



TECHNICAL REPORT OF THE FINAL DEGREE PROJECT

**DEVELOPMENT OF A VIDEOGAME TO IMPROVE THE
READING AND WRITING ABILITIES OF CHILDREN WITH
DYSLEXIA IN A PLAYFUL WAY**

Víctor Colejo Blanco

Tutor: Begoña Martínez Salvador

Índex

1. Technical Proposal	5
1.1 Introduction and motivation of work	5
1.2 Related subjects	6
1.3 Objectives of the TFG	6
1.4 Task planning	6
1.5 Expected results	7
1.6 Tools	7
2. Game Design Document	8
2.1 Game overview	8
2.1.1 Game Concept	8
2.1.2 Genre	8
2.1.3 Target Audience.....	8
2.1.4 Pedagogical objective.....	9
2.2 Gameplay and Mechanics	9
2.2.1 Gameplay.....	9
2.2.2 Mechanics.....	9
2.2.2.1 Economy	11
2.3 Story, Game World and Characters	11
2.3.1 Story.....	11
2.3.2 Game World.....	11
2.3.3 Character: “Granny”	12
2.4 Levels	12
2.4.1 Level “The Apple tree”	12
2.4.2 Level “The train”	13
2.4.3 Level “The cake”	13
2.5 Interface	13
2.5.1 HUD.....	13
2.5.2 Audio, music, sound effects	14
2.6 AI	14
2.7 Technical	15

2.7.1	Target hardware	15
2.7.2	Development software	15
2.8	Game Art.....	15
3.	Project development	17
3.1	Dyslexia research.....	17
3.1.1	Competitors	17
3.2	Realization of the Game Design Document.	18
3.3	Programming.....	19
3.3.1	Main Menu	19
3.3.2	Main Scenario	19
3.3.2.1	Grid	19
3.3.2.2	States of the game	20
3.3.2.3	The game objects	20
3.3.3	Minigames	22
3.3.3.1	First minigame: “The Apple Tree”	22
3.3.3.2	Second minigame: “The Train”	22
3.3.4	Self-regulation of difficulty level	23
3.3.5	Saving and loading data.....	24
3.3.5.1	Sharing data between scenes	25
3.3.6	Adjustment to different screen resolutions	25
3.3.7	Audio in Unity	26
3.3.8	Particles System.....	26
3.4	Art Design	27
3.4.1	Icon	28
3.4.2	Art in Main Menu	28
3.4.3	Art in Main Scenario	29
3.4.4	Art in minigames.....	29
3.4.5	Art in materials	31
3.5	Animations	31
3.6	Sound.....	32
3.6.1	Sound effects	33
3.6.2	Background music	33

4. Results	35
4.1 The game	35
4.2 Testing	36
4.2.1 Verification	36
4.2.2 Validation.....	36
4.2.2.1 Tests with dyslexic persons	37
5. Conclusions	39
5.1 Achievement of objectives	39
5.2 Deviations in the project	40
5.2.1 Deviations in the game	40
5.2.2 Temporary deviations.....	41
5.3 Future work	41
Bibliography	43

Chapter 1

Technical Proposal

Autor: Víctor Colejo Blanco

Title of the TFG: Development of a videogame to improve the reading and writing abilities of children with dyslexia in a playful way.

Summary:

This document contains the technical proposal of the Final Project for the Bachelor's Degree in Video Game Design and Development. The work consists in the development of a didactic video game to improve the reading and writing skills of children with dyslexia. The game is presented as an alternative to tedious traditional exercises, providing players with a fun game experience while practicing in improving their skills. This will be achieved by means of the adaptation of habitual therapeutic exercises to different mini-games that will be adjusted to the progress of the player. The videogame will be an application for mobile devices, developed in Unity3D.

Keywords: entertainment, serious games, children, therapy, dyslexia.

1.1 Introduction and motivation of work

People with dyslexia have difficulties in reading and writing. As students, this difficulty means school failure in most cases. In general, there is no “cure” [1], but there are exercises to improve their skills. However, in many cases the activities are too tedious and, being directed especially towards a children's audience, the children get bored and end up abandoning them [2]. Faced with this, my proposal is materialized in the form of a video game that allows to exercise the abilities of children while they have fun. Currently, there are applications that move paper activities to mobile devices [3]. My goal is to make a similar and free app, taking as a priority the fun factor and the adaptability to the player's progress. The gamification will be achieved by means of mini-games with an important action component, where not only will it be sought to entertain while practicing exercises, but also to improve other abilities such as attention [4], which is usually a big problem for dyslexic people.

1.2 Related subjects

- VJ1215. Algoritmos y estructuras de datos.
- VJ1222. Diseño conceptual.
- VJ1231. Inteligencia Artificial.
- VJ1238. Fundamentos para el diseño de videojuegos didácticos.

1.3 Objectives of the TFG

O1- Adapting therapeutic exercises for dyslexia to mini-games, prevailing its didactic function.

O2- Create an AI that adjusts the difficulty of the levels to the progress of the player, focusing on their errors.

O3- Elaborate an original art, with colorful and childish aesthetics, that forms part of the appeal of the game for a young audience.

O4- Integrate all the previous elements in a mobile app.

1.4 Task planning

Tasks	
Task 1 - Research on similar apps and documentation	15
Task 2 - Technical proposal	10
Task 3 – Game design	15
Task 4 – Art design	60
Task 5 – AI and programming	100
Task 6 – Technical resources	5
Task 7 - Game assembly	25
Task 8 - Playtesting	10
Task 9 – Final memory	50
Task 10 - Presentation	10
Total hours	300

1.5 Expected results

It's expected to develop a demo of a mobile app that contains several levels with games that work on the improvement of reading and writing skills in dyslexic children as a result of research around exercises intended for this audience. It is desired that these games adapt to the capacities that the player demonstrates, instead of being structured in a linear manner. In addition, it is expected to achieve a game with an aesthetic based on original, colorful and childish art.

1.6 Tools

For this project, the following tools will be used:

- **Unity 2017.2.0f3:** the video game engine where the game project will be developed.
- **Visual Studio 2015:** the programming tool used, programming in C#.
- **Adobe Audition CC 2018:** audio editing tool used to manipulate the sounds and music of the game.
- **Adobe Photoshop CC 2018:** 2D drawing tool with which the art of the video game will be made.
- **Microsoft Office Word 2016:** text editor with which all the necessary written documents will be made, including the final report of the project.
- **Camtasia Studio 9:** video editor where videos created to show this project will be recorded and edited.

Chapter 2

Game Design Document

2.1 Game overview

2.1.1 Game Concept

Farmlexia is a 2D game for Android, set in a ruinous farm that the player must improve. To achieve this, the player will have to face a series of mini-games, distributed on the stage, which will have different challenges. These challenges, in addition to amusing the player, are aimed at improving the reading and writing skills of dyslexic players.

In summary, it is an interactive experience that gamifies literacy skills reinforcement activities for dyslexic people.

2.1.2 Genre

It is a serious game since it has an educational purpose: the improvement of reading and writing skills in dyslexic people through activities with an important playable component. In addition, it can also be said that it is a simulation game, since the player will have to manage a farm, although always being a very superficial management.

2.1.3 Target Audience

As we discussed earlier, Farmlexia is a game whose main purpose is to improve the reading and writing skills of its players. In particular, the target audience of the game are dyslexic children. The age range to which this game is intended includes all primary school children, from the first courses, when the children begin to read, to the final courses. Regardless of age, the game will adapt automatically to the capabilities of each player.

2.1.4 Pedagogical objective

Dyslexia, also known as reading disorder, is characterized by trouble with reading and writing despite normal intelligence. Problems may include difficulties in spelling words, reading quickly, writing words, "sounding out" words in the head, pronouncing words when reading aloud and understanding what one reads [5].

The goal of the game will be to provide support to practice in improving reading and writing skills in dyslexic children of different ages. This will be achieved by adapting different reinforcement activities to games. These games will be, as such, activities with letters, but in a visually attractive and interactive environment that presents a background story with a system of challenges and rewards, making the process much more motivating for children.

2.2 Gameplay and Mechanics

2.2.1 Gameplay

The player will be in charge of a farm where he will have to apply numerous improvements to get it ready and beautiful. In order to make these improvements, the player will need money that he will get by performing the different tasks of the farm.

The player's goal is to complete the tasks of the farm to earn money making them, and use that money to continue improving the place. The tasks, accessible from the main stage, will consist of different mini-games in which the interactive game will be mixed with the realization of activities aimed at improving the reading ability of the dyslexic player.

The tasks are accessible through different buildings or objects of the farm, which the player can buy and place on the stage freely. Therefore, making money by doing the tasks will allow the player to have enough resources with which to continue adding elements to the farm, which in some cases will mean unlocking new mini-games and, in others, just adding decorative objects with which to beautify the place.

2.2.2 Mechanics

The game is composed of a set of levels or mini-games, which must be unlocked, and a main stage, which will work as a link between all of them (see Figure 2.1).

In the main scenario, consisting of the plot of the farm, the player can manage their resources and can buy with them new objects to place in the empty spaces of the grid in which the plot is divided. In addition to buying objects, the player can also freely place them in the space and customize the appearance of the farm.

There are two types of objects depending on their function in the game:

1. Decorative objects: elements that can be placed on the stage, but have no value in the game, beyond an aesthetic value and minimal interaction with the player.
2. Objects, "doors" to minigames: in addition to serving to decorate the stage, they allow the player to access the mini-games through them. In general, the access will consist of selecting these objects and accepting the message that appears on the screen indicating the possibility of entering a mini-game. There is the special case of apple trees that, to allow the player to enter the minigame that corresponds to the tree, they must have apples. So that the tree can have apples, the player has to water it and wait a few minutes for the fruit to grow.

Within each minigame or level there is a different game mechanics. The operation of each level will be explained in a later section. However, the player's interaction with the game will be similar in all mini-games, as well as in the main scenario. The interaction will consist in selecting objects or dragging them to another position according to the indications that the game contributes to the player to complete the objective of each level. On the other hand, there will be dialog boxes with buttons that will offer different actions or indications.

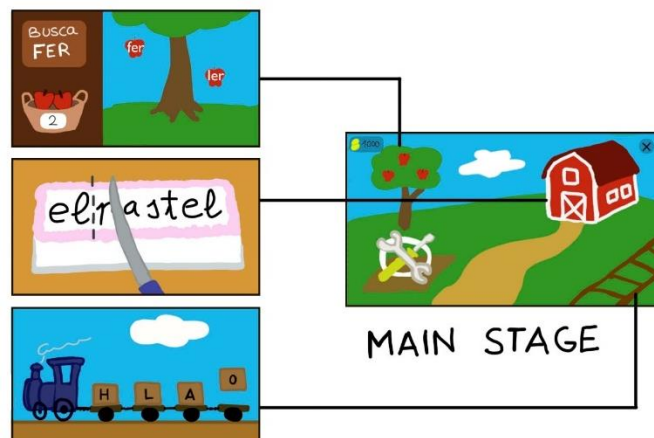


Figure 2.1: Example sketch of the structure of the scenarios.

2.2.2.1 Economy

The economy in the game will be very basic. The player will get coins with which he can pay the costs of the farm improvements. The coins will be obtained as a compensation for the completion of the tasks of the farm.

Within each level, the player will have to perform a repetitive task with a time limit. The score of each level will increase as the player achieves well-done repetitions of the task. For example, in the case of one of the levels, the task will be to collect specific apples. Each apple correctly collected will add one point to the level score. And finally, at the end of the level, the level score will be translated into coins that the player will get.

2.3 Story, Game World and Characters

2.3.1 Story

An endearing granny has a farm in the middle of the field. The farm is in ruins and it is necessary to make numerous arrangements to prevent it from being abandoned and may return to have the prosperity of yesteryear. However, the grandmother has neither much money nor forces to move the farm forward. That's why she publishes a job offer in a local newspaper with the aim of finding someone to help her. The player will be responsible for responding to that job offer and helping the granny to revitalize her farm.

2.3.2 Game World

Once the game starts we will find ourselves in front of a farm plot. Initially the appearance of the lot will be lifeless, with a single house built. The rest of the elements of the farm will be shown in ruins or represented by an area with construction material. In addition to that, it will highlight an apple tree in the plot. This tree will be the starting point of the game, allowing us access to the first level and to obtain the first income in it.

Once the game starts, we will find ourselves in front of a farm plot. Initially the plot will be completely empty and lifeless, waiting for the player to start building on it. In the distance are visible the train tracks through which a freight train frequently circulates. As a starting point we will have to place an apple tree in the plot. From

then on, the scenario will begin to come to life as the player incorporates new elements into it using the money he accumulates for playing the mini-games.

As discussed above, all levels are connected to each other through this main scenario. Special elements, as in the first case the apple tree, will work as a gateway to each level.

As the farm improves, the new elements will bring more life to the world through color and animations, which will generate a sense of dynamism in the environment.

2.3.3 Character: “Granny”

Granny will be the only character present in the game. She will have the typical look of an older woman, with a cheerful face and gray hair. The little information that will be known about her will be through the job offer announcement in the newspaper and through the dialog boxes that appear at specific moments of the game, where the character will give us indications. Granny’s personality will be defined as a loving and grateful person, who will speak to the player in a friendly tone.

2.4 Levels

For this project a demo of the desired game will be made, so the number of levels will be lower than those that would be considered for the complete game.

The levels will consist of mini-games where each one offers a game mode different from the others, but all being games with letters and words with the background theme of a farm. As we discussed earlier, access to the levels of the game is done through the main scenario (farm map) by selecting representative objects of each level, which will function as a gateway to them. In this project the following three levels will be developed:

2.4.1 Level “The Apple tree”

This first level will be accessed through the apple trees located on the map of the main stage.

The level will consist of an apple tree from which apples will be detached continuously. The player's task will be to collect those apples that are in good

condition before they fall to the ground. Each apple will have a syllable or word associated, depending on the level of the player. To distinguish those good apples that have to be picked, at the beginning of the level the syllable or word of those good apples will be indicated on the screen.

2.4.2 Level “The train”

This level will be accessed through the train tracks of the main map.

The stage shows a locomotive followed by several wagons. On each wagon there will be boxes of merchandise marked with one letter each. It will be the player's mission to reorder them appropriately, with the aim of forming words that exist, so that the train can depart.

2.4.3 Level “The cake”

The access to this level will be through the granny's house on the farm.

With the products of the farm a cake has been made. Clients have commissioned it with written sentences and have requested that each word of the phrase be found in a separate pie slice from the rest. The objective of the player will be to cut the cake in the portions that are asked separating the sentences by words.

2.5 Interface

2.5.1 HUD

Before starting the game, we will find the HUD of the main menu, where the buttons "Continue game", "Start game" and "Leave the game" will be.

In the farm scenario, the HUD will contain a bar with the amount of coins of the player in the upper left part of the screen and, on the right side, an "X" button to go to the main menu.

Within each level, the HUD will show on the screen indications about the mission that the player will have to perform, as well as the number of successes achieved and the time lapsed.

2.5.2 Audio, music, sound effects

The background music of the game will be soft with country style, typical of rural areas and countryside.

On the other hand, the ambient sound will be based on the noise of nature and farm animals.

In a game aimed at a children audience, sound is considered a very important factor in order to add attractiveness to the game. The interaction of the player with the buttons or elements that are part of the game will be accompanied with funny sound effects.

2.6 AI

At the beginning of the game, the game will not offer to choose a level of difficulty adapted to the age of the player. Instead, the game itself will adjust the difficulty of the game to the characteristics of the player while he is playing.

The levels of the game will have different levels of difficulty, starting at the lowest. Depending on the number of successes, failures and the time the player uses to react, the AI will decide to increase, decrease or maintain the level of difficulty for the level in which the player is playing.

The variation of the level of difficulty will have different behavior depending on the minigame in question. But in general, in all mini-games the lowest difficulty level will face the player with challenges that contain shorter and simpler words than in the higher difficulty levels.

In the specific case of the level of the apple tree, in addition to what was mentioned above, the variation of the difficulty also affects the rhythm of the game, varying the speed of the apples when falling from the tree and modifying the time it takes an apple to fall after having fallen another before. In parallel, at this level, the AI will take into account the cases in which the player commits more failures to make them repeat more frequently.

2.7 Technical

2.7.1 Target hardware

In Farmlexia the interaction of the player with the game will be by means of the selection and/or drag of objects, being the tactile screens the ideal platforms for it. Therefore, the game will be developed for mobile devices, thinking for those devices with Android operating system.

2.7.2 Development software

The video game will be made with the *Unity3D* engine, using *Microsoft Visual Studio* to write code. For the creation of the art *Adobe Photoshop* will be used and for audio editing, if necessary, the work will be done with *Adobe Audition*.

2.8 Game Art

The graphic style of the game, taking into account that it is a 2D game, will have as references the other games of similar themes, such as the well-known Farmville 2 of Zynga [www.zynga.com]. Considering its audience, colorful and luminous environments prevail, with warm colors and aesthetics with a certain childish air, (see Figure 2.2).



Figure 2.2: Farmville 2 graphic style.

In the part of the mini-games, the aesthetic seeks to be simpler, reducing the number of elements in the scene, enlarging the size of those elements that are shown and highlighting the letters so that they can be seen and understood clearly. In this case, we take as reference the style of applications like *DyetectiveU* (see Figure2.3).

The text font chosen to be included in the project is *OpenDyslexic* [6] because it is a typeface specifically designed to facilitate reading to people with dyslexia.

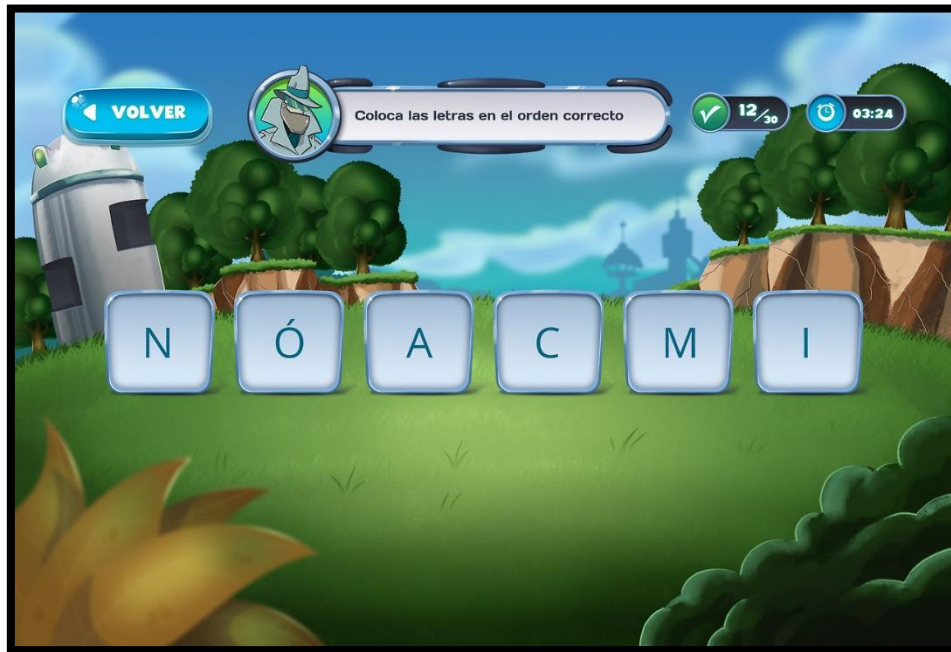


Figure 2.3: DyetectiveU graphic style.

Chapter 3

Project development

3.1 Dyslexia research

Before starting with the project, and taking into account its pedagogical objective, it was imperative to carry out an investigation about dyslexia.

The dyslexia concept turn out to be familiar, although to treat it in a game it was necessary to delve into it. Thus, the research began with the search of a clear definition of this learning disorder.

Once the problem is defined, it needs to be understood from another perspective: that of a person with dyslexia. This was an essential part of the investigation, because when you are going to make a video game to help dyslexic children, firstly you have to put yourself in their place and understand how they live with their problem. This was achieved through consultation in pages specialized in this disorder and with the vision of talks by dyslexic people telling their experience. In addition, the use of dyslexia simulators was very instructive to achieve empathy with people suffering from this disorder.

The research continued with the search of the different ways that dyslexic people have to improve their reading and writing skills. In this part, the research consisted of knowing the exercises that are useful to treat dyslexia and choosing those that could be better adapted to a video game. It was also essential to investigate the applications and videogames with the same purpose, which already existed (as will be explained in the next section), and to know what the video game that was going to be developed for this project could contribute.

Once the research was done, there was enough information to start preparing the Game Design Document.

3.1.1 Competitors

As previously mentioned, within the research process I searched for existing videogames with purposes similar to the proposed project. In general, there is no great variety of applications of this type, nor is it about big-budget videogames. The aesthetic used in all of them consists of colorful scenarios, with characters and environments of childish and funny appearance.

The two main video games aimed at children with dyslexia found are:

- *DytectiveU* [7] is the most complete game that has been found. It includes several reinforcement exercises grouped in different levels, which are accessed in a linear way, unlocking the levels one by one as they are completed. The exercises are presented as tables or boxes with letters or texts with which the player has to interact to meet the objective indicated in each exercise. In this way, the game consists of carrying out activities through a screen, instead of doing them on paper and having a stopwatch that limits the time to complete them.
In addition, there is a brief background story, showing protagonists and antagonists characters, which is presented as comic vignettes, where the aim is to motivate the player to carry out the activities.
- *Dyseggixa* is another game aimed at mobile devices that includes several activities that work with different types of exercises to reinforce reading skills. These exercises are done with a similar game mechanics, using cubes with letters with which the player has to interact, for example, to form words, separate them or change letters by others to form correct words.
This game does not include a story with which to motivate the player, but it does present a funny penguin, whose appearance is customizable by means of "points" that the player obtains when carrying out the activities of the video game.

3.2 Realization of the Game Design Document.

As a previous step to start developing the game, the Game Design Document, corresponding to Chapter 2, was elaborated. In that document it was specified what the game was going to consist of, how it was going to be structured, its aesthetics and, ultimately, all the necessary content to be able to understand and develop the video game.

3.3 Programming

In this section we will discuss the different parts of the game where the programming and the operation of each of them are involved.

The game is structured in several scenarios, starting with a menu that gives access to a main game scenario. From that main scenario you can access other scenarios that consist of mini-games. The change between levels is managed by a script that loads the scenarios simultaneously showing a bar with the progress of the load.

Due to the fact that each scene has a singular operation with respect to the rest, this section will be divided first into sections corresponding to the game scenarios, where the implementation of the programming made in each of them will be explained.

The project is accessible through the following link to a repository:
<https://github.com/viccob/Farmlexia-Game>

3.3.1 Main Menu

The main menu of the game, which appears when it starts, allows you to start a new game after asking to enter the name of the player as a prerequisite.

In cases where a saved game already exists, the game will be displayed with the player's name and the date of the last save. In this case, the game can be eliminated by clicking on the trash can button that appears in the interface to start a new game, or pressing the button with the "Play" symbol to load the main scenario of the saved game.

The menu also contains an "Exit" button that allows you to close the application.

3.3.2 Main Scenario

The main behavior of this scenario is done through the *MainSceneManager* script, which manages the different stage states that it passes through during the game.

3.3.2.1 Grid

When the stage is loaded, a grid is generated consisting of cells where objects can be placed or, if there is a saved game, where the previously placed objects will be loaded. The visibility of the grid will depend on the game state in which the game is located. The grid is programmed to, depending on the position that an

object occupies within it, objects are displayed simulating being in a 3D world, despite being a game in 2D. This is achieved by varying the scale of the objects and the order of priority of their rendering on screen (`Renderer.sortingOrder`) in order to simulate distance in space, since the objects located in the upper area of the grid will be shown smaller and behind the objects located in the lower rows, creating distance sensation.

3.3.2.2 States of the game

In this scenario, three main game states will be managed:

- *“In game” state*: is the normal state of the game, through this state the player can interact with the objects of the scenario. This interaction will depend on the type of object in question, as will be explained later.
- *“Placing” state*: this status is accessed by pressing the blue button that appears in the interface. It allows to visualize the grid and to select the objects to move them to other cells of the grid or to eliminate them in the trash. The grid will be shown partially transparent, except in the cell where we are dragging the selected object, which will be shown in green. On the other hand, the cells adjacent to the cell on which the selected object is dragged will be displayed with less transparency than the rest. Finally, the cells that are already occupied, on which we will not be allowed to place objects, will be shown in red. During the dragging of objects, these will automatically adapt to the cells on which they are moved simulating a 3D space, as it was explained in Section 3.3.2.1.
- *“Building” state*: this status is accessed from the yellow button of the interface. Displays a sliding bar with the objects that can be bought and placed in the grid. Items that the player can not pay will appear transparent.

There are other less relevant states that control the movement of the camera when loading the scene or the states in which scrollable messages appear on the screen to load another level or at the beginning of a new game.

3.3.2.3 The game objects

Except for exception, all objects in the main scenario can be purchased, placed, moved or deleted from the scenario grid (function managed through the `MovableObject` script). That will be the player's common interaction with all objects. However, in addition to the above, each object of different types has its

own interaction with the player. Although in general you can distinguish between objects whose interaction with the player allows access to scenarios of mini-games and those other objects that only interact with some animation.

For the demo of the developed game the following objects have been included:

- *Train Station*: except for the rest, the station can not be placed freely and is automatically established in a scenario location. Nor can it be moved or eliminated. The interaction with this building is that, by selecting it, it allows loading the corresponding scenario to the minigame called "The Train".
- *Apple tree*: this object can be presented with three different states (see Figure 3.1), behaving differently depending on the state in which it is at each moment. When the tree is without water, it will ask for water through an icon that will be displayed on top of it. The player touching the tree will make a system of particles, which simulates water drops, water the tree. This will lead to the next state, in which the apple tree is watered and a descending time counter is established. When the counter reaches zero it will change to the last state, in which apples will grow in the tree and the interaction that occurs when the player touches the apple tree will allow loading the corresponding scenario to the minigame called "The Apple Tree".

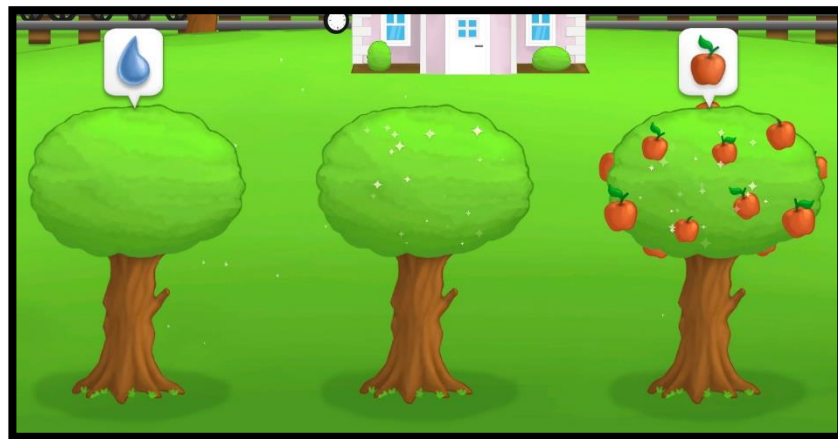


Figure 3.1. Apple trees in their three possible states.

- *Pig*: is a decorative object. The interaction that takes place when touching the pig will consist in allowing the player to caress the animal and make it respond to the caresses thanks to an animation.

3.3.3 Minigames

For this project we have made the demo of the proposed game, which contains two mini-games. In Farmlexia the order of the mini-games does not mean that the first ones are easier than those levels that are accessed later. Each mini-game deals with a different kind of reading reinforcement exercise and it is within each of them that there is a mechanism that is responsible for regulating the difficulty of each game automatically. This part, because of its importance, will be explained individually later (Section 3.3.4).

3.3.3.1 *First minigame: "The Apple Tree"*

This mini-game, as already explained in Section 2.4.1, consists of a tree that throws apples with syllables, the player having to select the apples with the correct syllable, which is indicated at the beginning of the mini-game. The behavior of this scenario is managed by the *AppleLevelManager* script.

When starting the scene, a group of syllables is selected with which to perform the game. Within that group, the syllable that the player should look for in the apples that fall from the tree is assigned. The process that determines the syllables that will fall from the tree is not a completely random process. Within the group of words a percentage is assigned to each of the words, which will control the probability of each of them leaving. Initially, the probability that the correct syllable comes out is set at 40%, while the rest of the words are distributed equally 60%. In spite of this, it is controlled that the correct syllable appears every a certain number of apples, happening this if it does not happen before probabilistically. As the game progresses, making mistakes in a certain word will increase the probability that the word comes out again, guaranteeing a minimum of probability for all the others. This aims, once a word that is more difficult to distinguish for the player is detected, to focus on that word through its repetition and achieve with the repetition help the player to assimilate it better.

3.3.3.2 *Second minigame: "The Train"*

The second mini-game, explained in detail in Section 2.4.2, consists of a scenario where freight trains appear loaded with boxes of letters. These letters form words, but they are disordered. The player has to order them correctly to get the maximum number of trains completed within a time limit. The behavior of this minigame is managed by the *TrainLevelManager* script.

At the beginning of the game, you access a large group of words grouped by the level of difficulty they present. Depending on the level of the player, the words of the group of difficulty corresponding to him are selected and a freight train is

generated. The train is assigned the word chosen so that it is built with the same number of wagons as the word has. In addition, a function is responsible for messing up the word to introduce it in the wagons of the train. Once the wagon appears on the scene, the *ScaleScenario* script adjusts the game scenario to the length of the train so that it can be displayed correctly within the limits of the screen. During the game, the player interacts with the train and can pick up and drag the boxes of goods that the wagons carry. In this way it is possible to reposition the boxes from one car to another simply by dragging them and releasing them on any desired wagon.

For each train that is created it is checked continuously if the order of the letters of its wagons corresponds to the correct word to be formed, completing the train in the affirmative case. Each time it is completed, a new train is automatically generated. Internally there is a timer for each train that detects if the player is taking too long to complete it. In case of taking too long, to avoid breaking the flow of the game, the train leaves the scene, destroying itself after a while, and a new one is generated.

3.3.4 Self-regulation of difficulty level

One of the most distinctive parts of the game is the way to vary the difficulty within the mini-games in order to adapt it to the level of each player.

While the player plays facing the challenges of each mini-game, all his progress is being analyzed at the moment. From this analysis, the game may decide to vary its difficulty. The change of difficulty will be experienced automatically during the game, adapting simultaneously to the level that corresponds to the player based on the skills that this demonstrates while completing the challenges of the mini-games. Once the minigame is finished, the skill level of the player will be saved in order to maintain a difficulty adjusted to his characteristics when he returns to play the same minigame next time.

Because each mini-game is totally different, the way to regulate the variation of each difficulty level will also be different:

- In the first mini-game, "The Apple Tree", the factors that determine the level of difficulty are the speed with which the apples fall and the space between the time an apple falls and the next.
The decision to change the difficulty depends on different conditions, which vary depending on the level of difficulty of each moment. This is managed by a decision tree that takes into account the number of hits and failures committed each time the function responsible for adjusting the difficulty is called during the game, which occurs once every certain number of apples fallen.

In this way, when the player makes frequent mistakes, the level of difficulty is reduced, while, if the player demonstrates great skill in completing the minigame, its difficulty will be adapted to him, increasing it.

- In the second mini-game, "The Train", the level of difficulty is determined by the size of the words that the player has to order. Because the longer the words are, the greater number of possible combinations will exist at the time of messing up their letters.

Another decision tree will be in charge of deciding when to change the difficulty level of the minigame. For this, it will use a time counter that starts each time a train is generated. With this counter, the game determines how long it takes the player to order the word of the train. If the train is completed in a very short space of time, a combo will be counted, which will result in an extra score and, in addition, with the accumulation of three successive combos, it will also mean an increase in the difficulty level of the game. On the contrary, as discussed in Section 3.3.3.2, if the time to order a word is exceeded, the train will leave the scene without being completed and will suppose the descent of the level of difficulty, whenever this happens frequently during the game.

3.3.5 Saving and loading data

In Farmlexia the game is automatically saved each time a change is made in the game. The data is stored in a file on the disk. Among the different stored data is the name of the player, the objects created with their positions within the grid of the main stage, the amount of coins that the player has and the date of the last saved. It should be noted that the functionality of the date is not only to appear next to the game saved in the main menu. When loading the game, the date allows the status of some objects to be updated, such as trees, which depend on a timer. In this way, even if the application is closed, when the timer is reloaded, it can simulate having continued working since the last time it was played.

After considering the use of the *PlayerPrefabs* objects that Unity offers, finally the decision was to serialize a class with the data that was going to be saved to store them in binary form within a file, a solution that offers more security and is equally easy to implement. The script that manages the stored data and that contains the *Save* and *Load* methods is called *GameData*.

With the *Save* method, data stored in a serializable class is serialized and stored in a binary file with the extension *.gd* in the address indicated by the *persistentDataPath* method, which makes it easier to find an address to store the data.

The *Load* method does just the opposite by loading, if there is a saved file, all the information stored on disk. In this way, the work of this method will be to deserialize the binary file and restore the saved data.

3.3.5.1 *Sharing data between scenes*

The *GameData* script also serves to save the data that the game wants to keep when changing from one scene to another. Typically, in Unity all objects and data in a scene are deleted when a new one is loaded. However, there is a method called *DontDestroyOnLoad* that, when called within a class that is assigned to a *gameObject*, allows that when loading a new scene the *gameObject* is not eliminated and remains in existence. In this way the data of that *gameObject* can be shared between scenes without major problem.

3.3.6 Adjustment to different screen resolutions

Being an application designed to run on any of the mobile devices with the Android operating system, it is important that it can be displayed correctly in all of them. There is a great diversity of devices that could run the application, starting from different models of mobile phones to tablets, all of them with very different screen resolutions and that at first can not show the game in the same way. The goal, therefore, is to adapt the game to any mobile device regardless of the size of its screen or its resolution.

To adjust the HUD, allowing it to be displayed in a fixed way on any type of screen, Unity offers a component for the Canvas, called *CanvasScaler*. Within this component it is allowed to modify some parameters from the Unity editor interface itself. In this way, the *Scale With Screen Size* option was established in *UI Scale Mode* and some other options were configured (see Figure 3.2).

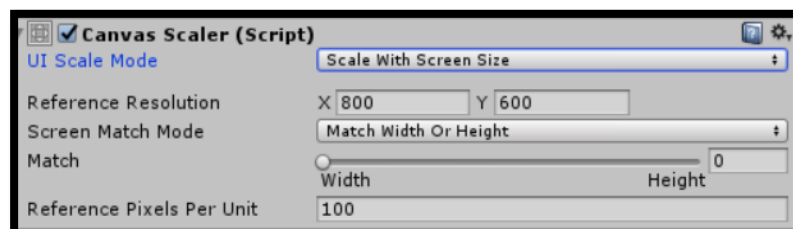


Figure 3.2. *Canvas Scaler Script configuration window.*

To adapt the rest of the *gameObjects* that are not part of the Canvas to the different screens, the *ScaleScript* class was programmed. This script modifies the scale of the objects to which they are assigned, adjusting it to the screen resolutions of each device. From the Unity interface it is possible to individually select whether

you want to scale only the width of the object or, on the contrary, the height of the object (see Figure 3.3).

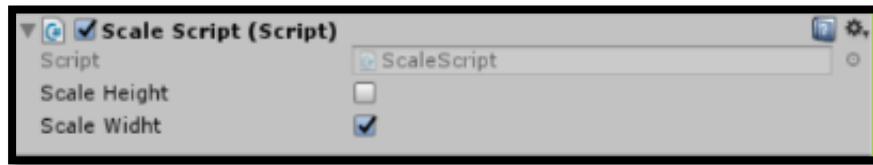


Figure 3.3. Scale script configuration window.

3.3.7 Audio in Unity

To play audio clips in Unity scenes, the components called *Audio Source* have been used, where different clip playback options have been configured, such as the volume of the audio or its repetition. To manage the sound, the *AudioScript* script has been created, which contains public methods that allow reproducing all the sound effects of the videogame when called from other classes.

Within Unity there are audio optimization options that allow you to modify the load time and the size of the audio files in the project. Here has been established the compression format of the sound, the quality and the load type of the audio file, which will be done at the same moment that the scene is loaded.

The process of selecting the audios and their preparation before entering the project in Unity will be explained in detail later on.

3.3.8 Particles System

To create effects with particles in the video game, the particle system offered by Unity has been used, where it is possible to configure the parameters of the particle system comfortably. In the game, particle systems have been used several times to generate visually attractive effects on the stage. For example, the uses that have been given to the particles within the video game have been to simulate the drops of water with which the trees are watered, the fireworks that occur when some objective is completed, the brightness of the trees irrigated or the pollen that floats on the stage as a decorative element.

3.4 Art Design

This section will deal with all the artistic work carried out in the development of the video game. Farmlexia is a 2D game and, therefore, all the art created consists of 2D art. As already mentioned in Chapter 1, the main tool for the creation of art is *Photoshop*, with which I have worked using a *Wacom* graphic drawing tablet [8].

Art has been considered a very important part in the development of the videogame, since it is a game aimed at children, it must be an element that attracts them and motivates them to interact with the game and its activities. To achieve this, bright and cheerful colors with children's forms have been applied in the developed art. In general, the aspect that has been sought is to simulate the pages of the stories of the fables, where it is common to find drawings of animals and environments of nature, as it happens in the scenarios of the videogame.

The main color range of the game will be composed of those colors common in nature: green, blue and brown.

All the elaborated art is grouped in several Texture Atlas (see Figure 3.4) of a size of 2048x2048 pixels each, considered a more efficient size because it is power of 2 and it is not necessary to rescale the texture at execution time [9]. In this way we seek to optimize the game and reduce the number of image files with textures, thereby reducing the weight in memory of the final game.

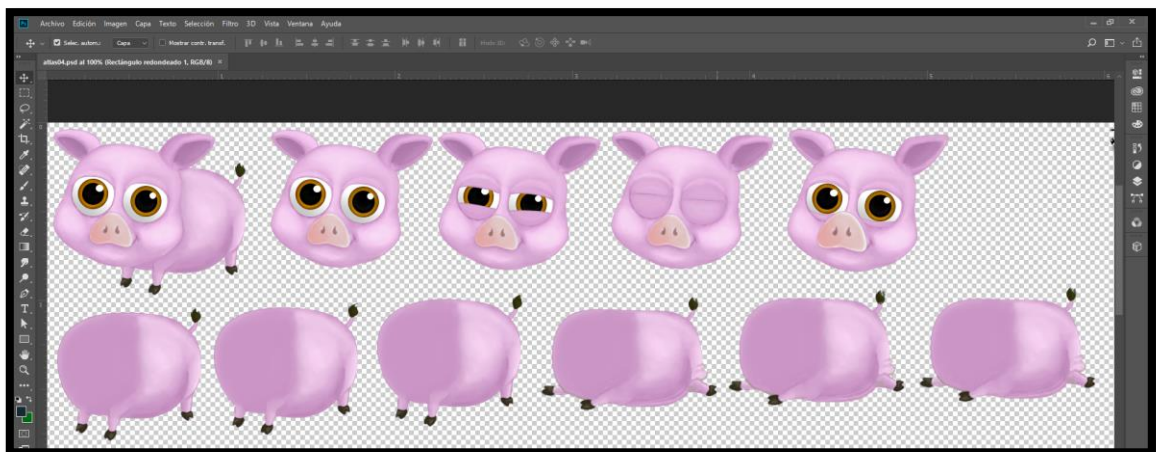


Figure 3.4. Screenshot of a section of one of the atlases used in the game. Contains the sprites with which the animation of the pig is built.

3.4.1 Icon

Even before the application opens, the first thing the player will see will be the video game icon. Therefore, this small image is about the letter of presentation of the game. The icon depicts a representative image of the video game with a cheerful and childish character surrounded by cheerful colors, corresponding to the range of colors used in the game's scenarios (see Figure 3.5).



Figure 3.5. The game icon.

3.4.2 Art in Main Menu

The main menu consists of a simple scenario, where the title of the game appears (see Figure 3.6), the main element of the scene that is accompanied by a bee and the different sprites used for its animation. The rest of the art corresponds to the menu buttons and some basic objects, such as the tree or the clouds, which are frequently reused in other scenes.



Figure 3.6. Screenshot of the Main Menu screen with the title.

3.4.3 Art in Main Scenario

The main scenario (see Figure 3.7) is the scene that contains the most amount of art in the game. This is due to the great variety of objects, some of them with several sprites or animations, which can be placed on the stage and to the different types of game states that show, depending on each state, different elements of art such as the page of a newspaper, windows to select levels, the grid that divides the map into cells or the construction bar with all its art components.



Figure 3.7. Screenshot of the main stage with some objects placed.

3.4.4 Art in minigames

The minigames share common artistic elements, such as the stopwatch located on the HUD or the messages that appear on the screen when performing actions correctly. However, each level has art created specifically for itself.

For the level of the apple tree, shown in Figure 3.8, the boxes of apples, the sprites that are used for the animation of the shaking of the tree crown and the sprites that animate the apples when they fall from the tree or split on the ground have been drawn.



Figure 3.8. Screenshot of the mini-game "The Apple Tree".

For the level of the train a locomotive, the wagons that follow it and boxes of merchandise have been drawn among other objects (See Figure 3.9).

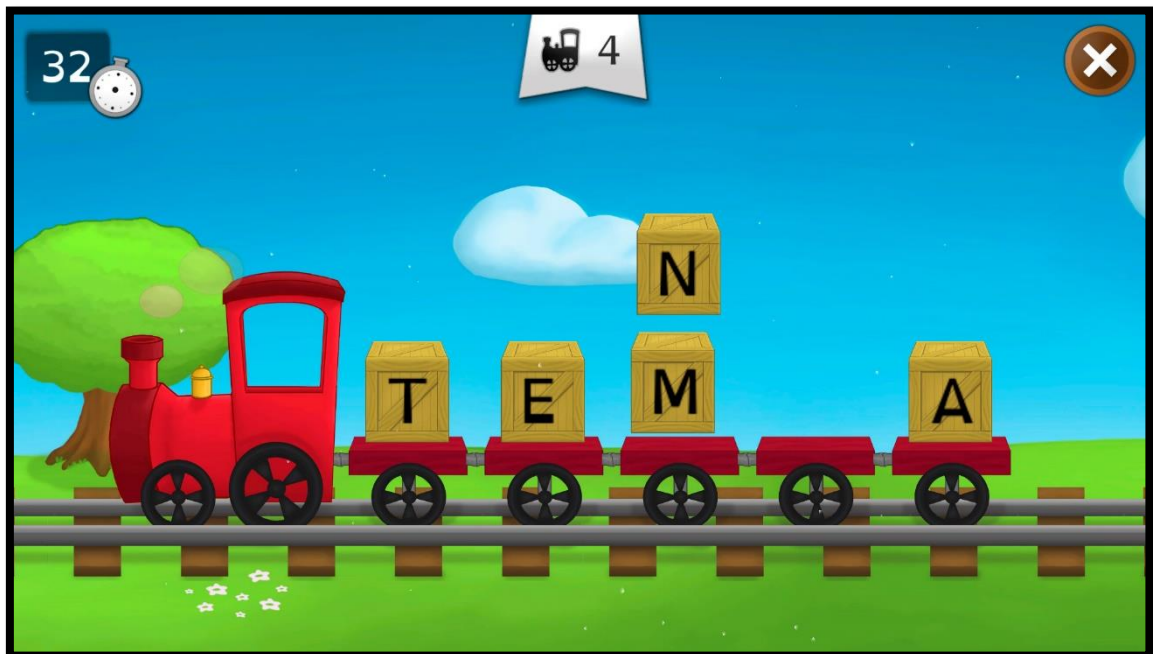


Figure 3.9. Screenshot of the mini-game "The Train".

3.4.5 Art in materials

In addition to the atlases used for the sprites of the elements of the game, individual textures have also been developed to be used in the creation of materials.

In Unity the materials are necessary to be applied with components of *Mesh Renderers* type or in *Particle Systems*, allowing to determine how an object will be rendered.

In the game the materials have been used to apply in particle systems. As we discussed earlier, there are particles that simulate water or brightness. The textures created for this are those of a drop of water and a figure of four points that simulates a shine. These textures are assigned to the Albedo map of the material corresponding to each of the particle systems that are intended to be achieved. Finally, in the renderer section, within the particle system component, the desired material for the particles is assigned.

3.5 Animations

Animations are an indispensable part of the game. With them it is possible to give life to the static figures of the stage, besides generating behaviors in the objects that are fun and complement the visual appeal of the game.

The medium used to make the animations has been Unity's *Animator* tool. With it, practically all the objects of the game have been animated, from the levitation of the title of the game located in the main menu, to the different animations that define the behavior of the animals of the video game.

Unity provides a timeline where "keys" points are established. These points define the visual state of an object at a particular moment within the animation. Among some of the possibilities of this tool, which have also been used for this project, is the modification of the position, scale or rotation of the objects, as well as the variation of the color, transparency or change of the sprite of the object that is being animated.

Not only have the animations been created, also, through the *Animator Controller* and some scripts, the different animations of each object have been controlled, managing the changes between animations through established parameters or defining at which point of the animation begins to reproduce this, among other things.

An example, shown in figure 3.10, we have in the case of one of the most common objects of the game: the apple tree. As previously mentioned, the apple object has many different states that define its behavior. Consequently, the appearance of the tree in the game varies depending on the state in which it is. For this object there are seven animations. Several of them define the movement of the tree when being dragged by the player and the other animations reproduce other concrete cases like when the

apples grow in the tree, when the apple tree is selected by the player or simply when the tree is in a normal state. Three different parameters are used to manage the change between animations, which are modified via script.

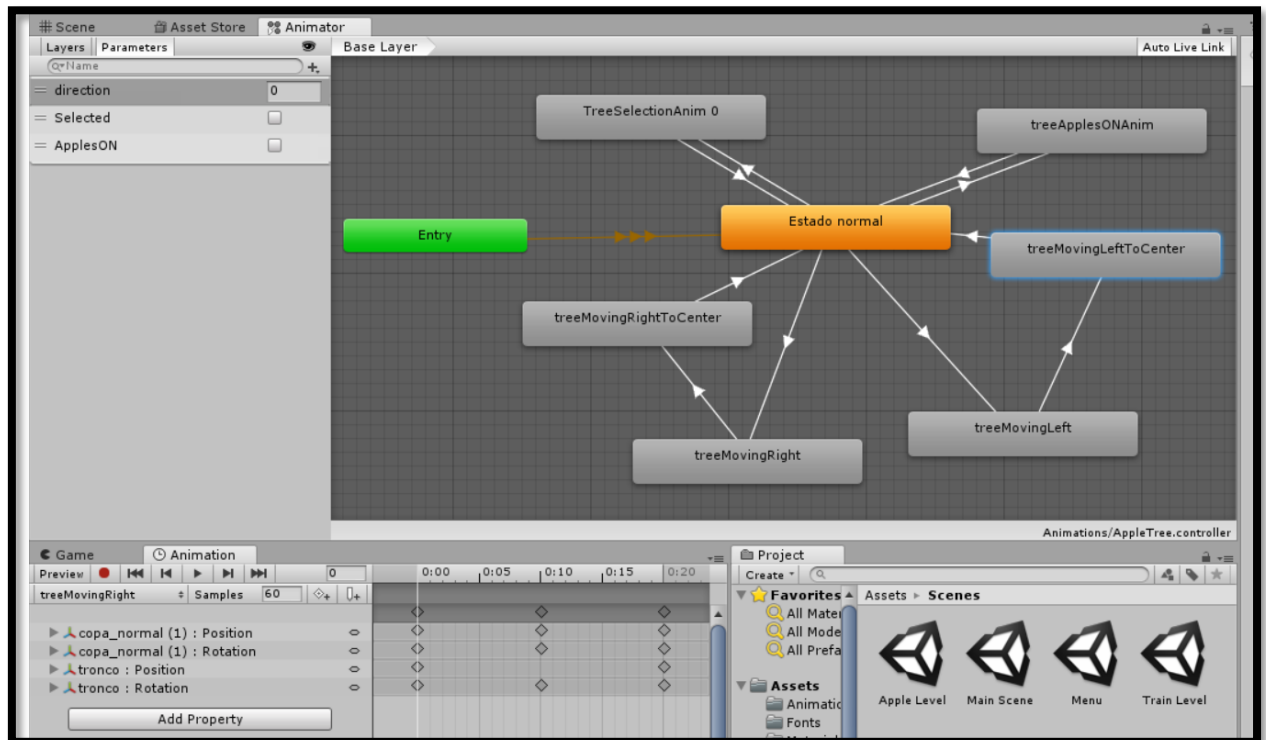


Figure 3.10. Graph of the apple tree animations along with the timeline of one of the animations in the lower part of the capture.

3.6 Sound

Like the images and animations, sound is a part of great importance in the video game. The sounds complement what the player sees on the screen. This is achieved through audios that allow to transmit the states of the game to the player, help to understand what is happening or complete the aspects that define characters or objects thanks to being able to relate sounds with them.

In this video game we can distinguish between two types of audios: sound effects and background music.

3.6.1 Sound effects

The sound effects are used to deepen the environment of the video game world, allowing us to hear the characters or animals for example. This type of audios also have an even more important function, because they serve to help the player to better understand what is happening in the game. This means that by means of sound the player can distinguish when he makes mistakes or successes, when he presses a button or performs some determined action.

An example of sound effects is given when the player moves objects through the grid of the main stage. When the objects are grasped, a sound effect is generated that tries to simulate the sound of lifting a thing from the ground and, on the contrary, when placing the object in a cell, a sound effect is generated (the same as the previous one, but inverted) that conveys that the object has landed on the ground.

3.6.2 Background music

In the video game, two different background music are used. The first corresponds to the background music used on the menu screen and on the main stage. The second music is the one used in the scenarios of the mini-games.

The first music consists of quiet and positive music. It transmits the peace that there are two scenarios in which it is used, since in them the game takes place without any type of action or element of stress. In the case of the music of the mini-games, a change is perceived. This audio continues to have a slow rhythm with a childish tone, however it no longer transmits tranquility. With this new song the player is activated, because he perceives that something is going to happen. And thinking that the player is right, because he is in the scenarios of the minigames where he has to act to complete the challenges before the time runs out.

The music and sound effects used in the video game come mainly from web pages dedicated to the exchange of audio files for free and legal. Freesound [10] is the web page that has been most useful in the search for audios, from which most of the videogame sounds come.

Once the desired audio was obtained, it was moved to the audio editing program *Adobe Audition* (see Figure 3.11), where the sounds were edited to suit the needs of its function in the game. The modifications consisted of segmenting the audio, varying its duration, applying noise elimination and inverting the sounds in some cases. Another function carried out with this program was the audio optimization. The optimization consisted of converting all the sound files to the MP3 format, a format that reduces the

quality in a little perceptible way, but that reduces even more the size of the audios, occupying little space in the disc. So great is the reduction when converting audio files from formats like WAV to MP3, that in one case a file of 6 MB varied its size to occupy barely 500 KB.

Once selected, edited and optimized, the audios were introduced to Unity, part that was explained in Section 3.3.7.

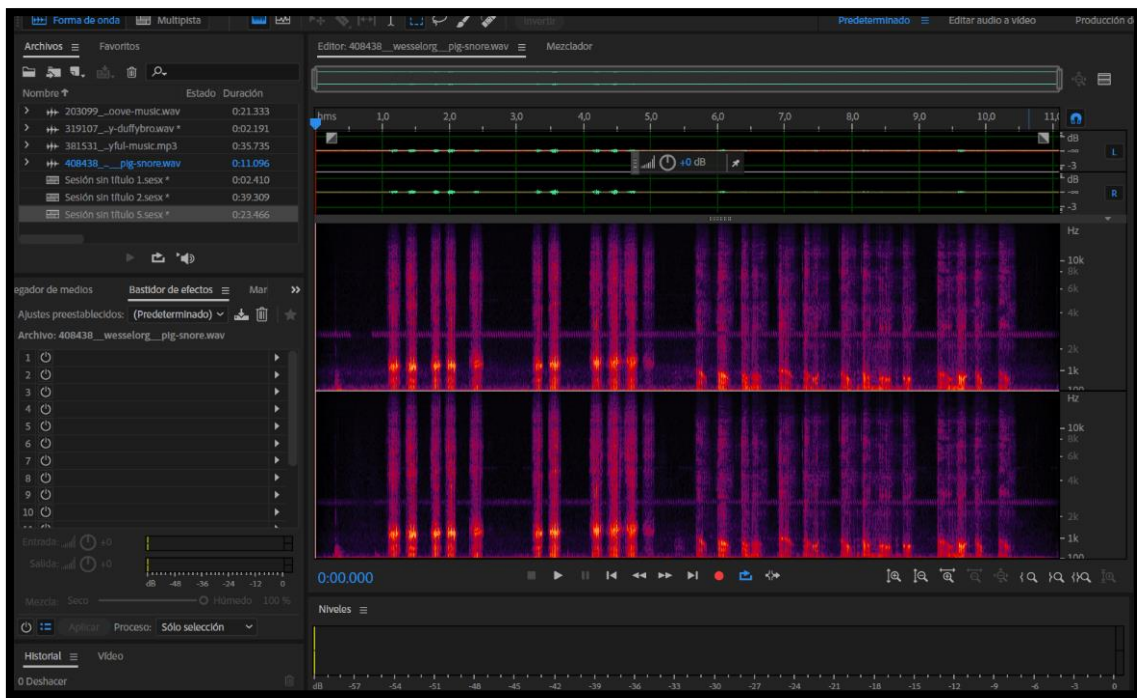


Figure 3.11 Screenshot of the edition of one of the audios of the video game in Adobe Audition.

Chapter 4

Results

In this chapter we will comment on the results obtained after the development of the video game, which has been carried out as explained in the previous chapter, and the testing processes followed during that development to achieve the expected results.

4.1 The game

As a result of this project, an application has been obtained that can be executed on any mobile device with the Android operating system. This application consists of the demo of a 2D video game that mixes the genres of serious games with simulation and management. The game contains different minigames that vary in difficulty and an automatic game saving system.

The game is a light application, which is executed in very short load times, partly as a result of the optimization made in the assets of image and sound.

Regarding the pedagogical aspect of the video game, which is its main purpose, it has been possible to include some types of exercises for the reinforcement of reading skills aimed at people with dyslexia, whose effectiveness was already proven. The inclusion of these exercises in the game has been done as expected, introducing with them game mechanics that turn the activity into an entertainment.

At the aesthetic level, the resulting application has a visual and sound style suitable for the target audience.

On the other hand, an application capable of self-regulating has been achieved, varying its behavior and difficulty depending on the skills that the game detects in the player.

In the following links it is possible to access the apk file of the video game to be downloaded and a demonstration video of the gameplay:

- Apk file:
<https://drive.google.com/drive/folders/1uW1PvnDyUZh1fdtWik7VLNIAyuqse9Ny?usp=sharing>
- Video gameplay:
<https://drive.google.com/open?id=1LF9qxlZxXkuTBYZybFMlmLbiqdLrLG0k>

4.2 Testing

During the development of the video game, and once it was finished, tests have been carried out continuously in order to verify the correct functioning of the characteristics that were being implemented in the game. In this way, it has been possible to correct bugs and get feedback from people who have tested the video game, in order to improve it. Within this section, the verification and validation processes of the product created have been carried out.

4.2.1 Verification

This process has been carried out simultaneously with the development of the video game. As a developer, I have been in charge of performing this task.

In this phase, the objective was to check, parallel to the implementation of the characteristics of the game, if the product that was being created met the characteristics set at the beginning of the project. That is to say, in the verification it was verified that the game was being constructed correctly.

In this part, multiple software tests have been carried out in order to verify that the program worked correctly, consisting mainly of unit tests, where the correct functioning of a part of the added code is checked, and integration tests, where it is verified that the different parts of the code work well together. In addition, actions have been forced in many cases to put the game to the test, with the aim of finding bugs in rare cases, because programming failures sometimes go unnoticed during development.

In short, all the programming errors that have arisen during the creation of the videogame have been searched, found and corrected.

4.2.2 Validation

In this process, different people outside the project have tested the videogame. This group of people who have tried the application is formed, on the one hand, by friends and family and, on the other hand, by people with dyslexia with which the effectiveness of the game's activities has been proven. Here it has been validated, above all, that the final product built is what was expected to achieve and meets the expectations of the target audience.

In this way it has been possible to find new bugs on the one hand, because when a game programmer tries out his game, unconsciously he always tends to do the same tests that do not cause failures, since he knows perfectly how the game is

programmed. However, a person outside the project usually finds errors quite easily by performing actions that in many cases the developer has not checked before. In this project this part has been fundamental to find the most unexpected errors that only friends and family have been able to discover, allowing me to correct them.

Another usefulness of having the help of people outside the development of the project is that, once tested the game, they have been able to give me feedback, giving opinions on the different parts of the game, about what they would change to improve the result or would like to be add to the final game.

For example, several of the people who tried the game suggested reducing the speed with which the apples fell in the "Apple Tree" minigame. Also, regarding the art of the videogame, their opinions were decisive when choosing between different aesthetic models proposed for each object. In addition, they proposed alternative ways to interact with the game's interface, among other things. In most cases the suggested changes were introduced in the video game.

4.2.2.1 Tests with dyslexic persons

As previously mentioned, the tests have also been carried out by people with dyslexia in order to know the degree of effectiveness of the videogame activities. For this, I have had the help of two dyslexic children who will evaluate the application. The results of their tests are expected to be received in the near future. On the other hand, the test has also been conducted with a sixteen year old dyslexic teenager, which results are already available.

The results of the tests consist of, once the game is tested, the dyslexic player will answer a series of questions about the application. Some of the questions asked are the following:

- Do you like the game?
- Is it difficult to understand how the game works?
- Is the difficulty of the mini-games adequate?
- At some point have you found a greater difficulty to perform any activity of the game? What was that difficulty?
- Is the type of videogame activities a good way to help children with dyslexia? I mean, is the game effective?
- For what age range do you consider the game to be more appropriate? (Question for older players)
- Which minigame do you consider most useful as an activity for a person with dyslexia?
- What would you improve in the game?

As mentioned before, a sixteen year old dyslexic boy has tested the videogame. After testing the application, he responded to the above questions.

In his opinion, the video game uses effective activities that help to improve the reading skills of people with dyslexia. He considers that the difficulty is adequate, although he thinks that for older people it is easy enough. Therefore he says that, although the game may be useful in older dyslexic people, the application is especially useful for children, being a good way to learn for them. Finally, he also considered that the “Apple Tree” minigame is the most effective activity as an exercise intended for people with dyslexia, although the minigame of the train was the activity that he liked the most.

Chapter 5

Conclusions

In this project has been raised and carried out the creation of an educational video game aimed at dyslexic children. During the development process I was able to put into practice part of the knowledge acquired during the four years of my degree, having to develop the art, design the mechanics of the game and program.

The application obtained is an interesting product that could be applied in the educational field, because the interactive activities proposed as games capture better the attention of children than the traditional teaching model. In addition, including such special applications for children with learning disorders at school could help reduce the exclusion that sometimes exists for them within schools [11].

Next, I will comment on the degree of achievement of the objectives set at the beginning of the project, the modifications that have been made during the development process and some ideas in order to transform the demo I have made into a complete application.

5.1 Achievement of objectives

At the beginning of this project, as can be seen in section 1.3, the following objectives were established to be fulfilled once the development of the videogame ended:

- Adapting therapeutic exercises for dyslexia to mini-games, prevailing its didactic function.
- Create an AI that adjusts the difficulty of the levels to the progress of the player, focusing on their errors.
- Elaborate an original art, with colorful and childish aesthetics, that forms part of the appeal of the game for a young audience.
- Integrate all the previous elements in a mobile app.

All the objectives established in the beginning have been fulfilled, obtaining the mobile application that was expected. Therefore, based on the degree of achievement of the objectives set, the final result is completely positive. However, the work can be considerably improved by dedicating more time in the future, as will be discussed later.

5.2 Deviations in the project

Throughout the development of the project, factors such as time or the change of criteria regarding how to carry out some parts of the project have produced changes in the final result of the work with respect to the approach that had been made in the beginning.

5.2.1 Deviations in the game

As we said before, changes in the criteria when making the game caused the final result to vary.

The changes mainly occurred when implementing the main scenario, where the player can freely buy and place objects on the stage. However, in the idea proposed in a first version of the Game Design Document, this scenario would have a more limited functionality, since the inspiration for this scene was based on the level screen of the game *Candy Crush Saga* [12] and not in a management game like *Farmville*.

For that reason, instead of existing a grid with cells where to place objects, the scenario would count with all the objects already placed in fixed points from the beginning and it would be the player's function to unlock them with coins to be able to make use of them. On the other hand, the objects of the scenario, once unlocked, would have as their only function to allow access to minigames, therefore, there are no objects that we call decorative.

The criteria that motivated the changes were the excessive simplicity of the proposed scenario and the search for a more stimulating game experience for the children. Therefore, the main stage became a more important part of the game, because it serves to motivate children to perform the activities of the mini-games where they get money that they can use to buy different objects with which to customize the appearance of the farm.

This change served to improve the game by making it more attractive. However, it required more effort and time than had been planned, which meant having to reduce the time required to perform other parts of the game. In order to avoid that this supposed a loss of quality in the game in general, it was decided not to realize one of the minigames that had been planned.

5.2.2 Temporary deviations

For the realization of the project it is estimated that 300 hours of work are necessary. However, it is quite likely that the work carried out for the realization of this project has required more hours than estimated, because the smallest of the tasks of the process of creating the game involves several hours of work.

Regarding the initial task planning, carried out in section 1.4, which established the time to be devoted to each of the tasks, it is estimated that there have been some variations (See Table 4.1).

Tasks	Estimated hours	Hours used
Task 1 - Research on similar apps and documentation	15	10
Task 2 - Technical proposal	10	5
Task 3 – Game design	15	25
Task 4 – Art design	60	65
Task 5 – AI and programming	100	110
Task 6 – Technical resources	5	5
Task 7 - Game assembly	25	35
Task 8 - Playtesting	10	15
Task 9 – Final memory	50	50
Task 10 - Presentation	10	10
Total hours	300	330

Table 4.1. Difference of estimated and used hours in the project.

5.3 Future work

As discussed at the end of Section 5.1, in the future it is possible to make improvements to the game, since the current version is a demo with certain shortcomings.

On the one hand, more minigames would be added later than there are now. As with the minigames that are already created, each new minigame that would be included would deal with a different type of activity to reinforce reading skills.

In addition, in a final version of the game is expected to introduce a variety of objects with which to customize the farm, compared to the limited number of them that exists now. The creation of a greater variety of animals would be highlighted, since the tests carried out by people outside the project revealed that animals are a very successful element of the game.

Another of the improvements proposed would be to introduce a tutorial at the beginning of the game to help the player to better understand the operation of the game. On the other hand, in the main menu it would be expected to introduce a button that allows access to a screen with information about the operation of each minigame and the type of reinforcement exercise of the reading and writing skills on which each one is based.

Finally, it would also have been interesting for this demo, and necessary for a more advanced version of the game, to include indications of how to perform the activities through a voice and exercises that use sound to dictate words. That would be a good improvement since, in the case of people with difficulty reading, the sounds would be of great help to them.

Although all the features discussed above have not been introduced in the demo, is something that is taken into account and would be included in a final version of the game.

Bibliography

[1] Bob Cunningham, Ed.M (n.d.). ¿Por qué la dislexia no desaparece con una enseñanza adecuada?. *Understood* . Recovered from <https://www.understood.org/es-mx/learning-attention-issues/child-learning-disabilities/dyslexia/does-dyslexia-go-away> (last access: Feb 05, 2018)

[2] Solís, Michelle (2013, September). Videojuegos para la dislexia. *Investigación y Ciencia*, 62. Recovered from <https://www.investigacionyciencia.es/revistas/mente-y-cerebro/el-legado-de-freud-583/videojuegos-para-la-dislexia-11384> (last access: Feb 05, 2018)

[3] Dyseggixa (game), <https://itunes.apple.com/us/app/dyseggixa/id534986729?l=ca&ls=1&mt=8> (last access: Feb 05, 2018)

[4] Los videojuegos de acción pueden ayudar a las personas con dislexia. (2017, December 12) *La Vanguardia*. Recovered from <http://www.lavanguardia.com/ocio/20171212/433589865915/los-videojuegos-de-accion-pueden-ayudar-a-las-personas-con-dislexia.html> (last access: Feb 06, 2018)

[5] Dyslexia. (n.d.). On Wikipedia. Recovered on February 24, 2018 from <https://en.wikipedia.org/wiki/Dyslexia> (last access: Feb 06, 2018)

[6] Official website of OpenDyslexic, <https://www.opendyslexic.org/> (last access: May 31, 2018)

[7] Official website of DyetectiveU (game), <https://dyetectiveu.org/> (last access: June 24, 2018)

[8] Official website of Wacom, <https://www.wacom.com/en-us> (last access: June 10, 2018)

[9] Guía de buenas prácticas para assets de arte. Manual de Unity, <https://docs.unity3d.com/es/current/Manual/HOWTO-ArtAssetBestPracticeGuide.html> (last access: June 11, 2018)

[10] Official website of freesound, <https://freesound.org/> (last access: June 12, 2018)

[11] Blanco García, Ana I. (2009, January). Dislexia, escuela y exclusión social: un estudio desde la sociología acerca de la educación especial. *Configurações. Revista de sociología*. Recovered from <https://journals.openedition.org/> (last access: June 16, 2018).

[12] Official website of Candy Crush Saga, <https://king.com/es/game/candycrush> (last access: June 15, 2018)