Catch the Thief: An Approach to an Accessible Video Game with Unity

Edwin Patricio Flores-Garzón[a,1], Luis José Intriago-Echeverría[a,2], Angel Jaramillo-Alcázar[a,3], Santiago Criollo-C[a,4], Sergio Luján-Mora[b]

[a] Facultad de Ingenierías y Ciencias Aplicadas, Universidad de Las Américas, Quito, Ecuador
E-mail:[1]edwin.flores@udla.edu.ec; [2]luis.intriago.echeverria@udla.edu.ec; [3]angel.jaramillo@udla.edu.ec; [4]luis.criollo@udla.edu.ec

[b] Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Alicante, España
E-mail: sergio.lujan@ua.es

Abstract— Today, the video game industry is one of the most profitable business markets in the world. Video games are not only being used as a means of entertainment but also to reinforce education. Even though there are unbroken barriers for disabled people to use this kind of applications. Lack of accessible technologies and functions are real problems and a way of discrimination. It is a challenge for every software development organization, even for those who focuses in video game line of work. Many impaired people enjoy playing games in despite of their disabilities; however, some limitations appear when they start to play. This article presents an approach for an accessible video game developing, using Unity Engine and some of its accessibility complements to implement some functions to get better the player experience. This way, people who suffer of visual and hearing disability can be able to play and learn. Within the spectrum of disabilities this project covers are; visual and hearing, multiple variants of color-blindness and reduced vision problems. A series of settings options will have implemented with the final purpose of giving users an easier way to interact with the video game. It should be emphasized that game mechanics are based on various parameters to offer accessibility as brightness reduction, contrast and font-size adjustment, and more. Disability simulation tests will have done in order to prove the video game functionality. This research tries to increase the accessibility for people with impairments in the world of video games.

Keywords— accessibility; video games; software development; education; impairments.

I. INTRODUCTION

Video game industry was born in the 70s, with releases like Pong, Spacewar, among others [1]. In 2007, the industry had already grown $ 35 billion in annual revenue, just 11 years later; the value of revenue for gaming line of business grew up by $100 billion dollars, generating $ 137.9 billion income for 2018 [2]. Playing is a fundamental activity for human development. Commonly people play for fun, entertainment even for education. In the book, "The Art of Game Design", the author affirms that it is a vital function for any people capable of learning [3].

Nowadays, video games industry is generating many profits and revenues, becoming a great source of entertainment and learning. Plenty of these video games recreate fantasy, simulation, or virtual environments that millions of people enjoy around the world. Driving gamers to improve some skills such as creativity and imagination, which is why, it had been propounded to use them as an alternative way for education [4].

In case of serious games production, it is a little different because the main focusing for these kind of video games is learning. In 2016, the serious games market represents the 3% of the industry generating revenues of $ 2.73 billion dollars in comparison to the $90.6 generated by video game industry [5]. One of the main reasons for this notorious difference is people ignorance about serious games existence.

The first step for the development of video games is planning or preproduction, this is where the members of the developer team consisting mainly of: programmers, artists and designers, create the initial documentation that will carry out the development as such. This preliminary process of product creation can be divided into three stages or deliverables [6]: high concept, presentation and concept. The first document is a brief description, the second is an offer or proposal that presents the strengths that can be used to sell the game and the reasons why it would be profitable, and the third is already a document with more detail that includes the information that is generated by the video game (genre, description, history, public, estimated calendar, etc.)

After the planning phase comes the production stage, which has also changed not only due to technology but also to the wide variety of video game genres. In this period the necessary tools and the interaction of the different parts of the equipment are used to give life to the product. There are many processes that converge to launch a "final" product such as programming, level design, artwork; which are the assets of the video game created by the artists. Basically, the aesthetics of the game. Also, the musical and sound environment joins the development where sound effects are involved that will be part of the game universe as; character voices if applicable, soundtrack, etc. Finished the video game where you could say that it is "playable", it goes to the stage known as testing, in this phase the game goes through different periods known as: Pre-Alpha, Alpha and Beta.
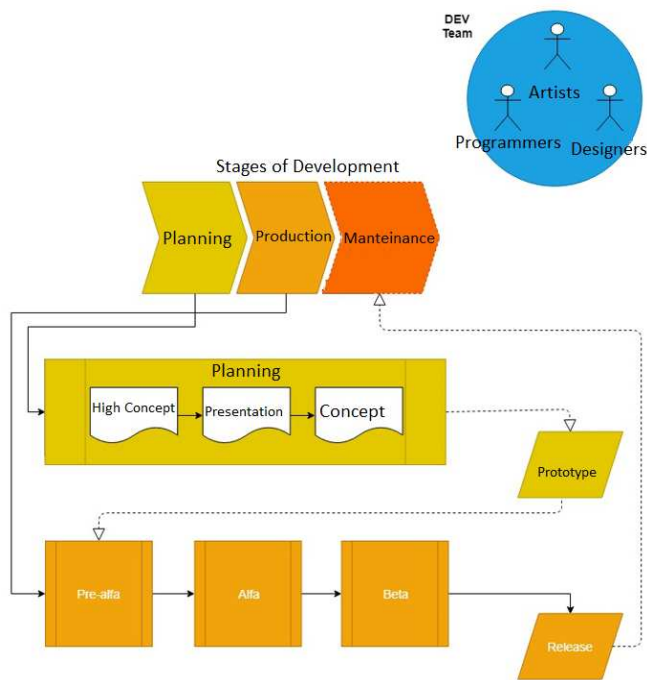

Fig. 1 Video Game Stages of Development

The Pre-Alpha phase are the processes that are carried out before the testing stage, from development and design, to unit tests of each of the functionalities. The Alpha phase begins when the product passes to the testing team, who are responsible for evaluating and ending the game's functionalities, as well as bug fixes, among others. Additionally, in this phase more functionalities can be added for the outcome of the project. After the Alpha version, it is considered that the game is in a stage of mature development, it is at this stage when the increase or addition of new features and functionalities to the video game is frozen. The beta phase of the product can be opened, that is, it is available to the general public; or closed, where the testing team are selected users or who gain the privilege of being beta testers of this version of the game, the selection method differs according to the publisher of the game. We can see all the process in Figure 1.

It could be considered that, after correcting the small errors that should appear in a Beta, the game is probably ready to be launched, after this you can really see the

reception it has, and depending on the game, it enters a stage called Maintenance. More and more games entail this phase due to the reception that online games have had and all their derivations, since these require a follow-up by a support team and constant updates.

A. Education in video games

In general, playful activities implementation to develop people skills is not an actual notion; in fact, lot of research had been done years ago just to prove the potential of entertainment as apprenticeship. Showing as result, game developing based on simulations.

In terms of its classification, it is hard to explain due to there is no substantial information about. It may say that serious games might been catalogued on rating lists like PEGI (Pan European Game Information) or ESRB (Entertainment Software Rating Board). However, according its teaching method serious games could be grouped in two types: educational games based on enjoyment, known as Edutainment (Education Entertainment); and the ones that fun it is not a priority. Simulations, games based on information transference and even promoting ideologies trough advertising, belong to the last group previously mentioned [7].

B. Accessibility in Video Games

Video games accessibility is an undeveloped field, since games with the strongest attainability option are developed for a specific segment of people and not for general public [8]. Even though accessible options could be implemented, there are not part of the main focusing in game coding yet. In some way, it's hard to imagine all the favorite video game themes with these sorts of programmable settings. Even so, some authors propose some accessibility guidelines for video game design [9]–[12].

Nowadays, people interest is increasing in video games, as result as they are currently used by the entertainment, education, rehabilitation and health industries. There are many characteristics and parameters to consider; due to there are different types of disabilities and diseases that reduced motor, visual, hearing and cognitive or intellectual capacity, even the people that suffer color-blindness [13].

Nowadays, people with disabilities have much more difficulty understanding the world and its environment, in the way that anyone without any disability could do it easily [14]. It is well known that a serious game gives more importance to knowledge beyond of entertainment. At present there are very few games that have a mode of accessibility, whether they are focused on entertainment or serious games, taking into account that a serious game seeks to give equality to the majority of users, so that they can develop some type of learning taking advantage of a full experience of it [15].

C. Visual Impairments

Approximately 1.3 billion people live with vision impairments [16]. According to the International Classification of Diseases, the World Health Organization (WHO) classifies vision impairments in two notable groups, considering the distance and how near the objects display.

Distance vision impairment:
- Mild – presenting visual acuity worse than 6/12
- Moderate – presenting visual acuity worse than 6/18
- Severe – presenting visual acuity worse than 6/60
- Blindness – presenting visual acuity worse than 3/60

Dalton's first scientific paper described his and his brother affection called Daltonism, evaluating a defective perception of red and green colors. The way people notice colors and recognize them. Daltonics cannot perceive colors correctly, depending on the type of color blindness, in most cases the colors green, red and blue [17].

1) Protanopia: People with this affectation have defective long wavelength cones in the retina of the eye, that difficulty distinguishing between the colors blue and green and between the colors red and green [17].
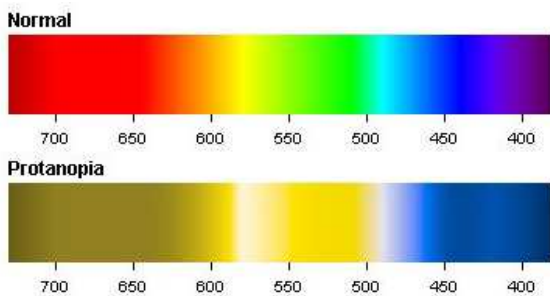

Fig. 2 Protanopia Color Spectrum

2) Deuteranopia: A deuteranopic person has issues to distinguish red and green color from normal color Spectrum. In this kind of colorblindness, the medium wavelength sensitive cones (green) are missing or failing in the retina [17].
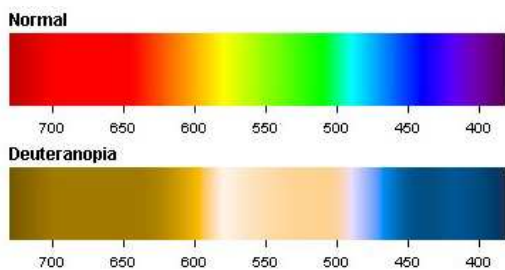

Fig. 3 Deuteranopia Color Spectrum

3) Tritanopia: Tritanopia is a type of dichromatism caused by short wave cones. This impairment in vision causes an incorrect perception of the blue color. Tritanopic people only have the medium and long wave cones [17].
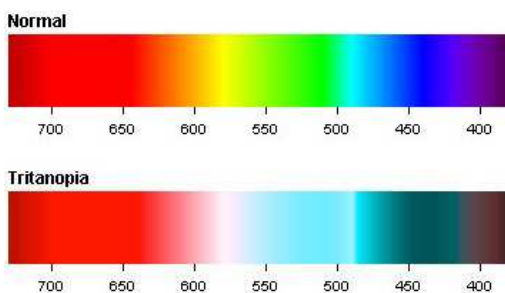

Fig. 4 Tritanopia Color Spectrum

D. Hearing Impairments

Hearing impaired people often have difficulties in their day-to-day life, in other words, they cannot hear plenty of sounds or even they don't hear anything [18]. Many times, this type of disability usually influences the rest of the senses such as body sensitivity, balance, vision, ability to do lip Reading [19]. Some problems most of the time lead to poor reading comprehension, difficulty in oral expression and grammar, and often has repercussions on the ability to understand and express during a conversation.

According to the WHO [20], there are four grades of hearing impairment or hearing loss, which are classified on a decibel scale:
- Slight/mild hearing loss (26 - 40dB).
- Moderate hearing loss (41 - 60dB).
- Severe hearing loss (61 - 80dB).
- Profound hearing loss (over 81dB).

This article is organized as follows: In section 4 is explained the methodology used to design the video game and, we present the accessibility features implemented in the video game. In section 3, an analysis is made according to the results obtained. Finally, in section 4, some conclusions are presented and future work is proposed.

II. MATERIAL AND METHOD

Extreme Programming (XP) is an agile software developing technic, which has a short-term production approach. This method puts forward the progressive presentation of release versions known as iterations. Each one contains a set of proceedings, which are planning, designing, coding, and testing [21].

Typically, XP-based developments tend to have between 10 and 15 iterations. Each of the iterations covers all development processes (analysis, design, implementation and testing). This process makes the project life cycle very dynamic. In addition, to ensure that the release version of each iteration is stable, four principles are established which govern this methodology: cost, time, quality and scope. In addition to this, it is considered that the client (s) are not able to define the program requirements from the beginning [22].
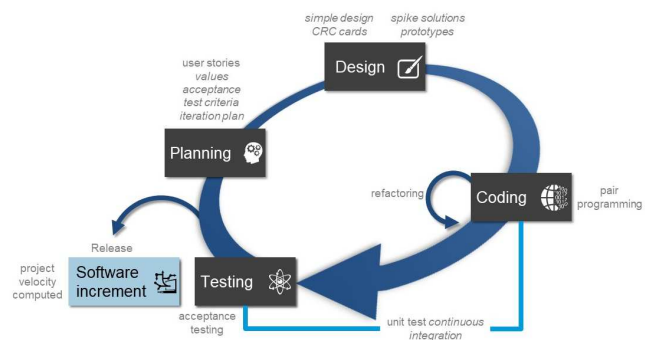

Fig. 5 XP Workflow

In order to perform this project in the right way, XP software developing methodology guidelines will be used. A summary of each principal process is going to be shown; due to software and video game development, is not this article intention.

A. Planning

Planning procedure is probably the most important step in software developing, such as it is in XP. Its importance is attributable to all the analysis that should be done in this phase, like the functional information compilation and stuff. In XP the first move is creating user stories. For this project a total of twenty of these stories had been created, that is also the recommended number of them in order to make a good design. After user stories are done, a schedule known as "release plan" is generated as result of user stories making. This timetable is a developing time estimation. It contains a timeline with each iteration lifespan. Once release plan is finished, iteration planning start. This layout takes part once every iteration begins. A programming order is established based on iteration user stories priority, including failed ones of passed iterations. This process gives place to task cards, which are like user stories for programmers [22].

1) User stories. User stories are initially a brief description by the client. This makes the focus on the user's needs, leaving aside technical details that may arise in the planning. The main function of user stories is to prepare a delivery schedule, which is why a high level of detail is not expected in the description by which an implementation time can be calculated.

2) Delivery plan: A release plan is born from the meeting of interested parties for technical decision making as well as the business itself. Each user story is evaluated to determine a time in weeks of their implementation, creating a development schedule where all parties can commit. In addition, it is important to define a plan without underestimating times, since it can lead to problems in the other phases of the methodology. Especially in the iteration planning, since it is generated from the order defined in the delivery plan.

3) Iteration plan. This planning is carried out at the beginning of each iteration. An order and a programming plan based on the priority of each user story designated by the client are proposed, including those that have not passed the acceptance test of the previous iterations. In this planning task cards are detailed for each iteration, which are like user stories, but for programmers.

4) Stand-up meeting. For the correct development of the development team, daily meetings called stand-up meetings are proposed. Unlike the typical project meetings, where the contribution of the participants is minimal and a lot of development time is lost, the daily meetings proposed by XP are small meetings where promptly it should be informed: what processes were completed the previous day, processes that are planned for the current day and what problems are causing delay. To avoid discussions that last long periods of time, all attendees stand up and form a circle.

B. Design

An extreme programming design should be simple. In order to measure a subjective measurement as simplicity is, it is advisable to apply some qualities named TUBE (Testable, Understandable, Browsable, Explainable), at least for codification.

- Testable: could be able to prove and pass tests.
- Understandable: could been understood by anyone that is not part of the develop-er team.
- Browsable: all code has sense and order so it is easy to find.
- Explainable: could been explained by anyone that is not part of the developer team [22].

Additionally, there are other necessary concepts to know at the design stage:

1) Spike solutions: A spike solution is a small program developed to solve a specific problem that is presented in the project. Its purpose is to reduce the risk of technical failure or make the estimated time of a user story more reliable.

2) Refactoring: In XP this concept refers to the recoding of the program. By recoding it means not reusing code just because it works. Using this practice avoids redundancy and possible lines of junk code. Thus, the codification goes through a rejuvenation process, to finally have a clean and concise programming.

3) System metaphor: The metaphor system is more than a deliverable a good practice for program design. An understandable design is proposed for all the parts that are part of the project, emphasizing the simplicity mentioned above. It must have an easy to understand structure, so that if a new collaborator enters the project, it is easy to adapt to the design without having to review so much documentation. This includes the appointment of classes and methods in a way that can be intuitive and can function as a guide to the functionalities within the code.

C. Coding

XP propound a series of programming advice to follow the methodology in a right way. Notions like customer availability, pair programming, constant code integration and collective ownership are considered for this project development. Good practices such as code standardization usage must be part of it as well. The *customer availability* is considered as a requirement for XP to be properly applied, since it has a fundamental role throughout the development of the project. In the method, the client is not only the one who sets the requirements, but part of the team as such. XP promotes *pair programming*, since cooperative work is more functional than that of a single person. Considering that peer programming is not a social and non-technical skill, it is considered that there is difficulty or discomfort, but it is something to be accustomed to. To accomplish this purpose, it is recommended that two programmers work on the same computer, sitting side by side.

Constant code integration is essential. Each pair of programmers are responsible for integrating their work into the project. The integration of the code should be done preferably every hour, in case of not doing it with this regularity the changes should not wait to be integrated beyond one day. In the methodology, the *collective property* is proposed, through this definition the collaboration in the design and codification of the whole team is manifested. Something that is not common in other development methodologies. By establishing that, any programmer can make changes to any part of the code as classes or methods.

D. Testing

Testing procedure contains one of the main mainstays that supports XP methodology, test suite is proved in order to check that each functionality works properly. The number of tests is up on the developer team to decide, but exist two foremost demonstrations defined by Extreme Programming, which are unit and acceptance tests. Both are part of the developing video game in addition to the own project tests like feasibility, user acceptance and disability simulation ones.

1) Unit tests. It is considered a pillar within XP as it is recommended that unit tests be created before the code. For this, the first thing to do is to use a test framework compatible with the development IDE. These are part of the programming itself, as they must be in the same repository as the code that is tested. Moreover, both as classes and methods must have their acceptance test so that the iteration can be delivered, if this is not the case one must be created instantly. Implementing a unit testing tool is easier to find problems. In addition, these evaluations can be recoded, as they can verify whether a tiny change in the program affects functionality or not. Allowing to solve small problems that arise. This takes less time than solving major problems that arise near the delivery date.

2) Acceptance Tests. Acceptance tests are born from user stories. These verifications appear during the iteration plan, where the client tests the operation of the project in different scenarios, demonstrating that the functionality of the project is fully operational. There may be more than one acceptance test for each user story. In addition, any bug found during this phase of the program must be corrected at that time and generates an acceptance test.

E. Versions Control System

A version control system is a very useful tool for storing and managing versions of one or more files. Basically, it keeps the record in a database of the changes and the versions through which different projects have passed. In addition, it is a very versatile tool because it allows you to have total control of the files such as backing up files to previous versions, or completely reversing an entire project. There are several ways to use version control systems. Among which are: local, centralized or distributed.

F. Git and GitHub

Git is a version control system developed by Linus Torvalds in response to the need for a DVCS (Distributed Version Control System). The system was created to handle the versions and development of Linux because the program they used (BitKeeper) stopped offering their license for free, this caused it to lose relation to the development of the free code operating system. On the other hand, the video game accessibility implementation is described below:

G. Unity

Probably one of the most used video game engines, Unity is a tool that allows you to design, edit, program, compile, etc. Integrated into a single development environment. In addition to this, it is an expandable environment, that is, it works well with external development or design tools.

In addition to advantages such as high performance, or the variety of platforms it supports, it also has a large library of additional and customizable tools that match the different needs of programmers. This makes this engine a very complete platform for the development of the wide variety of video games that exist, and the reason why it was chosen for the development of this project. There are some useful concepts that developers need to know when they work with Unity:

1) GameObjects. In Unity this is the most important type of object, since each object of the game inherits its properties, since all the objects that are in the game are children of the GameObject class. In addition to the properties that the GameObjects have, each component has some special features that compose and characterize it, it should be noted that the GameObjects without the Behaivour Scripts would not perform any action.

2) Singletons and Shared Instances. It is a way of ensuring that a class has always only one globally accessible instance available. Behave like a regular static class, but with some advantages. This is very useful for creating global administrator type classes that contain global variables and functions that many other classes need to access. There are many advantages that Singletons have and that is to have a GameObject that is globally accessible, that is, it is not necessary to make a reference or search for the Object as such, other fundamental points are the persistence of data throughout the scenes.

3) PlayerPrefs. The PlayerPrefs are used to save variables that the game will use when initializing, with this the system will save all the data that is stored in PlayerPrefs. In order to start loading with the configuration previously saved, they also use it to store the score table in our case, Player preferences, everything stored in PlayerPref is stored in operating system records specifically in the HKCU\Software \folder [company name] \ [product name], where company and product names are the names configured in Project Settings.

H. UAP

UI Accessibility Plugin is a Unity component with the principal function of making an interactive interface for blind people [23]. Using developed components for Text To Speech (TTS) technology, which generate a synthetized voice from text labels. This component groups all the user interface elements in an "Accessible UI Group Root". For elements to be grouped they must have an accessible feature and belong to a father game object whit accessible root characteristic, in order to make interface parts usable when UAP is enabled. Figure 6 shows how to add an accessible feature to a button object specifically, the same process works for other elements such as labels, toggle buttons, slide bars, etc. Now UAP it's properly working only for English and German. Even though more languages could be settable, the TTS plugin should be programmed and configured by the developer team.
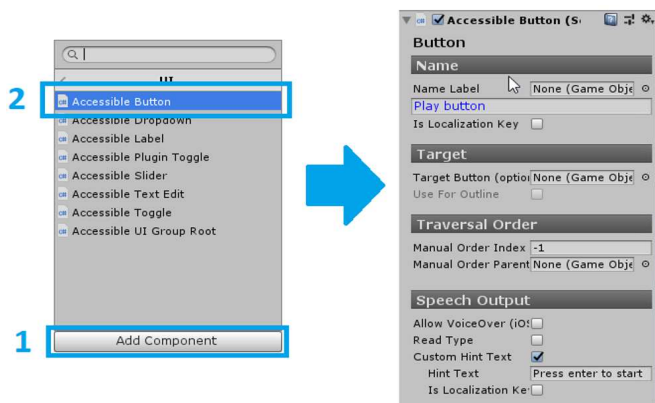
Fig. 6 Accessible features addition

The plugin allows navigation based on keyboard usage. Up and Down arrows are the buttons used for UI move inside, including Return key to interact with the elements mentioned before. When UAP is activated, the focus of the button with the highest priority is selected as can be seen in Figure 7.

Fig. 7 Enabling UAP plugin

Further than UAP usage, in order to accomplish more accessible guidelines [8] in this project decided the implementation of hotkeys for plugin's enable and disable functions. Considering keyboard typing characteristics it had been determined that F and J buttons are the most suitable for this functionality, due to a small relief these keys has. Bringing help for plugin activation to blind users if UAP is disabled.

I. Colorblind Effect

Colorblind Effect plugin main task is to apply filters and screen effects. The camera's image buffer is modified before it is displayed on the screen. It is a full camera effect that seeks to help color blindness impaired people by perceiving colors correctly. This helps to simulate the most severe cases of dichromatic vision, most colorblind people have absence of any of the 3 photo-pigments, which affects the way they perceive colors correctly [24].

The behavior script included in the Colorblind Effect Plugin it can be found in Uni-ty Asset Store. This code allows to select which kind of color-blindness the user wants to apply, but only from Unity's game editor. In order to access the editor variables, a method was developed letting modify the type of color blindness is set from the con-figuration menu window through a drop-down list control. In Figure 8 is explained the Daltonism settings.
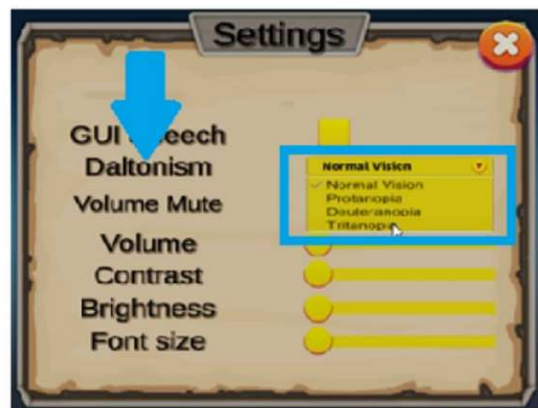

Fig. 8 Daltonism Setting

J. Post Processing Image

This plugin has a set of tools to improve the graphic quality of the game through configurations and image processing, applying filters and camera effects that modify the way the image is displayed [25]. The brightness and contrast settings are controlled through a Post Processing Pro-file different options, which loads the parameters into a Post Processing Behavior script. A rendering process applies for the camera configuration by modifying parameters from the Unity editor. In Figure 9 is presented an example without and with post processing image feature. However, in order to permit players to modify Post Processing parameters, a class named "Settings" was created, allowing Post Processing Profile instance access. That way, players can modify only the needed parameters. For the proper working of this task, data type established by the configuration slider is floating.

K. Video Game Logic Design

Within the design of the videogame you must know the logic that the game will follow, in order to capture it in the logical interfaces and sequences. For this you must be clear about the panorama and the problem that it will deal with, because this is the spine and moves all the scenes, cameras, objects, and more components, for this videogame the following flow chart will be used as the basis of implementation:
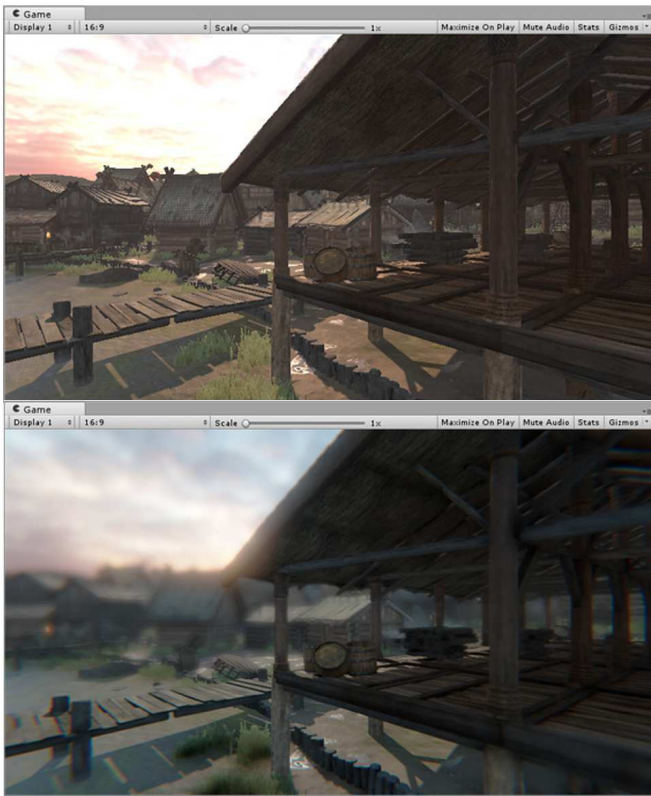
Fig. 9 Post Processing Image Feauture

In the logic of Catch the thief EC, several aspects must be validated, such as the correct question counter (n), the hours (Time). The first step is to verify that the time is not zero because if it is the case the player would lose the game and the GameOver window would run. If it is greater than zero, it is continued to evaluate if n is equal to 3, which means that the player has visited the three correct places after having followed the necessary clues. If, when performing the evaluation, the result is that if they are equal, the game would end up running the Winner window, otherwise the following window will be executed either a correct or incorrect place, in order to continue searching for clues to go to the correct place.
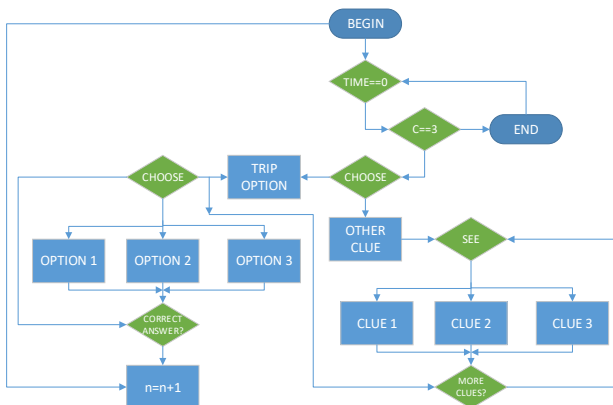

Fig. 10 Video Game Flow Diagram

L. Video Game Screen Design

The Design of the screens was carried out by means of Canvas and GameObjects to be able to perform a

manipulation of variables, since having in several Scenes the handling of variables, persistence of data and configurations is difficult.

1) Interaction Menu. This menu shows the four most important options during the game, the trip, track, help and configuration option, clicking on these buttons will open the different menus, following the logic of the program when executing the game will show us the place with a brief description, when entering the tracks the player can interrogate the witnesses in order to know where we have to travel.
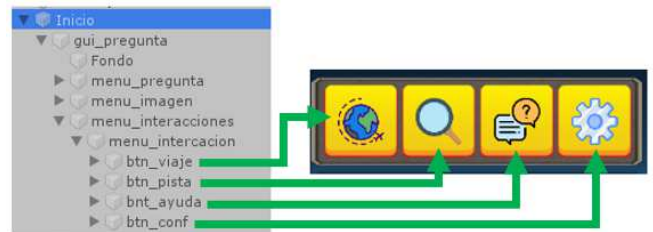

Fig. 11 Interaction Buttons

2) Clues Menu. When the track menu of each of the screens of the places is displayed, the player will have available a menu where he will be shown 3 places where he can interrogate the witnesses and obtain the necessary clues in order to decide, and travel to the right place.

Fig. 12 Interaction Buttons

3) Canvas Image. Within the image canvas or image menu the structure in the structure of each of the screens consists of the following items that are enabled when entering the different tracks, the canvas that is predetermined in the execution is the city canvas which shows us an image and description of the site on which we are.
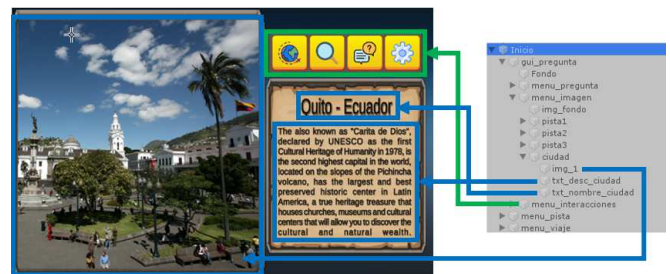

Fig. 13 Canvas Image Estructure

4) Settings Menu: In this menu the user will find the configuration of most of the components such as the Interface Reader, Colorblindness filters, Volume controls, Mute music, brightness control, contrast and Letter resizing.

911

Fig. 14 Settings Menu

M. Coding

Next, we present the more important scripts used in the video game.

1) GameController script: In this menu the user will find the configuration of most of the components such as the Interface Reader, Colorblindness filters, Volume controls, Mute music, brightness control, contrast and Letter resizing.

2) Scene Control Script: For the control of the scenes, parameters that allow to know in which state this object is managed. There are 3 three types of main objects that were used for the creation of each screen: Canvas, Panels and Empty Objects which have different properties will be exploited in order to design a friendly and intuitive interface.

3) Menu Manager Script: Within the handling of objects, Unity allows to assign the object to the variable so that you can control it easily and easily. The first thing to do is create the variables for Unity to display in its editor. It is necessary to consider the type of variable that is used when declaring since each of the variables has different properties, and know how to handle the definitions of the variables. Since these are private variables they can only be used in this class, but when being public variables just like other components have properties that are editable in the inspector. The developer can allow the variables to be edited from the inspector as well. When creating a script, the developer can add a new component to the Game Objects, in order to give a specific behavior to the Objects shown.

4) Questions Script: In this Script the logic of the game is handled, during the navigation between the screens of tracks and trips, this Script handles behavior and the modification of the variables declared in the GameController class since it handles the parameters of the Game, such as the name of the player, hours, score and the Booleans used for the behavior of each of the screens.

III. RESULT AND DISCUSSION

In order to prove how intuitive, the video game is, the developer team applies a concept test on each of its components. Moreover, system and component experiments were tested to find out how these plugins interact together. A user acceptance and usability trial were carried out a 10-person sample study. The test subjects were filled out a survey, before, during and after the tests. Then, two types of tests were performed. People who tried the game were first blindfolded and later they could play the game again without blindfolds.

Fig. 15 Post Processing Image applied

Some questions were answered before starting the game, in order to discover what people know about accessibility and if they had previously interacted with a game that gives them these features. Nine of ten interviewed subjects answered that they have not interacted with any video game that has the accessibility features. The first test of the functionalities of the video game was performed with the test subjects that were blindfolded as it can be seen in Figure 15. Execution times of each test were recorded in order to register the ability to adapt to interface reader characteristics. Resulting in an average of 33 minutes and 36 seconds.
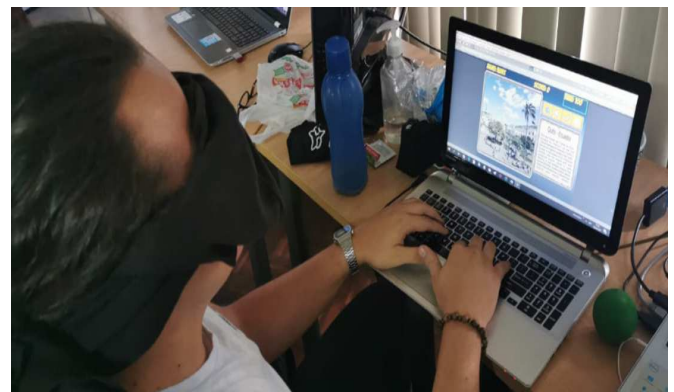

Fig. 16 Disability simulation test

This is an acceptable average execution time considering that they had to look for clues in the different places and travel to the cities to be able to finish the game. This was thrown back in the survey since 70% of the people surveyed answered, that they found it a little difficult and the remaining 30% that they did not find it difficult at all to play it. An important finding indicates that only 10% of the people surveyed indicated that they had played a game with accessibility features previously. This represents an important factor of analysis considering that it is a very low value.

IV. CONCLUSION

Accessibility in video games is starting to gain followers and more and more initiatives are being developed that allow people with disabilities to participate in more elements of society. Accessibility is not a common implemented functionality in the video game development industry, because of it is not a main characteristic at develop time. Only themes with large trajectory and a big community perform these options in order to engage more and more players.

UAP is a strong tool for accessible video games development, due to the possibility to apply it on several platforms like Android and iOS by the usage of TalkBack and VoiceOver. Because of its multi-platform characteristics, is highly recommended and should be one of the principal considerations now to develop an accessible game. Based on results it could be concluded that common players have never tried accessibility functions on a game before, also there is not strong knowledge about what accessible games means. In addition, it is deductible that the developed game does not show any difficulty to tested players, the solution for this event is: a better project planning and level designing.

It should be emphasized that entertainment as an education approach it is totally wasted. Nowadays people still thinking that video games are only a distraction or are catalogued just as a funny way to spending time. Causing the lack of use for all the possible educational uses they may have. After all, mixing fun with apprenticeship is a better way to learn. Accessibility should be one of the most important areas for developers to add into their technologies and applications in order to improve and provide an easily daily life for all gamers. In other words, it shall be humanity's duty to research and develop prototypes that benefit the interaction between humans and technology.

REFERENCES

[1] Spring D., "Gaming history: Computer and video games as historical scholarship," Rethinking History, vol. 19, no. 2. Routledge, pp. 207–221, 03-Apr-2015.

[2] Newzoo, (2018), Global Mobile Market Report Free. 1–25. [Online] Available: https://cdn2.hubspot.net/hubfs/700740/Reports/Newzoo_2018_Global_Games_Market_Report_Light.pdf%0Ahttps://resources.newzoo.com/hubfs/Re

[3] Schell J., The art of game design: A book of lenses, 3rd edition. CRC Press, 2019.

[4] Crompton H., Y.-C. Lin, D. Burke, and A. Block, "Mobile Digital Games as an Educational Tool in K-12 Schools," 2018, pp. 3–17.

[5] Statista, (2019) Serious games revenues worldwide 2024 [Online]. Available: https://www.statista.com/statistics/733616/game-based-learning-industry-revenue-world/

[6] Lindstrom L. and Jeffries R., "Extreme programming and agile software development methodologies," Inf. Syst. Manag., vol. 21, no. 3, pp. 41–52, 2004.

[7] Michael D. & Chen S. Serious games: games that educate, train and inform. Thomson Course Technology. 2006

[8] Aguado-Delgado J., Gutiérrez-Martínez J. M., Hilera J. R., de-Marcos L., and Otón S., Accessibility in video games: a systematic review, Univers. Access Inf. Soc., 2018.

[9] Jaramillo-Alcázar, A., & Luján-Mora, S. Mobile serious games: An accessibility assessment for people with visual impairments. International Conference Technological Ecosystems for Enhancing Multiculturality (TEEM), 2017, Article No.: 66

[10] Jaramillo-Alcázar, A., & Luján-Mora, S. An approach to mobile serious games accessibility assessment for people with hearing impairments, International Conference on Information Technology & Systems (ICITS), 2018, p. 552-562

[11] Jaramillo-Alcázar, A., & Luján-Mora, S. Accessibility Assessment of Mobile Serious Games for People with Cognitive Impairments, 2nd International Conference on Information Systems and Computer Science (INCISCOS), 2017, p. 323 – 328

[12] Jaramillo-Alcázar, A.; Salvador-Ullauri L.; Luján-Mora, S., A Mobile Serious Games Assessment Tool for People with Motor Impairments, 9th International Conference on Education Technology and Computers (ICETC), 2017, p. 172-177.

[13] Firth A., Low Vision and Colour Blindness, in Practical Web Inclusion and Accessibility, Berkeley, CA: Apress, 2019, pp. 55–92.

[14] Wasserman, D., Philosophical Issues in the Definition and Social Response to Disability, in Disability and Equality Law, Routledge, 2018, pp. 19–52.

[15] Cheiran, J. F. P. Nedel, L. and Pimenta, M. S., Inclusive games: A multimodal experience for blind players, in Brazilian Symposium on Games and Digital Entertainment, SBGAMES, 2011, pp. 164–172

[16] World Health Organization, (2018), Blindness and vision impairment. [Online] Available: https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment

[17] Color-blindness. (2018) [Online] Available: https://www.color-blindness.com/tritanopia-blue-yellow-color-blindness/

[18] World Health Organization, (2014), The global burden of disabling hearing impairment: a call to action, [Online] Available: https://www.who.int/bulletin/volumes/92/5/13-128728-ab/en/

[19] Yuan J., Sun Y., Sang S., Pham J. H., and Kong W. J., The risk of cognitive impairment associated with hearing function in older adults: A pooled analysis of data from eleven studies, Sci. Rep., vol. 8, no. 1, Dec. 2018

[20] World Health Organization, (2019), Deafness and hearing loss. [Online] Available: https://www.who.int/en/news-room/fact-sheets/detail/deafness-and-hearing-loss

[21] Abrahamsson P., Salo O., Ronkainen J., and Warsta J., Agile Software Development Methods: Review and Analysis, Sep. 2017.

[22] Parikh D. D. V. S., Extreme Programming: Values, Principles, Practices and UsE, J. Curr. Sci., vol. 20, no. 1, Jan. 2019.

[23] Unity, (2018), MonoBehaviour.Start() [Online] Available: https://docs.unity3d.com/es/current/ScriptReference/MonoBehaviour.Start.html

[24] Project Wilberforce. (2017), Colorblind Effect Unity Plugin User Guide. [Online] Available: https://projectwilberforce.github.io/colorblind/
Unity, (2019), Post-processing overview [Online] Available: https://docs.unity3d.com/Manual/PostProcessingOverview.html