	Teste	Teste	2016-08-10 18:26:34	Teste	Teste	9999	0
2	Edu	c6593435ddb1c	bq Aquaris M5	38.75332		2016-08-10 20:09:50	4:12	PuzzleSmall	9999	1
3	Francisco	c6593435ddb1c	bq Aquaris M5	38.75311		2016-08-10 20:19:20	9:55	EndScene	479.46	1
5	Tiago	PC	PC	PC	PC	2016-08-11 10:36:51	4:12	EndScene	244.868	1
12	Francisco	c6593435ddb1c	bq Aquaris M5	38.75223		2016-08-11 19:50:27	0:29	InsertName	9999	1
30	Francisco	c6593435ddb1c	bq Aquaris M5	NA	NA	2016-08-12 11:20:06	0:9	InsertName	9999	0
54	Sara	6568099e1d909	OUKITEL K6000	38.75307		2016-08-15 19:06:20	0:37	EqualParesSmall	9999	1
61	Pedro	c6593435ddb1c	bq Aquaris M5	NA	NA	2016-08-16 20:47:56	0:10	InsertName	9999	0
62	Pedro	c6593435ddb1c	bq Aquaris M5	38.75243		2016-08-17 12:27:31	0:48	EqualParesSmall	9999	1
63	Pedro	c6593435ddb1c	bq Aquaris M5	38.75299		2016-08-17 20:03:09	4:2	EndScene	228.803	1
64	Sara	6568099e1d909	OUKITEL K6000	38.75305		2016-08-17 20:11:18	4:22	EndScene	234.366	1
70	Pedro	c6593435ddb1c	bq Aquaris M5	NA	NA	2016-08-19 21:25:18			9999	0
71	Francisco	PC	PC	PC	PC	2016-08-30 15:23:00			9999	0
73	Francisco	c6593435ddb1c	bq Aquaris M5	0	0	2016-08-31 01:49:24	01:39	EqualParesSmall	9999	0
74	Francisco	PC	PC	PC	PC	2016-08-31 12:48:01			9999	1
75	Gy	45f51cf7637e28	samsung SM-A310F	38.6606		2016-08-31 15:05:00	01:39	EqualParesSmall	9999	1

Figure 5.1 - AppData Database Table



## 6 Conclusion and Future Work

### 6.1 Conclusion

The final solution fits the problematic of creating more personalized solutions for the players, by allowing the creation of game modules for each of the intended books.

This type of tool is very relevant in the market, due to its simplicity of use and variety of possible personalization. Since the application that was publicized was not the final solution the conclusions about this topic can only be taken based on the “base” application, that limits the possibilities but can be played in approximately the same way.

The amount of registered entries in the database is a good indicator of the solutions applicability on the actual market, since it represents the number of times the game was played by different players.

As expected there were some problems with the initial architecture since Unity has the limitation of only allowing for the assetbundle file creation in its Editor, making it so that the required process to make the content uploaded (by the developer in the game creation) is available to the player in a completely automatic sequence, without the functionality to create the assetbundle via script, the manual creation of the file is required, creating a dependence in the middle of the process.

Some other solutions to some problems, like the way the content is stored in the device, were studied and its analysis resulted in the selection of the solution considered more appropriate.

## 6.2 Future Work

After concluding this dissertation's solution, it's clear that the application will always be open for improvements, some of those possible improvements are already identified and are going to be presented below:

- Creation of more game modes;
- Placing the application in the official Google Play Store;
- Improving the quality of the images while reducing their dimension;
- Allowing for repeated game modes, this functionality was not implemented in order to reduce the applications global size;
- Adding animations to the application in order to increase its appeal, this solution was not implemented since the solution was created with only the book's content, in order to create animations, there would be a need to involve more people with different knowledge areas to implement this functionality;
- Using education professionals to validate the way the application is structured and its advantages for children, also to help develop new components and solutions to add.
- Adapt the application for other platforms like iOS and Windows phones;

These were only some of the suggested solutions, many other feature and functionalities can be changed and added to improve the final solution.



## 7 References

- “Age of Mythology Heaven: A Guide to Poseidon.” 2016. Accessed November 5. <http://aom.heavengames.com/strategy/articles/18>.
- Alvarez, Julian, and L Michaud. 2008. *Serious Games : Advergaming, Edugaming, Training and More. Idate*. doi:10.1145/1361083.1361093.
- “America’s Army.” 2016. Accessed October 17. <https://www.americasarmy.com/>.
- Anderson, Eike Falk, Steffen Engel, Peter Comminos, and Leigh McLoughlin. 2008. “The Case for Research in Game Engine Architecture.” *Proceedings of the 2008 Conference on Future Play Research, Play, Share - Future Play '08*, 228–31. doi:10.1145/1496984.1497031.
- Bellotti, F., R. Berta, and A. De Gloria. 2010. “Designing Effective Serious Games: Opportunities and Challenges for Research.” *International Journal of Emerging Technologies in Learning* 5 (SPECIAL ISSUE 2): 22–35. doi:10.3991/ijet.v5s3.1500.
- Breuer, Johannes Johannes, and Gary Bente. 2010. “Why so Serious? On the Relation of Serious Games and Learning.” *Eludamos. Journal for Computer Game Culture* 4 (1): 7–24. <http://www.eludamos.org/index.php/eludamos/article/view/vol4no1-2>.
- Bueno, Pedro. 2014. “Game Wizard.”
- “CSI: New York Walkthrough, Guide.” 2016. Accessed November 10. <http://www.bigfishgames.com/blog/walkthrough/csi-new-york-walkthrough/>.
- “Google Play.” n.d. <https://play.google.com/store>.
- Lewis, Michael; Jacobson, Jeffrey. 2002. “Games Engines in Scientific Research.” *Communications of the ACM* 45 (1): 27–31.
- Miller, Leslie, CI Chang, and Daniel Hoyt. 2010. “CSI Web Adventures: A Forensics Virtual Apprenticeship for Teaching Science and Inspiring STEM Careers.” *Science Scope* 1: 4.

- [http://www.gise.rice.edu/documents/ScienceScope\\_CSI\\_2010.pdf](http://www.gise.rice.edu/documents/ScienceScope_CSI_2010.pdf).
- “Pais Em Rede.” n.d. <http://paisemrede.pt/>.
- “SimCity 4.” 2016. Accessed October 23. [http://www.simcity.com/en\\_US/product/simcity4](http://www.simcity.com/en_US/product/simcity4).
- “Social Tech Booster.” 2016. Accessed June 26. <http://stb.uninova.pt/>.
- Squire, Kurt D. 2008a. “Video Game–based Learning: An Emerging Paradigm for Instruction.” *Performance Improvement Quarterly* 21 (2): 7–36. doi:10.1002/piq.20020.
- . 2008b. “Video Game–based Learning: An Emerging Paradigm for Instruction.” *Performance Improvement Quarterly* 21 (2): 7–36. doi:10.1002/piq.20020.
- Susi, Tarja, Mikael Johannesson, and Per Backlund. 2007. “Serious Games – An Overview.” *Elearning* 73 (10): 28. doi:10.1.1.105.7828.
- “The Sims 2 - EA Games.” 2016. <http://www.ea.com/the-sims-2>.
- “The Sims 2 Screenshots for Windows - MobyGames.” 2016. Accessed November 10. <http://www.mobygames.com/game/windows/sims-2/screenshots/gameShotId,86170/>.
- “Unity - Game Engine.” n.d. <https://unity3d.com/pt/>.
- Zook, Alexander, and Mark Riedl. 2014. “Automatic Game Design via Mechanic Generation.” *Aaai*, 530–36. <http://www.cc.gatech.edu/~riedl/pubs/zook-aaai14.pdf>.
- Zyda, M. 2005. “From Visual Simulation to Virtual Reality to Games.” *Computer* 38 (9): 25–32. doi:10.1109/MC.2005.297.

## **Appendix A – Developer Manual**





# Game Wizard

## Manual de Criação de Novos Jogos

**Desenvolvimento:**

Francisco Gonçalves

(fd.goncalves@campus.fct.unl.pt)

**Orientação:**

Prof. Tiago Cardoso

## Índice

Registrar como Developer .....	2
Criar um novo jogo.....	4
Preencher o conteúdo geral do jogo .....	4
Seleccionar modo de jogo a introduzir .....	6
Modo de jogo – Adivinhar a Palavra .....	7
Modo de jogo – Pares 5x4.....	8
Modo de jogo – Pares 6x4.....	9
Modo de jogo – Linhas Cruzadas .....	10
Modo de jogo – Escolha Múltipla .....	11
Modo de jogo – Puzzle 3x3 .....	12
Modo de jogo – Puzzle 4x4 .....	13
Fim da configuração do jogo.....	14

Registar

Nome:

Apelido:

Email:

Password:

Confirmar Password:

Submiter

Fig. 2 - Janela de regista de developer

3

### Criar um novo jogo

Após ter efectuado o registo e login pode agora iniciar a criação de um novo jogo.

Em primeiro lugar será solicitado que introduza o Nome do Jogo, este nome irá ser utilizado para distinguir os diferentes jogos existentes e para armazenar informações relativamente ao mesmo na base de dados.

O nome não deverá ser extenso de modo a ser correctamente apresentado nas diversas situações de utilização, como tal o limite é de 20 caracteres.

Novo jogo

Nome do Jogo:

Submiter

Fig. 3 - Janela de introdução do nome do novo jogo

### Preencher o conteúdo geral do jogo

Seguidamente é solicitada a introdução do conteúdo geral do jogo, este conteúdo consiste no seguinte:

- Música de fundo - Música que irá estar sempre a tocar ao longo do jogo, a mesma deverá ser seleccionada de modo a não criar distrações ao jogador. O ficheiro de áudio deverá ter uma dimensão máxima de 2MB no formato mp3.
- Som de resposta certa - Nos modos de jogo em que existe indicação sonora de resposta correcta, será este o som a ser reproduzido. O ficheiro de áudio deverá ser de duração muito breve (aproximadamente 2 segundos) e ter uma dimensão máxima de 2MB no formato mp3.
- Som de resposta errada - Nos modos de jogo em que existe indicação sonora de resposta incorrecta, será este o som a ser reproduzido. O ficheiro de áudio deverá ser

4

de duração muito breve (aproximadamente 2 segundos) e ter uma dimensão máxima de 2MB no formato mp3.

- Imagem de resposta certa - Nos modos de jogo em que existe indicação sonora de resposta correcta, será esta a imagem a ser apresentada. O ficheiro deverá ter uma dimensão máxima de 1MB no formato png.
- Imagem de resposta errada - Nos modos de jogo em que existe indicação sonora de resposta incorrecta, será esta a imagem a ser apresentada. O ficheiro deverá ter uma dimensão máxima de 1MB no formato png.

### Conteúdo geral do jogo

Música de fundo (.mp3):  
 No file chosen

Som da resposta certa (.mp3):  
 No file chosen

Som da resposta errada (.mp3):  
 No file chosen

Imagem da resposta certa (.png):  
 No file chosen

Imagem da resposta errada (.png):  
 No file chosen

Fig. 4 - Janela de introdução do conteúdo geral



#### Modo de jogo – Adivinhar a Palavra

O modo de jogo "Adivinhar a Palavra", é baseado no jogo da forca, em que é apresentada uma pergunta ao jogador, o qual terá a possibilidade de ter um determinado número de respostas erradas ao adivinhar as letras da resposta. Neste caso para preencher o conteúdo do modo de jogo, terá de introduzir a pergunta a utilizar, assim como a resposta correcta que o jogador terá de introduzir.

O conteúdo que terá de introduzir é o seguinte:

- Inserir imagem de fundo – Corresponde à imagem que irá ser usada como fundo neste modo de jogo, a imagem deverá ter no máximo 1MB e o formato jpg;
- Pergunta a apresentar – Pergunta que será apresentada no modo de jogo (pouco extensa);
- Resposta correcta – Resposta à pergunta apresentada, deverá ter no máximo 10 caracteres;
- Inserir 5 Imagens das "vidas" – Cada vez que o jogador erra na selecção da letra, é apresentada uma referência visual do "decrécimo de vidas" (ex: imagens com 5 corações, 4 corações, ...), as imagens deverão ter uma dimensão máxima de 1MB (cada uma) no formato png.



The screenshot shows a web form titled "Adivinhar a Palavra" with a red header. It contains four sections, each with a text input field and a "Choose File" button: "Inserir imagem de fundo (.jpg)", "Pergunta a apresentar:", "Resposta correcta (1 Palavra - Até 10 caracteres):", and "Inserir 5 Imagens das 'vidas' (.png)". A red "Submiter" button is located at the bottom right of the form.

Fig. 6 - Janela de preencher conteúdo do modo de jogo Adivinhar a Palavra

7

#### Modo de jogo – Pares 5x4

O modo de jogo "Pares 5x4", consiste num jogo de encontrar os pares, sendo que neste caso são introduzidas 10 imagens que serão duplicadas e "escondidas", tendo o jogador de encontrar os pares.

O conteúdo que terá de introduzir é o seguinte:

- Inserir imagem de fundo – Corresponde à imagem que irá ser usada como fundo neste modo de jogo, a imagem deverá ter no máximo 1MB e o formato jpg;
- Inserir as 10 imagens – Imagens que serão duplicadas, as imagens deverão ter no máximo 1MB (cada uma) no formato jpg.



The screenshot shows a web form titled "Pares 5x4" with a red header. It contains two sections, each with a text input field and a "Choose File" button: "Inserir imagem de fundo (.jpg)" and "Inserir as 10 imagens (.jpg)". A red "Submiter" button is located at the bottom right of the form.

Fig. 7 - Janela de preencher conteúdo do modo de jogo Pares 5x4

8

#### Modo de jogo – Pares 6x4

O modo de jogo "Pares 6x4", consiste num jogo de encontrar os pares, sendo que neste caso são introduzidas 12 imagens que serão duplicadas e "escondidas", tendo o jogador de encontrar os pares.

O conteúdo que terá de introduzir é o seguinte:

- Inserir imagem de fundo – Corresponde à imagem que irá ser usada como fundo neste modo de jogo, a imagem deverá ter no máximo 1MB e o formato jpg;
- Inserir as 12 imagens – Imagens que serão duplicadas, as imagens deverão ter no máximo 1MB (cada uma) no formato jpg.

Caso se tenha seleccionado ou se irá seleccionar o modo de jogo "Pares 5x4" não será necessário introduzir as 12 imagens, apenas 2 uma vez que as imagens são partilhadas entre os dois modos de jogo pelo que só é necessário complementar o modo de jogo anterior.



The image shows a web form titled "Pares 6x4" with a red header. It contains two file upload sections. The first section is labeled "Inserir imagem de fundo (.jpg):" and has a "Choose File" button and the text "No file chosen". The second section is labeled "Inserir as 12 imagens (2 se já tiver preenchido o Pares 5x4) (.jpg):" and has a "Choose Files" button and the text "No file chosen". At the bottom of the form is a red "Submiter" button.

Fig. 8 - Janela de preencher conteúdo do modo de jogo Pares 6x4

#### Modo de jogo – Escolha Múltipla

O modo de jogo “Escolha múltipla”, consiste na selecção da opção certa (entre 3 opções possíveis) como resposta à pergunta apresentada.

O conteúdo que terá de introduzir é o seguinte:

- Inserir imagem de fundo – Corresponde à imagem que irá ser usada como fundo neste modo de jogo, a imagem deverá ter no máximo 1MB e o formato jpg;
- Pergunta a apresentar – Pergunta que será apresentada no modo de jogo (pouco extensa);
- Resposta correcta – Resposta correcta à pergunta introduzida, deverá ser apenas uma palavra com no máximo 10 caracteres;
- Resposta errada – Resposta errada à pergunta introduzida, deverá ser apenas uma palavra com no máximo 10 caracteres;



The screenshot shows a form titled "Escolha múltipla" with a red header. It contains the following fields and controls:

- Inserir imagem de fundo (.jpg):** A file selection field with a "Choose File" button and the text "No file chosen".
- Pergunta a apresentar:** A text input field.
- Resposta correcta (1 Palavra - Até 10 caracteres):** A text input field.
- Resposta errada (1 Palavra - Até 10 caracteres):** A text input field.
- Resposta errada (1 Palavra - Até 10 caracteres):** A second text input field.
- Submiter:** A red button at the bottom.

Fig. 10 - Janela de preencher conteúdo do modo de jogo Escolha Múltipla

11

#### Modo de jogo – Puzzle 3x3

O modo de jogo “Puzzle 3x3”, consiste na organização dos segmentos de uma imagens, de modo a que os mesmos fiquem pela ordem correcta, apresentando deste modo a imagem original.

O conteúdo que terá de introduzir é o seguinte:

- Inserir imagem de fundo – Corresponde à imagem que irá ser usada como fundo neste modo de jogo, a imagem deverá ter no máximo 1MB e o formato jpg;
- Inserir imagem do puzzle – Imagem que será segmentada (3x3), deverá ter no máximo 1MB e formado jpg;



The screenshot shows a form titled "Puzzle 3x3" with a red header. It contains the following fields and controls:

- Inserir imagem de fundo (.jpg):** A file selection field with a "Choose File" button and the text "No file chosen".
- Inserir imagem do puzzle(.jpg):** A file selection field with a "Choose File" button and the text "No file chosen".
- Submiter:** A red button at the bottom.

Fig. 11 - Janela de preencher conteúdo do modo de jogo Puzzle 3x3

12

Modo de jogo – Puzzle 4x4

O modo de jogo "Puzzle 4x4", consiste na organização dos segmentos de uma imagens, de modo a que os mesmos fiquem pela ordem correcta, apresentando deste modo a imagem original.

O conteúdo que terá de introduzir é o seguinte:

- Inserir imagem de fundo – Corresponde à imagem que irá ser usada como fundo neste modo de jogo, a imagem deverá ter no máximo 1MB e o formato jpg;
- Inserir imagem do puzzle – Imagem que será segmentada (4x4) deverá ter no máximo 1MB e formado jpg;



**Puzzle 4x4**

**Inserir imagem de fundo (.jpg):**  
Choose File No file chosen

**Inserir imagem do puzzle(.jpg):**  
Choose File No file chosen

**Submiter**

Fig. 12 - Janela de preencher conteúdo do modo de jogo Pares 4x4