Implementation of user-centred design methods for developing accessibility features in a video game

Olga Montero Rodriguez

Final Degree Work

Bachelor's Degree in
Video Game Design and Development

Universitat Jaume I

May 28, 2021

Supervised by: José Vicente Martí Avilés

Acknowledgments

First of all, I would like to thank my mom and my sister for supporting me during this process and Coco and Rufo for being by my side while writing this report.

Besides, I would like to thank Esteban Galán, for bringing me the opportunity to discover the audiovisual accessibility world.

And a special thanks to Carlos, for supporting me during the whole degree and the development of this project.

I also would like to thank Sergio Barrachina Mir and José Vte. Martí Avilés for their inspiring LaTeX template for writing the Final Degree Work report, which I have used as a starting point in writing this report.

And finally, to every person who has participated in this project somehow, thank you, this wouldn't be possible without you.

Abstract

This document represents the Final Degree Project report of Olga Montero Rodriguez in Video Game Design and Development. The project consists of developing accessibility features for a platform video game using user-centred design methodology. This design method aims to include users with disabilities in the design process to be able to develop a final prototype that reaches their expectations.

Contents

CHAPTER 1

INTRODUCTION

Contents

This chapter explains the motivations and how the idea was proposed, its different objectives, and its beginning state. [15].

1.1 Work Motivation

During my third year of the degree, I had a subject called "Theory and practice of audiovisual production". As a result of my implication in the final project, the teacher responsible for the subject invited me to a congress about Audiovisual accessibility. We were encouraged to work with professionals from different audiovisual areas to share experiences and work in innovative solutions to improve accessibility in audiovisuals. During the congress, I started thinking about accessibility in video games, and I figured out that I didn't feel accessibility was given the same importance as an audiovisual product. Also, I have always been aware of the importance of accessibility in other fields, mainly because there are some disabled members around my inner circle, and some family members are health care professionals. Because of that, I've heard many different experiences in video games and acquired some specific knowledge about disabilities.

Those are the reasons that bring me to try to develop an accessible video game using the centred-design method and try to contribute to the video games industry, also in

other to prove that there is no need to have complex technologies to develop accessible features in a video game and help disabled players to feel included as any other.

1.2 Objectives

- Implement a user-centred design method to make users part of the development and design process.
- Develop a video game using Unity Engine [47] technologies and implement accessibility features without using any other complex software.
- Create an integrated accessibility menu to configure the game while playing, helping the player enjoy the game and not getting bored while navigating through complicated menus.
- Design accessibility features for a specific impairment or disability and the unique needs of inexperienced players or people with more than one disability and groups them in a multi accessible experience.

1.3 Environment and Initial State

The main idea of this project comes from a personal interest in audiovisual accessibility and personal concern for trying to understand why the current video games industry and big companies are ignoring the special needs of some of their users. Also, to know why sometimes they include inferior accessibility options that do not benefit players, leaving aside disabled users from the video game experiences.

Once the idea was chosen and concreted with the supervisor, it was necessary to develop and create a planning. The proposed methodology for developing accessibility features in the video game was a user-centred design method because of its multiple benefits to accomplish the users' expectations of a specific product. The project is classified into three main phases in order to organise all the tasks.

The video game would be developed using Unity Engine [47]. The visual and sound aspects would be created using a mix of assets from Unity Asset Store [43] and assets developed to accomplish some special needs of the target users.

1.4 Related subjects

- VJ1227 - Game Engines
- VJ1236 - Sound production and realisation techniques
- VJ1237 - Video game localisation

- VJ1224 - Software engineering
- VJ1222 - Conceptual design of video games

CHAPTER 2

Planning and resources evaluation

Contents

This chapter shows the planning that has been followed to organise and develop the project, the resources and tools to accomplish that purpose.

2.1 Planning

The next lines show the task that has been done to develop this project. Each task is part of a group dedicated to one specific phase due to a user-centred design method. Some tasks were developed in a specific order due to dependencies with another task, but others were made during the whole process of another task.

- **Previous documents (20h):** Before starting the video game, a Technical proposal and a Game Design document were made to explain and expose the project to the advisor on how the project would be developed and its different parts.
- **Phase I : Analysis and development of the first prototype (120h):**
  - **Previous games analysis and explore implementation possibilities with unity (5h):** A brief analysis of the most important games in the present, including accessibility features and how it has worked on players. In addition, research of those accessibility features and how Unity technologies [47] allow to develop them.

  - **Analysis of users requirements (10h):** Gather documentation and research about the impairments and disabilities of the target audience.
  - **Assets search and development (10h):** Search for free assets that work with the artistic style restrictions and develop new assets if needed.
  - **Main menu and tutorial design and development (15h):** Design and develop the main menu with basic functionalities and a tutorial for the game as a safe environment to configure the game.
  - **Accessibility features design and development (80):** Implementation of the accessibility menus on the tutorial and allows the user to configure all the features while playing. Implement four main groups of accessibility features: vision, sound, locomotor and cognitive.

- **Phase II: Testing the first prototype of the game with target users (50h):**
  - **Assembly of levels 1 and 2 (40h):** Use the assets and systems developed in the first phase to assemble two different levels.
  - **Play-testing with users (10h):** Develop a questionnaire for the users and collect the information and feedback from the video game.

- **Phase III: Implementation of new solutions (110h):**
  - **Design solutions based on the currents information (10h):** Develop new design solutions based on the video game systems already developed to reach users expectations.
  - **Explore implementation possibilities and implement new design solutions (50h):** Analyse unity technologies and implement the new solutions if possible.
  - **Final project report (50h).**

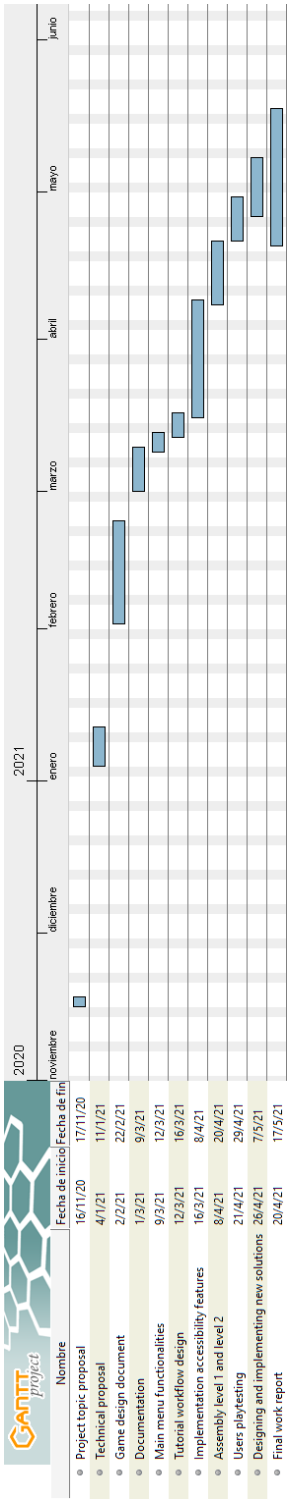


Figure 2.1: Gantt chart of the developed tasks (made with Gantt Project)

2.2 Resources evaluation and tools

- Unity Engine 2020.3.4f, game engine. [47] Cost: free
- Microsoft Visual Studio 2019, programming tool. [44] Cost: free.
- Adobe Illustrator 2019, vectorial design tool. [3] Cost: Student license, free.
- GitHub Desktop, code repository tool. [19] Cost: free.
- Overleaf, online tool for LaTex documents [33]. Cost: free.
- Unity Asset Store: game assets store. [43] Cost: free.
- Trello, to organise tasks. [48] Cost: free.
- Discord, voice application. [12] Cost: free.
- Gantt Project, application to draw Gantt charts. [37] Cost: free.
- Grammarly, online tool to correct texts. [20] Cost: free.
- Google Forms, online tool for questionnaires. [17] Cost: free.
- Google Spreadsheets, online tool for manage spreadsheets. [42] Cost: free.

CHAPTER 3

System Analysis and Design

Contents

This chapter presents the analysis research and the design methodology.

3.1 Objective Analysis

Many games include new accessibility features passing years but not every one of them is working as expected. The aim of this analysis is:

- Find what type of disabilities users have and how they experience video games.
- Recognise problems in current features in order to improve them.
- Study from the existing research on this field and try to learn how these problems were approached to design accessibility solutions.

3.2 User-centred design methodology

As mentioned before, for developing the accessibility features of the project, it would use the user-centred design methodology. First of all, let us clarify how this methodology works and how it will be adapted to this project.

User-centred design method [35] is an iterative process used by the designers to develop solutions focused on the users and their special needs. This means that the target users of the product that will be developed are active participants in the design process, often seen as equal partners in the development systems. This method is becoming more popular nowadays in fields such as human-computer interaction in computing studies.

Generally, this method is formed by four phases. Depending on the type of project and the time restrictions, the number of iterations by those phases is variable.

- **Phase 1:** Understand the context. This means designers have to analyse and understand the context and how users may use a system or a product.
- **Phase 2:** Specify user requirements. Designers have to identify the user's requirements. Depending on the method, this phase is made with interviews, questionnaires or simply researching the target users requirements in other similar products or systems.
- **Phase 3:** Design and develop solutions. Once the requirements have been listed, it is time to develop solutions to fit those requirements and expectations. It seems to be a very linear design process once the requirements are obtained, but it is a very creative phase and maybe the most difficult.
- **Phase 4:** Evaluate the design solutions implemented against the requirements. This phase can also be developed using interviews and questionnaires to evaluate how close the current design is to users' expectations or requirements. With those results and feedback, it is possible to design new solutions or specify new user requirements if needed.

In order to adapt the user-centred design method to reach the specifications of this project, there are three specific phases designed that fit the method differently. This adaptation is required mainly due two restrictions: the health crisis, which impedes working closer with the users who will participate in our tests and the time restrictions of this academic project.

The adapted phases for this thesis are:

- **Phase 1**: Includes part of the first, second and third phases of the general approach of user-centred design method. This phase is for analysing users requirements and also developing a prototype using the information gathered.

- **Phase 2**: Similar to the fourth phase of the general approach of user-centred design, but in this case, it is only for testing and analysing the feedback from users.
- **Phase 3**: It is also similar to the fourth phase mentioned above, but it implements explicitly new solutions if needed after getting feedback from users.

3.3 Research about users requirements

Nowadays, video games are becoming more popular as entertainment, with 2.6 billion players around the world. With an increasing number of people with disabilities playing video games, designers and game developers had to start using a user-centred approach to understand their specific needs in terms of playing video games.

3.3.1 Classification of impairments and disabilities referred

It is essential to know what types of circumstances affect users with impairments or disabilities and what grade before investigating how they interact with video games. The World Health Organization [31] has published a manual called *International Classification of Impairments, Disabilities and Handicaps* [32] which shows a classification used in this thesis to understand and analyse the user requirements for the prototype. Despite all the different kinds of disabilities, impairments and handicaps in this classification, for this thesis, we are going to analyse and group the most common that are affecting stimulus and reactions in terms of interactive technologies such as video games primarily. It is essential also to be aware that disabilities, impairments and handicaps, are many times presented as synonyms, but they are not. Impairment is when there is any loss or abnormality of a psychological, physiological or anatomical structure or function, disability is when there is any restriction or lack of ability to perform an activity considered normal for a human being, and handicap is a disadvantage for a given individual that limits the fulfilment of a role. In this thesis, this kind of terminology is used as synonymous because in terms of video games, a single type of accessibility feature can help an impairment, a disability, and a handicap.

Based on the technical specifications of video games, the most common groups of impairments or disabilities affecting players are:

- Visual impairments: Partial vision loss, legal blindness, complete blindness and colour blindness.
- Hearing impairments: Complete or partial loss of the ability to hear from one or both ears.
- Motor impairments: Loss or limitation of locomotive functionalities.
- Cognitive impairments: Mental and psychological disorders.

There is a proposed game interaction model in *Game Accessibility: A survey* [22] that helps identify what kind of barrier faces a player with disabilities or impairments. This model can be represented as a finite state machine of three states. The first one is when the players receive visual, haptic or auditory stimuli from the game. Based on these stimuli, the next state is when the player determines a response from an available set of actions depending on the game. The last state is when the player provides an input, and the player will receive a new stimulus to perform the next step of this internal state of the finite state machine. When a player cannot receive any stimuli, it is impossible to determine the response to provide, and the game experience is being disturbed.

In these classified four groups, there are also some details to be considered. The severity of the impairments and disabilities is very different in each case, and it is necessary to differentiate as critical or non-critical. When an impairment is critical, the player cannot play the game, and when it is non-critical, the game is playable, but the experience is reduced. Due to technological restrictions in this project, it will only be possible to develop solutions for the non-critical ones because the critical ones may need special devices such as *Xbox Adaptive Controller* [28].

3.3.2 Existing research and manuals about accessibility

Although accessibility in technology has been there for many years, there seems to be a gap between that accessibility approach and accessibility in video games. Despite the research made during the last ten years, a paper by Michael Heron [23] analyses the idea that many of the current problems about accessibility are caused by a lack of awareness. In the academic world, it is still difficult to find recent research about this field and many times, new findings in the accessibility field do not attain to the video game designers, and it is challenging to integrate them into the development.

There are also two erroneous ideas exposed in this paper. One is that not all disabled users enjoy playing video games "for disabled people". They want to enjoy mainstream video games and not feel apart from the big titles launched every year. The best idea to handle this is to design accessible video games not only for disabled users but for every player around. The second erroneous idea is that disabled users are a tiny segment of players, and it is not worth it to design around their particular needs. This is also a big mistake overlooked by the game developers because accessibility is not only for disabled users [39]. It is also a chance to improve users' experience with normalised impairments, such as the ones who wear glasses because of a lack of vision. In addition, accessibility is approached as a group of different pieces, where each one is not even related to the others. This is also an overlooked mistake because many impairments or disabilities can be related to common problems that can be faced within the exact design solutions.

To help video game developers to become closer to accessibility, many organisations had developed manuals [21] [10] to guide designers in this field. Two of the most relevant manuals are *Includification* [2], developed by *The Able Gamers* [1] foundation and *Game*

accessibility guidelines developed by a collaborative group of specialists and academics. These manuals are becoming popular passing years, being used as a general guideline to improve existing and new video games. Both manuals would be used to develop this project to ensure valuable features and test the guidelines with the target users.

3.4 Analysis of accessibility features in video games

In addition to the practical guides mentioned above and the analysis of specific disabilities and impairments from a theoretical perspective [6], it is essential to analyse in a practical way how launched video games had implemented accessibility features in the past and how are them implementing new ones in the present.

3.4.1 Hearing impairment options

As in the cinema and audiovisual industry, subtitles are the most common option in video games for users with hearing impairments. However, the fact that developers include this option does not mean that it is performed correctly. Subtitles style has to be clear, with an easy understanding of fonts and bigger enough. Also, it is essential to subtitle with different colours, every dialogue which includes more than one speaker to identify them. In video games, it is crucial to attend to every movement occurring, and subtitles have to be described in a complementary way to allow the player to understand what is happening.

Life is Strange 2 [24] includes options such as easy-to-read subtitles, variable sizes, and an optional speaker label for additional clarity. Also, *Assassin's Creed: Valhalla* [38] allows the user to activate a new label that informs who is speaking and change the style of the dialogue text background. These are some examples (See Figure 3.1) of user-friendly features that have been working for the last years, even though not being used as much as they should be, due to their easy implementation.

(a) Life is strange 2

(b) Assassins Creed: Valhalla

Figure 3.1: Different subtitle options and styles

Closed captions are often included in video games to describe, such as in audiovisual media, sound effects, or essential audio information for the user. It is important to empathise that this important audio may describe important information about game mechanics or the location from which the damage is coming in video games. In addition, this is why it is not enough to only write in the screen closed captions as subtitles.

To describe where the audio is, *Minecraft* [45] uses closed captions and an arrow indicating if the audio source is coming from the left or right side of the player (See Figure 3.2). Also, this is represented by graphics in *Final Fantasy XIV* [14] and *Fortnite* [16], where the audio source is represented on the screen with its corresponding colour or symbol to differentiate between audio sources. These are exemplary implementations (see Figure 3.3) that help users with impairments to enjoy the game still and play the whole experience because they can identify spatial information such as from what side is the sound coming and if it is far or close.

Figure 3.2: Minecraft, closed captions on the right side.

(a) Final Fantasy XIV

(b) Fortnite

Figure 3.3: Closed captions and visual feedback about audio source location

3.4.2 Visual impairment options

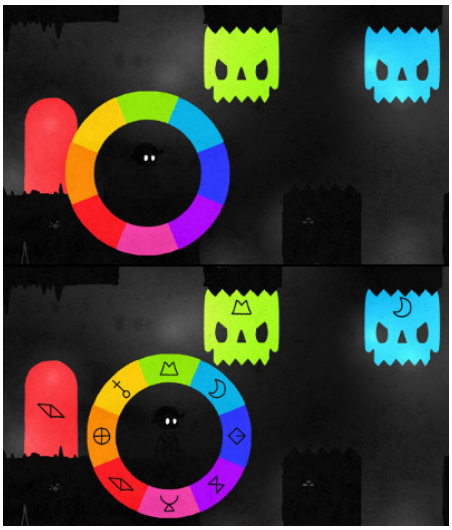
This kind of impairment is the most common experienced by video game players. The first to be implemented and the one which became more populous were the colour blind options. Many times these options are represented as a filter that changes colours by using a shader. These new colours are sometimes unclear and confusing for users with this type of impairment because not all users have the same grade of colour blindness. In the end, these kind of filters are not that useful, and instead, changing the colour palette of the most relevant elements in the game as enemy markers, waypoints, and relevant objects allows the player to differentiate those elements from the rest and still get that information.

Some games developed filters as colour blind options, but the community was not satisfied, like the ones in *Overwatch* [18] (See Figure 3.4) and *Fortnite* [16]. For this purpose, the best functionality was the most simple, as the ones developed in *Hue* [11] or *Battlefield 1* [8]. In the first one, developers decided to assign a symbol for each colour

(a) Colorblind filter disabled

(b) Colorblind filter enabled

Figure 3.4: Colorblind filter in Overwhatch



(a) Hue

(b) Battlefield 1

Figure 3.5: Colorblind options

to differentiate them, and in the second one, the game offers four different palettes for each kind of colour blindness to differentiate essential objects (See Figure 3.5).

Low vision is one of the most common impairments, and video games are starting to include options for resizing texts and changing the font style. Also, the option to resize HUD elements is widespread. If the lack of vision is severe, some games also implement particular sounds to give the player feedback while interacting with the game to understand where the important objects are. In *Borderlands: The Pre sequel* [27], there are many options to resize and relocate all the interface elements to fit the player's expectations. Also, some sports video games, such as *Madden 17* [41], have this option, which helps the player easily find some relevant symbols while playing (See Figure 3.6).

(a) Madden 17

(b) Borderlands: The Pre sequel

Figure 3.6: Resize options for the interface

3.4.3 Locomotor impairment options

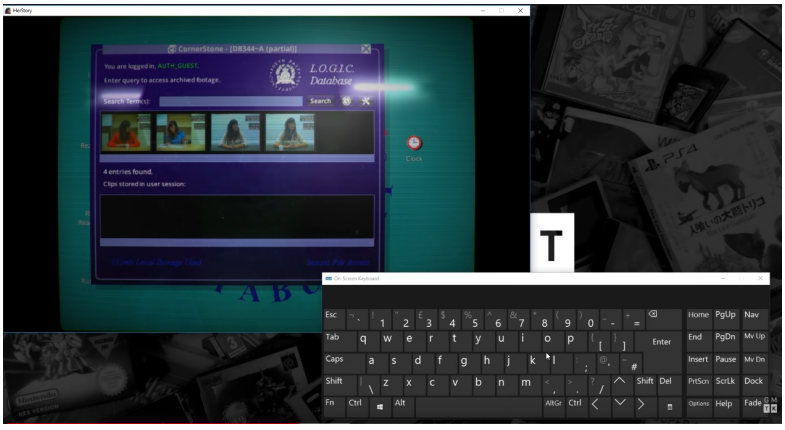
Passing years features to help players with locomotor impairments seem to be the most challenging options to integrate, but they are not. Usually, it is typically first to think that the only option for some users for playing video games is to use adaptive controllers as the primary input devices. However, as in other kinds of impairments, each user has a different grade of severity, and some of those impairments can be joined to develop a primary feature. Some of them are rebinding controls for the users who cannot use the default ones due to locomotor problems, choosing the input devices, game assistants for movement or cameras and difficulty selectors.

The most common and the most demanded by users with and without impairments is the rebinding system for the game's controls. *Tom's Clancys The Division 2* [29], among others, implement a very extensive rebind panel where the player can choose the controls at ease. Many players would like to use their own specific devices to play, and this is also a helpful option, developed in *Her Story* [7], which allows using the virtual keyboard if needed. Also, some games integrate a list of devices among which the player can switch between. This kind of feature was very disputed related to some Nintendo [25] games because it was needed to use their haptic or movable devices for playing the game. This was not configurable, and many players had to stop playing some Nintendo games because of that, such as *Twilight Princess HD* [4], in which the player had to use a haptic screen to interact with the game without offering any other configurable device to this purpose (See Figure 3.7).

In addition, for helping players with locomotor impairments, it is needed to adapt the

game's speed or the movement of its elements. This can be done with an assistant that does some actions for the player or by difficulty selectors to allow the player to choose the one that makes him feel comfortable. Camera assistant and auto-target are some of the features implemented in *Uncharted 4* [13] to help the players. Also, *Mario Kart 8* [5] implements a driver assistant to help players stay inside the circuit. In *Celeste* [26] is implemented one of the most valuable features as a variant of the expected difficulty selector, the adjustable game speed (See Figure 3.8).

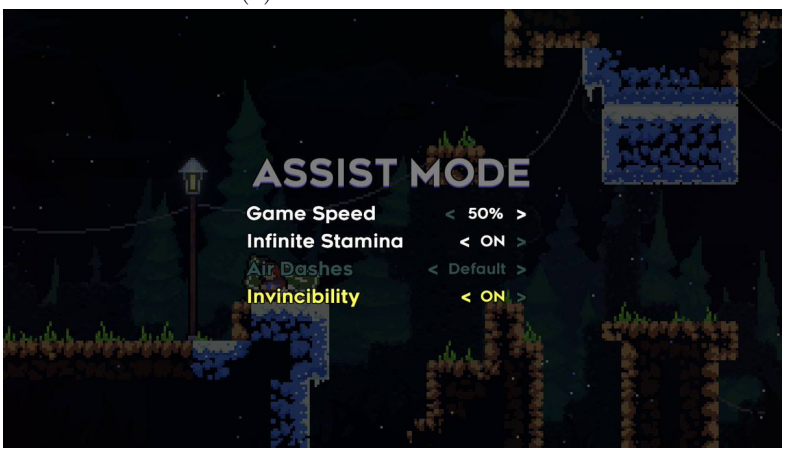
(a) Zelda ingame controls



(b) Her Story configurable device controls

Figure 3.7: Device controls options

(a) Mario Kart 8 assistant



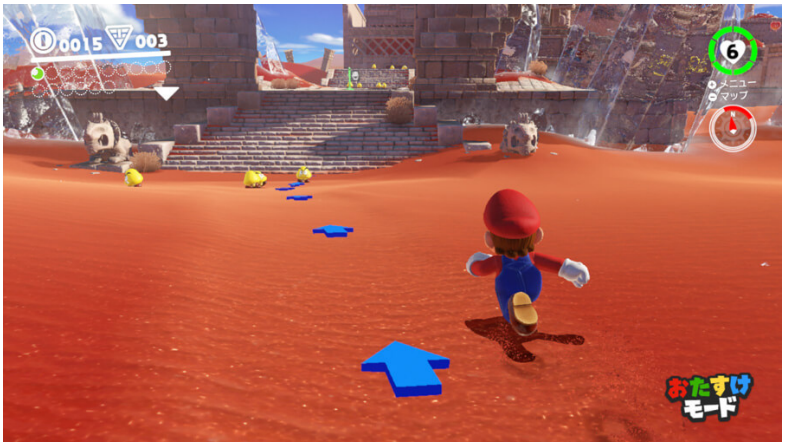
(b) Celeste assisst mode

Figure 3.8: Game assistant options

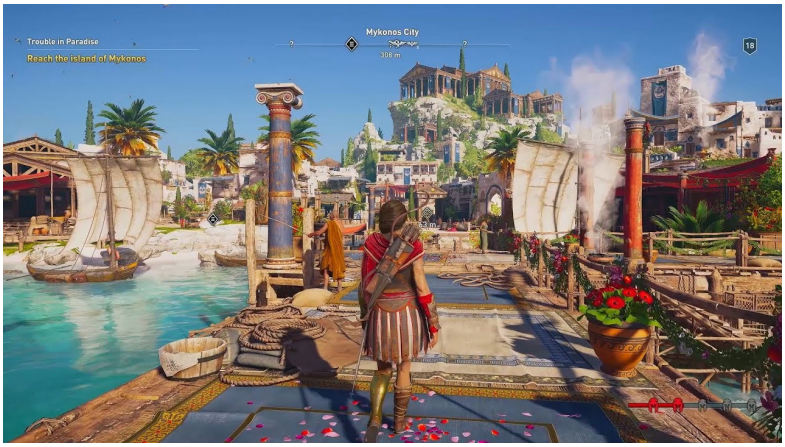
3.4.4 Cognitive impairment options

Some of the standard features to help players with cognitive impairments are included in the sections above because, as mentioned before in this report, it is possible to develop a single feature that could help players with multiple disabilities. Many cognitive disorders cause the player to get confused if the game shows much information simultaneously quickly. It is a valuable option to implement tutorials or testable levels in which the player can play and adapt to the game in a safe environment.

Some games as *Doom* [40] or *Mario Odyssey* [34] implement an assistant to help the player be guided through the game. *Assassin's Creed: Odyssey* [30] also implements a quest assistant to guide the player and help him keep focused on the current game objective (See Figure 3.9).



(a) Mario Odyssey



(b) Assassins Creed: Odyssey

Figure 3.9: Indicator assistant

3.5 Designing the analysis

To reach the analysis objectives, it was proposed to analyse the information above about the theoretical perspective of disabilities and some relevant examples currently implemented in video games. Also, the information extracted from research in this field was beneficial to suggest some design solutions to implement in the prototype.

For the second phase of the project, it was necessary to find users that would like to participate in the questionnaires about the prototype and give their feedback. The analysis raised in this chapter was potentially for the first phase, but finding users may last some time, so it required to be started almost at the beginning of the project. The primary step to start searching users was to find some streamer with disabilities on Twitch [49] and contact its community of viewers. This kind of video game stream-

ing platform offers the possibility to search by categories, and recently, accessibility was added as a tag reference. After a week of navigating through the platform, two streamers and their communities agree to chat about this project on their Discord [12] server. Due to the good reception and the interest of the server participants, a questionnaire was designed to be added to the first design phase. The questionnaire was sent by the text chat of both Discord servers to be answered by any person who wants to participate. This was not planned initially, and that is why it is not mentioned in the second chapter about planning.

The questionnaire information is handy for knowing how the users feel and letting them express themselves as much as possible. The questions to answer were:

1. Do you have any disability or impairment? You can explain here if you want, what type it is.
2. Do you usually play video games? Explain here, if you want, how much time do you invest in video games.
3. What kind of video games do you enjoy the most? Can you explain why? You can list different video games genres if you want.
4. Did you ever stop playing video games because you stopped enjoying the experience? Explain here, if you want, why.
5. When playing video games, is it hard for you to find the accessibility options menu? Explain why if you want.
6. Did you miss any kind of accessibility option while playing some video game? What kind? List them if you have more than one example.
7. Write below three video games that you enjoy because of their multiple accessibility features and three that you had to stop playing because it was very hard for you to play.

These questions allow many different answers and unlimited workspace for them, so the users can explain in their own words how they feel and offer suggestions about games and accessibility features to take into consideration before starting developing the prototype. Also, in the companion of the theoretical information from above, it is possible to analyse which features are the most valuable for the prototype and if is it possible or not to implement them with Unity Engine.

3.6 Analysis results

From the theoretical information about impairments and disabilities, the obtained information about the current studies from a user-centred perspective was very useful to

confirm that this design methodology that has already been applied in many human-computer interaction studies is also valuable for video games. Furthermore, it is proved that it is better to afront the design of accessibility features in a group perspective and leave behind the thought that each option or feature can only help one disability, as they are commonly related. After the analysis through the current accessibility features in the examples above and the users' feedback in the questionnaire, these are the conclusions:

- The most played video games genre by the interviewed users were visual novels, platform games, card games and first person shooters.
- Almost all impairments of the users were non-critical; this means that it is possible to implement features similar to those in the examples above.
- Due to technological restrictions, it is impossible to design the prototype around different devices to play, but as a computer game, it would be nice to allow playing with a keyboard and gamepads.
- It is advantageous to configure all the accessibility options before starting playing or in a tutorial, because players will feel comfortable and in a safe environment to train.
- Some options presented in the examples can be joined and presented as a single option. It is not always positive to have many options because the users can be confused.
- Players prefer to play a less complex game and be able to enjoy the whole experience, but this does not mean that they do not want to play the most significant games been launched every year.

Design must be based on these results, and in order to accomplish this, the game genre will be a platform game. It will also be developed for computers to allow playing with gamepads or keyboards. The game will present a testing level or tutorial to guide players through the different accessibility menus. There will be three menus: visual options, sound options and cognitive/difficulty options. Menus have to be presented with enough possibilities to allow the player to be comfortable but not confused. The testing level will include clues about how to play and will let the player be used to the controls and the game mechanics. Before the testing level, the game will have two more levels. Both levels will be designed around the basic game mechanics shown in the tutorial, allowing the player to face more complex challenges.

3.7 Analysis of requirements

3.7.1 Functional requirements:

- R1: The player can start the game.

- R2: The player can move through the game environment.
- R3: The player can navigate through the menus.
- R4: The player can pause and resume the game.
- R5: The player can play the current level again.
- R6: The player can collect items.
- R7: The player can jump.
- R8: The player can attack.
- R9: The player can exit the game.
- R10: The player can receive feedback from the game.
- R11: The player can change the visual options.
- R12: The player can change the sound options.
- R13: The player can rebind the controls.
- R14: The player can select the difficulty.
- R16: The player can be killed.
- R17: The player can be healed.

3.7.2 Non-functional requirements:

- RN1: The game style will be low poly.
- RN2: The enemies and object will have animations.
- RN3: The game will have moving platforms.
- RN4: The game is placed in an unrealistic city.
- RN5: The game mechanics will be simple to learn.

3.8 System Architecture

These sections describe the specifications of the device where the game was developed and the required specifications for playing the game:
Development device:

- Processor: Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz

- RAM: 16 GB
- Video Card:NVIDIA GeForce GTX 1060 6GB
- Operating System: Windows 10, ver. 10.0.19041

Playable device:

- Processor: X64 architecture with SSE2 instruction set support
- Graphics API: DX10, DX11, and DX12-capable GPUs
- Operating System: Windows 7 (SP1+) and Windows 10, 64-bit versions only.

CHAPTER 4

Work Development and Results

Contents

This chapter explains all the development process details from the start until the end of the project development. It will be explained in chronological order due to the user-centred methodology applied. It also includes a brief analysis of the final results and an assessment of planning changes.

4.1 Work Development

As mentioned before, the whole work development of this project is planned in three different phases. Some tasks were developed during more than one phase due to specific time dependencies, and some tasks were reconsidered during this process.

4.2 First phase: Analysis and development of the first prototype of the game

This phase consists of designing and developing an analysis of users requirements. Then, with this information, it was possible to explore the implementation possibilities of the chosen features. Finally, a prototype was developed that includes a main menu and two different levels ready to be tested by users in the next phase of the development.

4.2.1 Setting up the project

For starting the development of the prototype, the first step was to choose the Unity Engine version according to the tools used. The initial version was 2019.4.4f1 but was updated to 2020.3.4f1 later for using some packages that were not available in the previous versions.

The main packages used to manage some systems and game mechanics were: Cinemachine [9] for the different game cameras, Input System [46] for the controls and the different device options, and finally, Text Mesh Pro [36] to develop different elements of the user interface.

Finally, for managing the project updates, it was necessary to use a repository on Github [19] to keep track of the source code updates.

4.2.2 Game design, level structure and workflow

Using the analysis results, the game genre proposed was a platform game, and it will be formed by the main menu, a testing level or tutorial and two different levels more. The workflow proposed was a one-way flow (See Figure 4.1), in which the player has to play the testing level or tutorial at the beginning. Once this level is overcome, the first level will be available and finally the second one. The reason that the first level is always the tutorial is that its purpose is to configure the game before starting the challenging part of the game, and if it is the first time playing or not, the user can take its time and configure all the options to get used to the game mechanics.

The level design of the tutorial had to be developed around the user's expected behaviour to show all the configurations and game mechanics carefully to the player. Based on the analysis results, the tutorial has to be represented as a safe environment where the player can test everything without getting punished excessively. This will be a good starting point to guide players with locomotor or cognitive disabilities and help them not feel frustrated initially.

As a brief overview of the game, the player will have to go through the city, collect crystals, and reach the final point of the levels. There will be different kinds of enemies

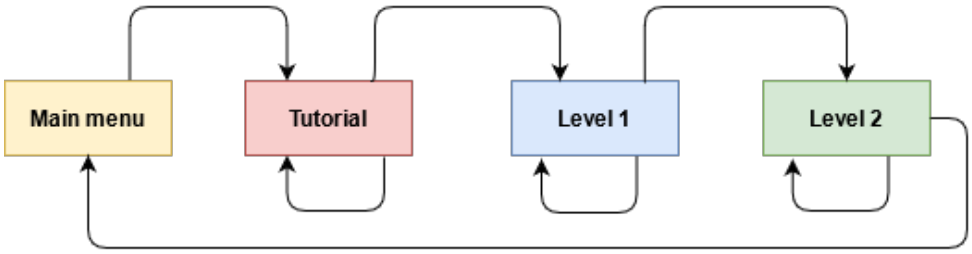


Figure 4.1: General workflow of the game

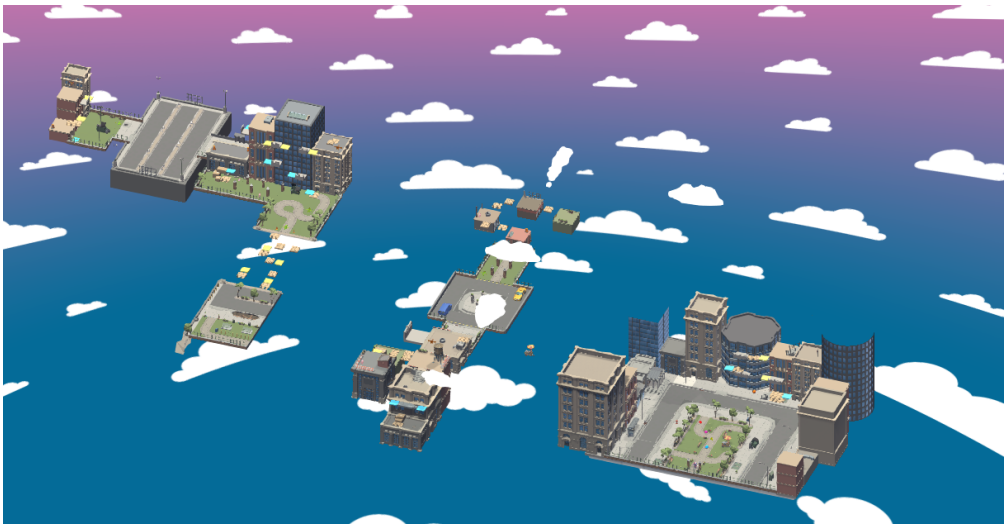
that can be destroyed by collecting some attack items and interacting with them. If the player gets hurt, the medical kits will be healing its health bar. Once a level is finished, a score screen is shown and loads the next level. The player will win, reaching the final point of level 2.

The workflow expected for the tutorial starts with the player moving from the starting position to the first interactable object, which will show the visual options available. The reason to place this kind of configurable options first is that due to the multiple dependencies between the different options and how they can affect each other, the visual aspect is crucial. It was expected that the player keeps going to the following configuration step, related to locomotor and cognitive options. Once the player reaches this part, two of the three configuration steps will be already shown to the player to accomplish the dependencies established. The last point of the configuration will be the sound part of the game because this part is the least determining. With the accessibility options configured, the next step is to show the player what the game mechanics are and how they work. Using the same method as the one with the configuration steps, one interactable object will present the attack mechanic and how to use it against the enemies. During the tutorial, the enemies were designed to be shown after all the configuration steps, so the player can fight them with everything set up as they consider. These kinds of enemies had to be the easiest to beat and ensure that the player did not die many times against them. Finally, to end the tutorial, there will be a short platform challenge where the player can test the controls and start to familiarise themselves with the game's primary challenge.

After thinking about all the designing solutions, it was also necessary to know how the information about the current accessibility configuration would be sent to the different levels. The game will be configured initially, but it is crucial to think about how this configuration data will be saved to be applied to the following levels. Sending information between scenes can be confusing and unsafe, so the solution was to create the first scene for the main menu (See Figure 4.2a) and then develop a second one to contain the other three stages: tutorial, first level, and second level (See Figure 4.2b). This will help to

keep the data about the accessibility configuration unaffected while navigating through the levels. Each level stage has a camera, and it will be enabled or disabled as needed. When the player overcomes a level, it feels like loading a new scene, but it is a fade out of the camera while switching where the next level is.

(a) Overview of the main menu



(b) Overview of the three levels in the same scene

Figure 4.2: Scenes of the game

4.2.3 Assets search and development

One of the most user-friendly artistic styles for video games is the low poly style based on the analysis results. This is because the flat colours and the simple shapes help players with visual and cognitive disabilities quickly identify all the essential information coming from the game objects. As mentioned in the research about users' requirements, many games provide visual and cognitive stimuli like fast movements of the environment elements, sophisticated particle systems or elaborated shaders that some players can not receive. Low Poly environments provide a simplified three-dimensional perspective due to their flats colours and simple models that become easy to understand.

All the three-dimensional assets were provided by the Unity Asset Store and extracted from six different packages. Although the assets were not from the same package, it was easy to combine them because the style was very similar. Everything was personally developed for the user interface assets to accomplish the design and visual restrictions about accessibility. Head-up-display elements and menus were developed using Adobe Illustrator, and all the texts were made using Text Mesh Pro.

4.2.4 Implementation of accessibility features

After designing the tutorial workflow, it was necessary to start developing the accessibility features to be configured. This development was divided into four stages ordered by the priority of its implementation. At the same time, the build of the environment for the tutorial and the main menu was started.

The first stage was the implementation of the visual options. At the same time, the design of the user interface was started because those tasks were significantly related. The user interface has to show the player information about the game's current status and provide an easy composition to read the information fluently. In this case, this game information was composed by the collectable objects counter, the attack objects counter, and finally, by the player's health bar.

Using the results from the analysis and after checking the implementation possibilities using Unity, the visual options menu was developed using two different panels. The first panel was related to the colourblind options and was composed of one configurable option. The second panel was about low vision options, with three different configurable options. The final list of visual options developed in both panels was:

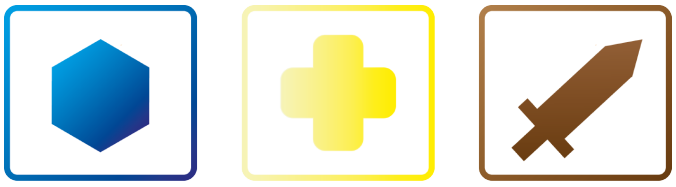
- Two adaptable colourblind palettes for the head-up-display icons and health bar.
- Switching on/off an outline to remark in different colours the important game objects.
- Resize all the texts on menus.
- Resize the icons and scores in the head-up-display.

The head-up-display icons were designed as recognisable figure inside a white square to help differentiate the primary shape on the background. The colours for each symbol were related to their representation inside the game, such as the blue crystals, but it was needed to develop two more different palettes for the icons to implement a dropdown to select the colour palette (See Figure 4.3). For these different options, it was necessary to research the different types of colourblindness and how they affect. There are three main types of colourblindness which are the most common affecting users:

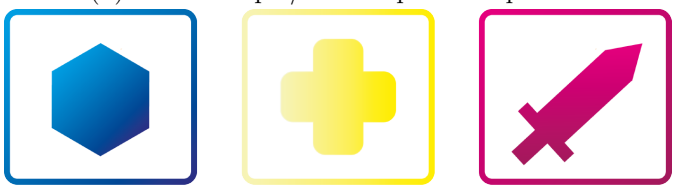
- Protanopia and deuteranopia: cannot tell the difference between red and green.
- Tritanopia: cannot tell the difference between blue and green, purple and red, and yellow and pink. It also makes colours less bright.

As these were the most critical colourblind impairments, the result was to develop two different colour palettes: one for protanopia and deuteranopia and the other for tritanopia. These colour palettes will be applied only to the most relevant game objects to help players get important information from the current game situation. In this case, it was applied only to the head-up-display elements because, as shown in the analysis, complex shaders that will modify all the scene's colours are not that useful in the end.

(a) Basic color palette



(b) Deuteranopia/Protanopia color palette



(c) Tritanopia color palette

Figure 4.3: Different colorblind adaptable palettes for icons

The feature that remarks with an outline the important game object was implemented using a toggle. These user interface elements allow the player to switch the

outline on/off as needed. To remark every important game object, it was necessary to add an outline component to each prefab on the game that was relevant or interactable. This component is developed to draw a line and offers the possibility to choose the colour and the thickness. Every colour was chosen, taking into account the colour of the game prefab to help the player differentiate each element. The system that enables and disables the outline is made using a list of all the outline elements on the scene, which accessing that list the toggle can make the outlines visible (See Figure 4.4).

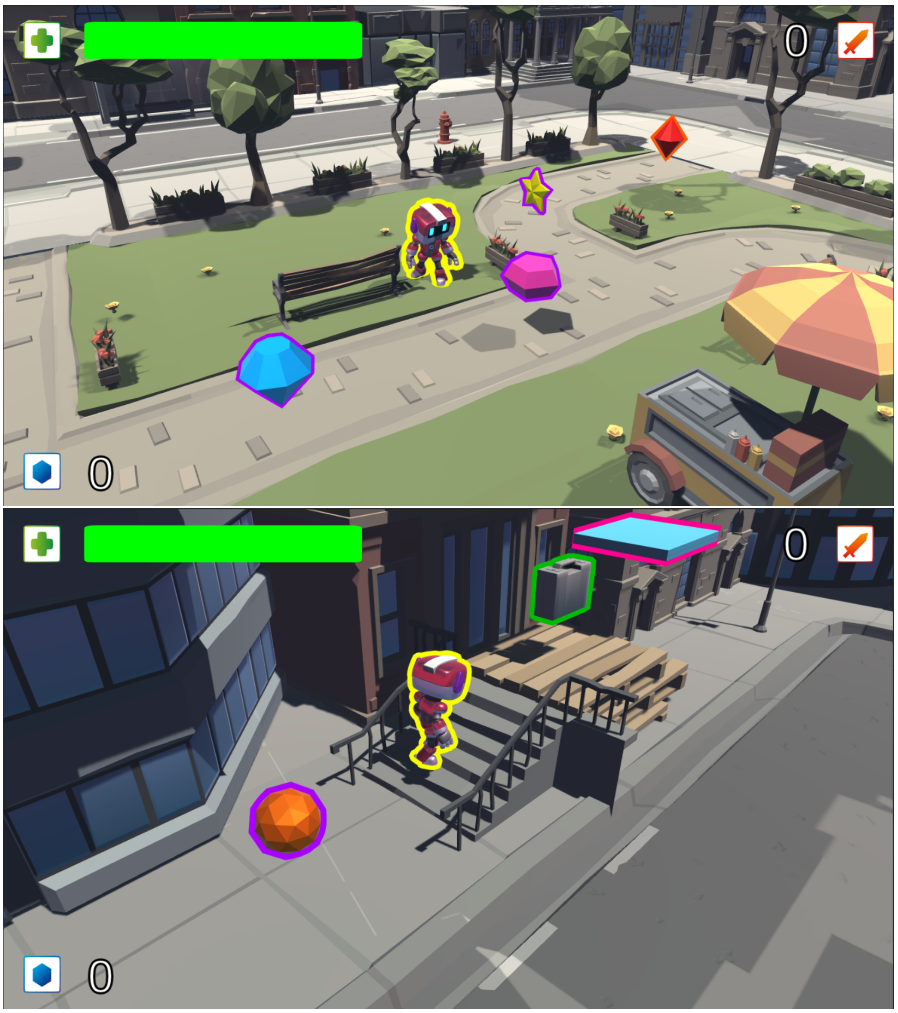
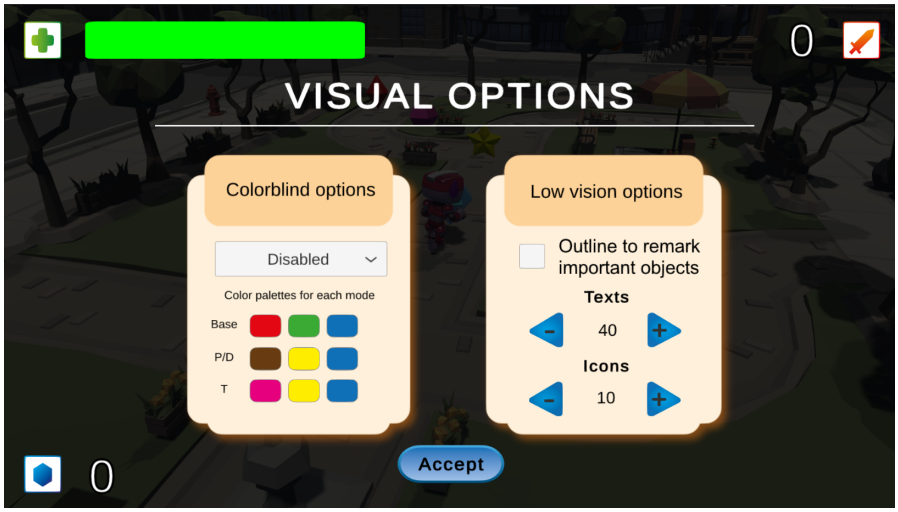


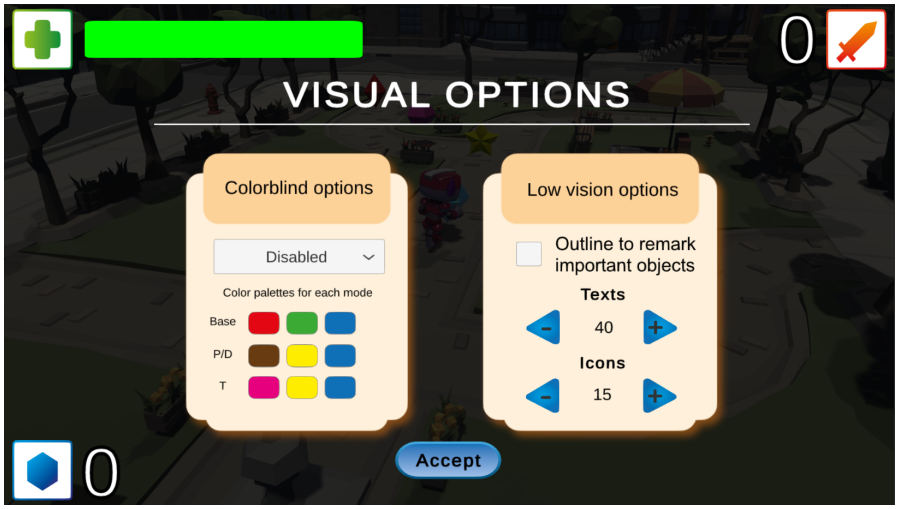
Figure 4.4: Different game objects with outlines.

For developing both resizing options, a system was implemented very similarly to the one used for the outline. Resizing the text and the icons were designed as two different options because it is more beneficial for the player to choose separately to accomplish their needs. Also, both systems were very similar but had to be developed by modifying

some details. The reason was that changing the text size (See Figure 4.6) depends on the value applied to a font size parameter in the text component, but in the case of the icon (See Figure 4.5), the value changed was weight and height on its transform component. Inside both option systems, there is a list of the component to resize. By clicking on the arrows, the value increases or decreases, and then the component values are updated on runtime.



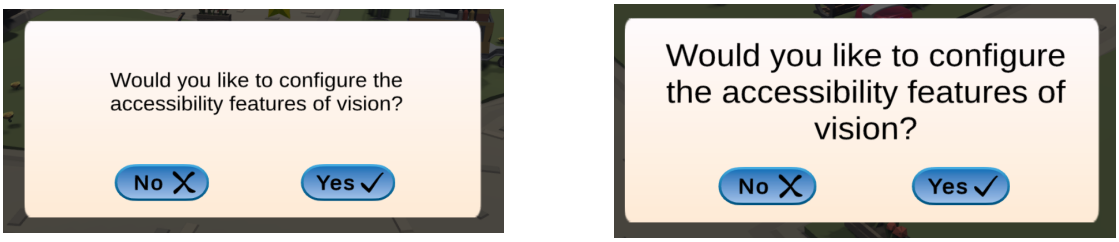
(a) Default icons size



(b) Biggest icons size

Figure 4.5: Overview head-up-display icons with different sizes

The second stage was implementing the locomotor accessibility features. For this purpose, the main configurable option was a rebinding panel that allows the player to



(a) Default text size

(b) Biggest text size

Figure 4.6: Overview of different text sizes

change all the game controls no matter if he is using a keyboard or a gamepad. This kind of accessibility option is widespread, and in passing years it is more demand, not only for the disabled players but for everyone. This functionality was developed using the Input System package provided by Unity. Searching documentation about this system, I decided to use version 1.1.0 rather than version 1.0.2, which Unity verifies. This was because version 1.1.0 offers some new features related to saving data of the rebounded controls in a JSON document. This kind of data file is widely used in video games to store information about configurations to ease the game experience. It was challenging to use a more recent package with less documentation, but the results were helpful in the end. The JSON file keeps saving the new bindings after changing the default controls, and this offers the possibility to play the game with the same configuration, even if the game is reloaded.

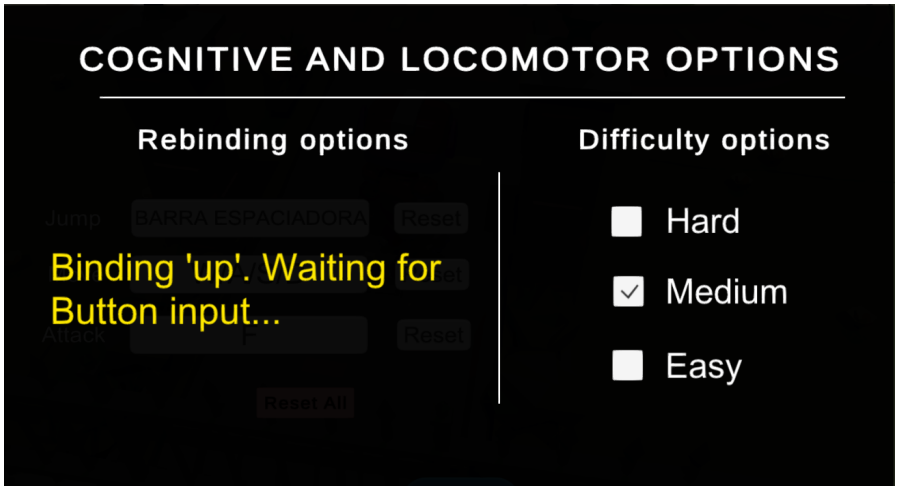
The rebinding panel has four buttons with the default binding text related to the current action map. An action map is a set of actions where each action represents a game control. The player has an action map where there are defined four actions by default:

- **Move**: A/S/D/W on the keyboard and left joystick of the gamepad.
- **Jump**: Spacebar on the keyboard and south button on the gamepad.
- **Attack**: Key F on the keyboard and west button on the gamepad.
- **Display the pause menu**: Escape key on the keyboard and select button on the gamepad.

The input menu shows on each button one of these controls with its binding by default. To rebind them, the player can press one button control (See Figure 4.7), and the rebinding action will start. This process starts waiting to read a new input from the keyboard or a gamepad and shows a grey panel to inform the player that the system is waiting. When a new control is pressed, the button changes the text to the new binding, the action map is updated and stops the rebinding action. This system has to check while reading a new input after completing the rebind if the current control is excluded.

It was necessary to exclude some controls such as the mouse pointer, both mouse buttons and the escape key to maintain the consistency of the general controls for the game. A problem may be encountered if the player could rebind any control as the escape key, whose purpose is to show the pause menu. Every time one control is rebounded, the JSON document is overwritten and saves the new controls. Next to each control button in the menu, there is a reset button to replace the selected control as default.

(a) Rebinding and difficulty options panel



(b) Waiting for new input using the rebinding system

Figure 4.7: Overview of different cognitive and locomotor accessibility options.

After the analysis results, it was proposed to develop some kind of camera options that could allow the player to configure the camera movement as a free camera or target camera. Other possible features would be changing the camera axis or the camera

speed while rotating. These options were discarded because it was not helpful to have those options in a platform game where the camera follows the player. After all, there is no need to see any other perspective. On the one hand, if the camera is following the player, constantly locating him at the centre of the screen, changing the axis of the camera will not be a substantial change that benefits the player. On the other hand, the rotation speed depends on the player movement, so if the movement is slow enough to move correctly, the camera will be rotating at a handy speed.

The third stage was implementing a selector for the difficulty, which is part of the cognitive accessibility options (See Figure 4.7a). As this is a platform game, the adaptable difficulty level is based on how fast the platforms move. There is a base difficulty level inherent on each level, defined by the number of enemies and healing kits provided to the player. This inherent difficulty is necessary to keep the game challenging while setting the easy mode on the selector affecting the platform movement. This feature is a multi accessible one because, at the same time, helping players with cognitive impairments can also be helpful to players with locomotion impairments. Each platform has three different values for the easy, normal and hard mode velocity, and the movement system assigns the corresponding value on runtime to modify the difficulty. This is very useful also to change the difficulty once the game has been started, so the players can adapt to the speed of the platforms if they get stuck at some point of the game.

The fourth stage, and the last one, was the implementation of the sound options (See Figure 4.8). This has to be developed as the last feature because it was needed to develop some unique sounds and find a soundtrack that fits the game style. It was needed that all the interactions and functionalities were implemented to list how many sounds it would be necessary to have to give enough feedback to the player. The implemented features were:

- Slider to adjust the background music volume.
- Slider to adjust the effect sound volume.
- An enemy indicator showed in canvas.

As shown in the analysis results, one valuable option from the users in the questionnaire was to adjust the volume of the background music and the sound effects separately. This is because the sound effects are used many times as feedback to the player from any interaction, and for the ones with hearing, impairments will be beneficial to adjust the volume of these sounds. It is also relevant to implement some feedback for the players with critical hearing impairments; any sound adjustment will not help. For this purpose, an enemy indicator was developed to show the player what position the enemies are coming from, as the sound feedback is not helpful. The indicator is represented as an arrow rotating in the head-up display. At the same time that the player or the enemies are moving, this arrow is updating its position to show if the enemies are in a determined

range. The player will see as many arrows as enemies around, and they will disappear if the enemies go far or get destroyed (See Figure 4.9).

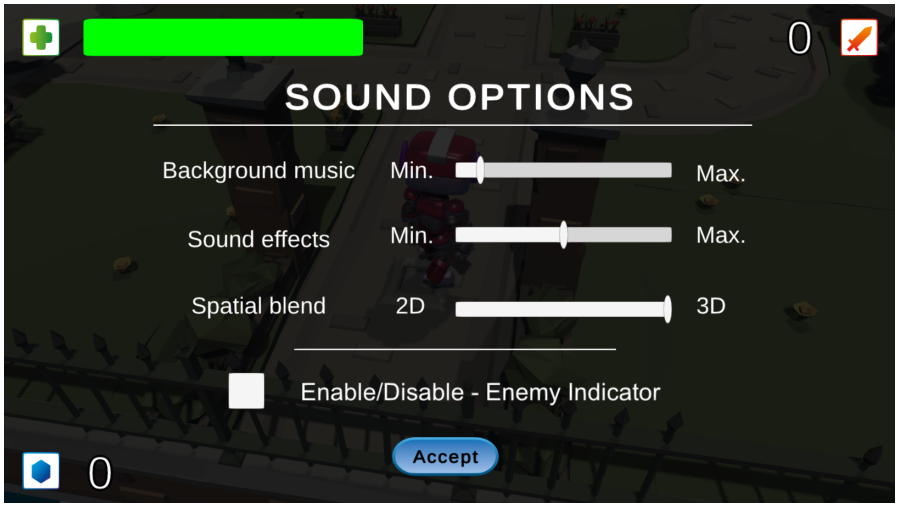
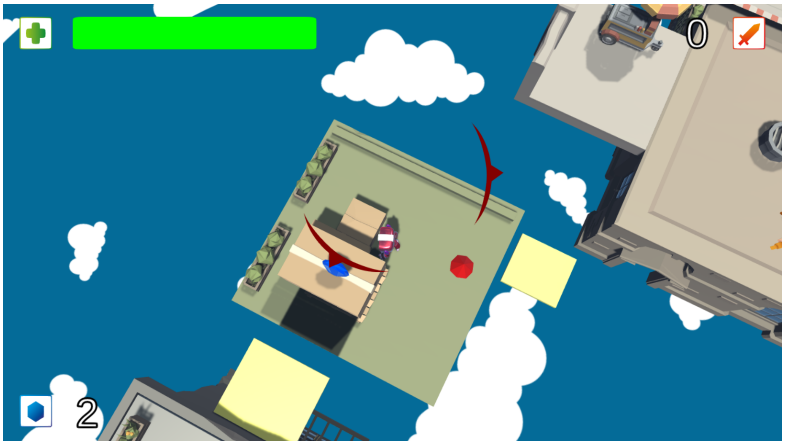


Figure 4.8: Overview of sound accessibility options.

All these features were implemented successfully while designing and assembling the whole tutorial environment and general game mechanics. The implementation of these game mechanics started with the user interface interactions and the player movement at the beginning while designing the visual features as a base to work on the other game mechanics. Then, the interaction with game elements such as collectable items and platforms was developed, and finally, the enemies' artificial intelligence. All the enemies were developed using the navigation system of Unity to ensure a patrol movement, which is helpful to escalate the difficulty of the enemies and its variety. Level transitions were the last implementation.

4.2.5 Design and implementation of the different menus

The game menus needed special treatment in terms of design because they had to be usable and understandable to every player. When the game shows a menu, the background is darkened to keep the player focused on the menu information. The colours chosen for the menus were different types of orange, blue, black and white. This is because the menus were also designed around the visual and cognitive impairments that some players may have, like colour blindness. The font used is a universal font that may be not very original but helpful to some impairments such as dyslexia. The buttons that involve critical decisions such as closing or saving the configuration also added a graphical symbol to refer to the usability of the button.



(a) Two enemy damage indicators

(b) One enemy damage indicator

Figure 4.9: Overview of the enemy indicator which shows where the enemies are.

4.3 Second phase: Testing the first prototype of the game with target users

This phase consists of testing the prototype developed in the first phase. To gather information about the game, a questionnaire was designed and its results were used to improve the prototype.

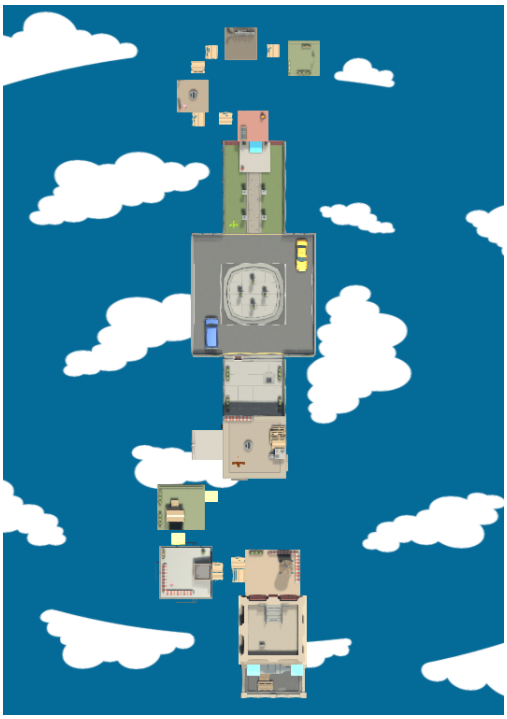
4.3.1 Developing the questionnaires and testing

Once the tutorial was finished, the next step was to send it to the users, but instead of that, the development of the first and second level was also finished to be able to send a completed game. This was not planned at the beginning, but it would be better if the players can experience the whole game instead of only a piece (See Figure 4.10).

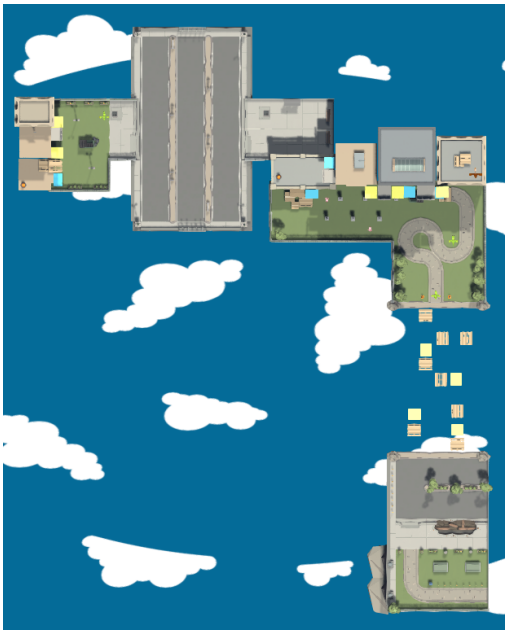
Furthermore, all the suggestions or new design solutions can also be applied to level one and two and not only to the tutorial. For this purpose, it was necessary to develop a solid questionnaire to approach the users' feedback while testing the game. According to the *Interview Techniques for UX Practitioners: A User-centered Design Method* [50], there are many different methods to approach a testing phase. In this case, the most suitable method is the semi-structured interview; due to the current restrictions, questionnaires will be a list of predefined questions and sent to the target users. The reason to choose this method is that semi-structured interviews are used when there is some knowledge about the topics under investigation, but it still needs to look into details. The method also provides quantitative and qualitative data, which in this case, qualitative data is the most relevant. It will also help gather data when there is no possibility to observe the users' behaviour directly, as in this case, all the process will be done online.

The predefined questions were answered by choosing a single value from six standard qualifiers such as: *Very good*, *good*, *standard*, *bad*, *very bad* or *non-answer*. In the beginning, the user can write if he has any disability or impairment, but this is an optional question. Also, at the end of the questionnaire, the user could provide any additional information to give some general feedback about any aspect of the game. The set of developed questions was:

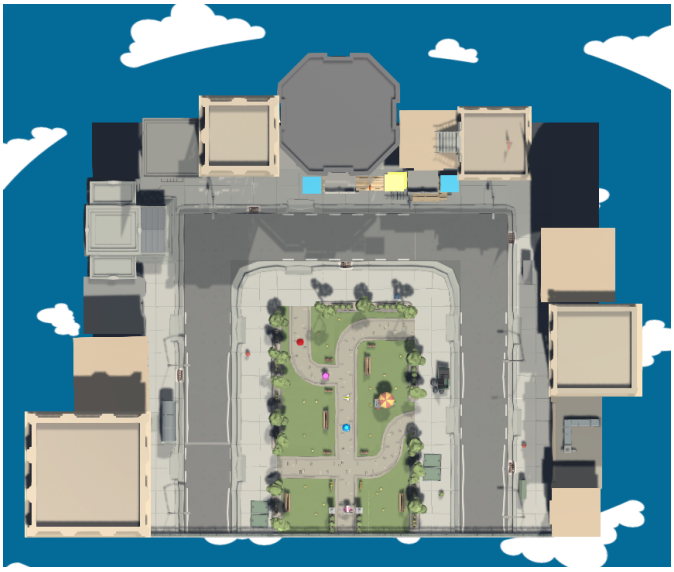
1. Do you have any impairment or disability at the time answering this questionnaire? You can describe what kind here if you want.
2. How was the navigation through the menus.
3. How was the player movement and interactions.
4. How funny was, in general, the game.
5. How was the performance of the game.
6. How easy to follow were the tutorial configuration steps.
7. How helpful were the visual options.
8. How helpful were the cognitive options.
9. How helpful were the locomotor options.
10. How helpful were the sound options.
11. How was the visual feedback from the game.
12. How was the sound feedback from the game.
13. How challenging was the game.
14. How was the visual aspect of the game.



(a) Top view level 1 stage



(b) Top view level 2 stage



(c) Top view tutorial stage

Figure 4.10: Overview of the final design of the game stages.

15. Use this question to write about any aspect related to the game; you can also make your suggestions about accessibility or any aspect of the game.

This questionnaire was developed using Google Forms, which was very helpful to organise all the answers at the end of the testing phase.

4.3.2 Results from the feedback

The questionnaire was open for ten days, and 47 users sent their answers, which was more than expected. All the users, besides two, answered the first question about their kind of impairment or disability. Here is a brief summary of the results:

- Twenty-two users with visual impairments, such as colourblindness, and different types of lack of vision.
- Thirteen users with sound impairments, such as deafness, and is hard to hear from one side.
- Six users with cognitive impairments such as dyslexia, attention-deficit/hyperactivity disorder and memory loss.
- Four users with locomotor impairments such as arthritis and muscular dystrophy.
- Two users did not answer.

This data can also be very helpful to understand the rest of the answers because it is significantly related to the special needs that this kind of users will be looking at in the game. Also, it is important to remember that one of the objectives of this thesis was to use a multi accessible point of view in order to develop all the features. This is the reason why the first question was a voluntary appointment. If the users did not answer, it would also be helpful to read the gathered data about all the game aspects in the rest of the questionnaire.

As a result of the gathered data, a list of features was planned to achieve the questionnaire's users' suggestions. This list showed some changes that could be done in the current systems of the game and would not take much time to implement them. The most mentioned features to improve where:

- The text size is still tiny. It is necessary to increase the biggest value of size.
- Camera movements are confusing and fast, especially during the second level.
- It would be nice if all the accessibility features could be accessed by any game level and not only from the tutorial.

Moreover, users suggested a lot of interesting features that would be analysed before deciding on implementation possibilities. This kind of features had to be evaluated to see how they will affect the general gameplay or the technology restrictions. The most mentioned were:

- Change the sound from stereo to mono to be able to hear all the game sounds only from one side.
- Change the difficulty selector and make it available to change the platforms' waiting time apart from the velocity.
- Develop a reset button for all the bindings that set the default value.
- Push the player back while being hit by an enemy in order to send more feedback.

4.3.3 Conclusions and Implications of the testing phase

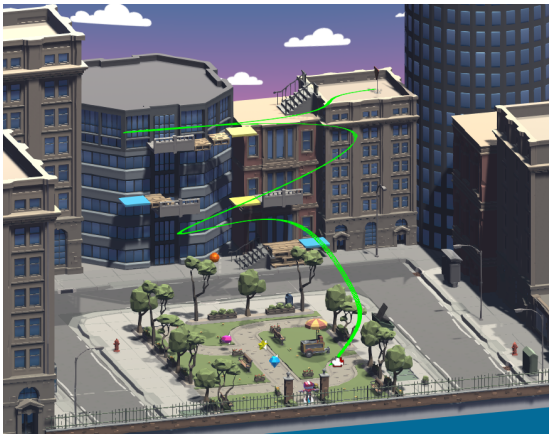
This second phase was the most tricky because it was very dependent on the participation of users. Besides that, users were collaborative and conscientious of the objective of the thesis, and the results were beneficial to the project itself. In the beginning, it was uncertain how many users would participate because the environment of the study was so impersonal due to the health crisis. However, thanks to all the technologies from nowadays, it is becoming easy to develop processes like this. Time was also a restriction, but as this is a project set in the educational environment, the outcome was gratifying and proof that it can be developed professionally.

4.4 Third phase: Implementation of new solutions for the final prototype and final results

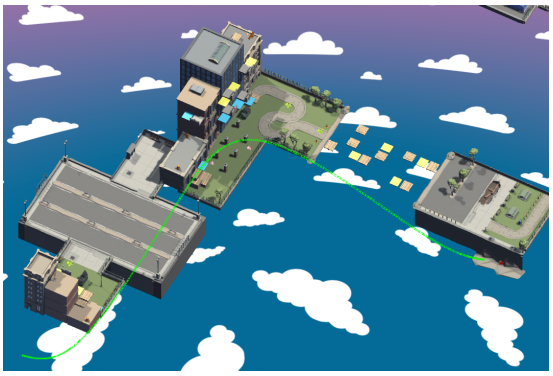
Once the suggestions were listed and classified, it was necessary to analyse what we are going to implement. To decide it, there were two important variables to take into account: the technological restrictions and the available time. The suggestions to improve the current features were the first to analyse because its implementation was to add changes to the systems that had been already implemented.

For the text resizing, it was only necessary to increase the maximum threshold. Also, to access the accessibility menus, another button at the pause menu showed all the three accessibility menus and allowed the user to reconfigure everything at runtime. The only feature that was longer to develop was the camera movement. As the current movement was fast and confusing, the new implementation uses another feature of the Cinemachine package to create a virtual dolly track (See Figure 4.11) to guide the camera through the environment. This allows the design of a preconfigured path to ensure the camera does not move as much as before.

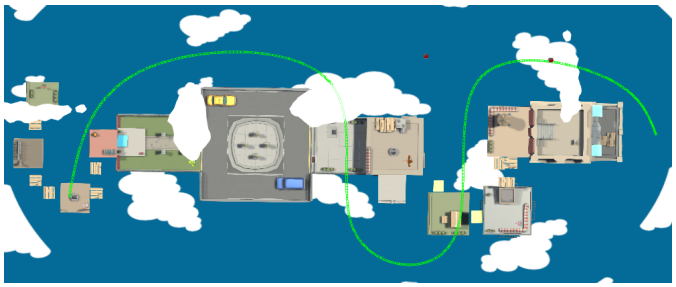
After changing the mentioned features, it was time to decide which suggestions will be implemented. The mono/stereo sound was complicated not in a technical way but in a usability way. This feature was about changing from mono to stereo, which involves the audio sources of the game and the audio device used to play. This means that the experience will be variable using different devices, and the change of the mono/stereo



(a) Dolly track tutorial



(b) Dolly trach level 2



(c) Dolly track level 1

Figure 4.11: Overview of the dolly tracks made using Cinemachine.

mode will not be very helpful. Besides that, another slider was implemented to change the sound from a 2D mode to a 3D one. This would be helpful because, in the 2D mode, every audio source would be heard at the same distance, and the player would be able to hear every sound, even if his impairment is to not hear from one ear.

Following the suggestion listed above, to implement the changes in the difficulty selector, it was necessary to add another list of three values to each platform that will represent the waiting time on each mode. The system designed for the speed mode was also used to change the waiting time, so this suggestion was finally not even difficult to implement.

Developing a general reset button for the rebinding panel was also quite fast because its functionality was already done in the individual rebinding buttons. This suggestion was not proposed initially because it would not provide anything worthwhile, but a user suggested the testing phase, so it was implemented.

The last feature of the suggestion list was something like a "knockback" to push the main character while being attacked by and enemy to warn the player of being damaged

again. This was implemented by adding a force calculated in the opposite direction from the force source to push the player back. Once all the features were implemented, the second prototype was already finished.

4.5 Results

4.5.1 Objectives reached

At the beginning of this document, the objectives of the whole project were presented as four main objectives. It is gratifying to express that almost all the objectives were reached, but some just in some points.

Implementing the user-centred methodology was an exciting proposal initially, and the adapted way to apply it to this project was very challenging. This methodology requires an iterative process to reach the most completed prototype possible, but it only was possible to do one iteration due to time constraints. This does not mean that the objective was not reached, only that with more time to be closer to the users and iterate around the prototype, the final product would reach the users' expectations at the highest level.

Using Unity Engine to develop this game was also challenging in some ways due to the requirements to use extra packages to implement some functionalities, but in the end, no complex technologies out of those packages were used, so the second objective was also reached.

Related to the accessibility menu, it was an interesting opportunity to prove that a well-integrated menu is an easy way to offer the accessibility options of our games to the players, showing that sometimes usability it is everything. During the testing phase, users appreciated this kind of tutorial that includes the configuration, proving that it was a helpful design decision and reached this objective.

Finally, about the multi-accessible experience, it was also challenging at the beginning because the starting perspective was that a more accessible game should include more options. Thanks to the analysis and the users' feedback, the final prototype is a multi accessible product, which implements three accessibility menus with enough configurable options, always considering the technological restrictions.

4.5.2 Planning changes

Despite planning every detail of this project, it was needed to do some changes during the process to improve the first and second phases. There were two crucial changes in the planning:

- Adding a questionnaire also to the designing phase, due to the good relationship established with the users participating in this thesis.
- Developing level 1 and level 2 to add them to the prototype for the users testing phase.

During the project's development, both changes were made after getting more informed about this field because they could help me get more information about users requirements and users' feedback. Despite not being in the initial planning, other tasks were not highly affected, only adding some days of extra work.

CHAPTER 5

Conclusions and Future Work

Contents

In this chapter, the conclusions of the work, as well as its future extensions are shown.

5.1 Conclusions

The video games industry is proliferating, and some big studios are focusing their efforts many times only to expand economically. Video games are becoming one of the most common forms of entertainment around the world, and it is necessary to be aware and start developing more inclusive products, not for economic benefits but social well-being.

Developing this project was an opportunity to focus on a field of personal interest and demonstrate that as a game designer with no special resources, everyone can contribute to an inclusive industry. It was a challenging project because despite thinking that user-centred design will be the best methodology to develop the game features, this implies interacting with people and trying to understand their point of view, and many times it is not easy to feel what they feel.

At the beginning of this project, I was worried about the implementation part, which needs an excellent level of programming. During the degree, I have not been confident with the programmer role and always focused on designing or artistic roles at the group projects. However, during this project, I faced my insecurities and tried my best because

I really believe in my initial idea of the project. In the end, I am delighted with the results, and I have learned about many different things related to the programming area. Also, I am very grateful to the people that helped me and have participated in this research. Nothing could have worked without their goodwill.

5.2 Future work

This project was developed with an adaptation of user-centred design methodology, but the next step would be trying to expand the game accessibility features, using a more extensive testing phase in which the interviews could be more personalised. The knowledge acquired with this project was beneficial to try to become an accessibility specialist in the future.

Bibliography

[1] AbleGamers. The able gamers charity. https://ablegamers.org/. Accessed: 2021-05-14.

[2] AbleGamers. Includification: A practical guide to game accessibility. https://accessible.games/includification/. Accessed: 2021-05-14.

[3] Adobe. Adobe illustrator official web page. https://www.adobe.com/products/illustrator.html. Accessed: 2021-05-14.

[4] Nintendo Entertainment Analysis and Development. The legend of zelda: Twilight princess. Nintendo GameCube, Wii, Wii U, Nvidia Shield TV 16 nov. 2006. Accessed: 2021-05-14.

[5] Nintendo. Nintendo Entertainment Analysis and Development. Mario kart 8. Wii U, Nintendo Switch. 29 may. 2014. Accessed: 2021-05-14.

[6] Jen Beeston1. Christopher Power. Paul Cairns. Mark Barlet. Characteristics and motivations of players with disabilities in digital games. Department of Computer Science Deramore Lane, University of York, Heslington. Accessed: 2021-05-14.

[7] Sam Barlow. Her story. Android, iOS, Microsoft Windows, Mac OS. 23 jun. 2015. Accessed: 2021-05-14.

[8] EA Digital Illusions CE. Battlefield 1. PlayStation 4, Xbox One, Microsoft Windows. 21 oct. 2016. Accessed: 2021-05-14.

[9] Cinemachine. Cinemachine, about cinemachine documentation. https://docs.unity3d.com/Packages/com.unity.cinemachine@2.1/manual/index.html. Accessed: 2021-05-14.

[10] Accessibility Guidelines collaborative group. Game accessibility guidelines. http://gameaccessibilityguidelines.com/. Accessed: 2021-05-14.

[11] Fiddlesticks. Curve Digital. Hue. Windows, PS4, Xbox One. 30 aug. 2016. Accessed: 2021-05-14.

[12] Discord. Discord official web page. https://discord.com/brand-new. Accessed: 2021-05-14.

[13] Naughty Dog. Uncharted 4: A thief's end. PlayStation 4. 10 may. 2016. Accessed: 2021-05-14.

[14] Square Enix. Final fantasy xiv. Microsoft Windows, 30 sep. 2010. Accessed: 2021-05-14.

[15] Germán Fabregat Llueca. Guía para la redacción de las memorias. http://mermaja.act.uji.es/itis/IS31/guia_memoria.pdf. Accessed: 2021-04-28.

[16] Epic Games. People Can Fly. Fortnite. PlayStation 4, Nintendo Switch, Xbox One, Android, iOS, Microsoft Windows, Mac OS, 21 jul. 2017. Accessed: 2021-05-14.

[17] Google Forms. Google forms about web page. https://www.google.com/forms/about/. Accessed: 2021-05-14.

[18] Blizzard Entertainment. Iron Galaxy. Overwatch. PlayStation 4, Xbox One, Nintendo Switch, Microsoft Windows. 3 may. 2016. Accessed: 2021-05-14.

[19] Github. Github desktop dowload archive. https://desktop.github.com/. Accessed: 2021-05-14.

[20] Grammarly. Grammarly: Free online writing asistant. https://app.grammarly.com/. Accessed: 2021-05-14.

[21] Anna Cavender. Shari Trewin. Vicki Hanson. Accessible writing guide. SIGACCESS Newsletter, 2008. Updated in 2015. Accessed: 2021-05-14.

[22] Bei Yuan. Eelke Folmer. Frederick C. Harris. Game accessibility: a survey. Universal Access in the Information Society volume 10, pages 81–100 (2011). https://rdcu.be/ckJ1t. Accessed: 2021-05-14.

[23] Michael Heron. Inaccessible through oversight: the need for inclusive game design. The Computer Games Journal. Whitsun. 2012. https://doi.org/10.1007/BF03392326. Accessed: 2021-05-14.

[24] Dontnod Entertainment. Feral Interactive. Life is strange 2. PlayStation 4, Xbox One, Microsoft Windows, Linux, Mac OS. 26 sep. 2018. Accessed: 2021-05-14.

[25] Nintendo Company Limited. Nintendo official web page spain. https://www.nintendo.es/. Accessed: 2021-05-14.

[26] Matt Makes Games. Extremely OK Games Ltd. Celeste. Nintendo Switch, PlayStation 4, Xbox One, Microsoft Windows, Linux, Mac OS clàssic, Google Stadia. 25 jan. 2018. Accessed: 2021-05-14.

[27] Gearbox Software. 2K Australia. Aspyr Media. Borderlands: The pre-sequel. PlayStation 3, Microsoft Windows, Linux, Mac OS 14 oct.2014. Accessed: 2021-05-14.

[28] Microsoft. Xbox adaptive controller. https://www.xbox.com/es-ES/accessories/controllers/xbox-adaptive-controller. Accessed: 2021-05-14.

[29] Massive Entertainment. Ubisoft Milan. Tom clancy's the division 2. PlayStation 4, Xbox One, Google Stadia, Microsoft Windows. 7 feb. 2019. Accessed: 2021-05-14.

[30] Ubisoft Montreal. Ubisoft Quebec. Ubisoft Singapore. Ubisoft Milan. Assassin's creed: Odyssey. PlayStation 4, Xbox One, Nintendo Switch, Google Stadia, Microsoft Windows. 2 oct. 2018. Accessed: 2021-05-14.

[31] World Health Organization. World heatl organization official web page. https://www.who.int/. Accessed: 2021-05-14.

[32] World Heath Organization. International classification of impairments, disabilities, and handicaps : a manual of classification relating to the consequences of disease. Published in accordance with resolution WHA29.35 of the Twenty-ninth World Health Assembly, May 1976. https://apps.who.int/iris/handle/10665/41003. Accessed: 2021-05-14.

[33] Overleaf. Overleaf online latex editor web page. https://www.overleaf.com/. Accessed: 2021-05-14.

[34] Nintendo Entertainment Planning and Development. Super mario odyssey. Nintendo Switch. 27 oct. 2017. Accessed: 2021-05-14.

[35] Chadia Abras. Diane Maloney-Krichmar. Jenny Preece. User-centered design. In Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications (in press). 2004. Accessed: 2021-05-14.

[36] Text Mesh Pro. Text mesh pro, user guide. https://docs.unity.cn/Packages/com.unity.textmeshpro@3.0/manual/index.html. Accessed: 2021-05-14.

[37] Gantt Project. Gantt project donwload archive. https://www.ganttproject.biz/. Accessed: 2021-05-14.

[38] Ubisoft Montreal. published by Ubisoft. Assassin's creed: Valhalla. Google Stadia, Microsoft Windows, PlayStation 4, Xbox One, Xbox Series X/S. 10 nov. 2020. Accessed: 2021-05-14.

[39] Graham Pullin. Design meets disability. The Computer Games Journal. Whitsun. 2012. https://doi.org/10.1007/BF03392326. Accessed: 2021-05-14.

[40] Bethesda Softworks. Doom eternal. PlayStation 4, Xbox One, PlayStation 5. 2020. Accessed: 2021-05-14.

[41] EA Tiburon. EA Sports. Madden nfl 17. PlayStation 4, Xbox One, Xbox 360. 23 aug. 2016. Accessed: 2021-05-14.

[42] Google Spreadsheets. Google spreadsheets about web page. https://www.google.com/sheets/about/. Accessed: 2021-05-14.

[43] Unity Asset Store. Unity asset store official web page. https://assetstore.unity.com/. Accessed: 2021-05-14.

[44] Microsoft Visual Studio. Microsoft visual studio official web page. https://visualstudio.microsoft.com/es//. Accessed: 2021-05-14.

[45] Mojang. Other Ocean Interactive. 4J Studios. Xbox Game Studios. Minecraft. Xbox One, Microsoft Windows, Linux, Mac OS. 18 nov. 2011. Accessed: 2021-05-14.

[46] Input System. Input system manual. https://docs.unity3d.com/Manual/com.unity.inputsystem.htm Accessed: 2021-05-14.

[47] Unity Technologies. Unity tecnhnologies official web page. https://unity.com/. Accessed: 2021-05-14.

[48] Trello. Trello official web page. https://trello.com/. Accessed: 2021-05-14.

[49] Twitch. Twitch official web page. https://www.twitch.tv/. Accessed: 2021-05-14.

[50] Chauncey Wilson. Interview techniques for ux practitioners: A user-centered design method. Newnes, 23 nov. 2013. Accessed: 2021-05-14.