TÉCNICO LISBOA

Treme-Treme 2.0 - A serious game to teach children earthquake preparedness

Duarte Nuno Gomes Botelho

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Rui Filipe Fernandes Prada
Dra. Mónica Maria Lopes de Sequeira Amaral Ferreira

Examination Committee

Chairperson: Prof. Miguel Nuno Dias Alves Pupo Correia
Supervisor: Prof. Rui Filipe Fernandes Prada
Member of the Committee: Prof. Carlos António Roque Martinho

June 2019

Acknowledgments

On the first page is only my name as the person who wrote this report to obtain a master's degree, but there should be many other names. Luckily there is this section so I would like to announce my thanks.

I would like to begin by thanking my parents for having made who I am today and for all the support they have given me over the years, through ups and downs. They were the key to enable me today to deliver this thesis and thus complete my master's degree. Without them, I would be nothing. Words are not enough to express my feeling for you. Thank you both so much.

I would also like to do it to Prof. Rui Prada and Mónica Ferreira for having chosen me for the project and for their guidance and support throughout the last year.

A special thank to Sara for being beside me through so many moments and for making me a better person since the day I met you. Oh, and for helping me with the translation of the game into French. Luckily I met the right polyglot :)

All my family, especially my grandmother that raised me so well, was also important to achieve this goal. Day by day, every single one of you was like a piece on a giant puzzle that made me what I am.

Lastly, but nonetheless, I would like to thank all my friends who have supported me whenever I needed them. All the group work, all the hours playing, all the rage, all the walks, all the trips, nothing would be the same without you guys. In particular, I would like to thank Pandaria, Noobs and Aptoiders and the Blockchain Team for their support.

I apologize for not specifying all other people who helped me deliver this report, but you know who you are and you will be forever in my heart for everything you've done. Whenever you need me, I'll be there count on me.

Thank you all,

Duarte

Brigadinhos pessoal! ;)

Abstract

Earthquakes continue to be one of the most destructive natural disasters. Even with the technological advances that have been made, it is impossible to predict when the next earthquake will occur or what magnitude it will have, so it is critical that everyone is prepared for an earthquake. In 2014 a serious game was developed to teach people, particularly primary school children, which behaviours should be adopted before, during and after an earthquake. This game, Treme-Treme, had a huge success and 5 years later still continues to be used by elementary school teachers as a complement to teaching. In addition, it was also adopted by schools of children with educational difficulties, having a greater impact than initially thought. The opportunity has now arisen to continue the development of this game, and this work portrays the whole process inherent to the realisation of the new version. Over the years, Treme-Treme's code has become obsolete, so the game has been successfully rebuilt from scratch and improved, using a new platform, Godot. Game design was rethought and successfully implemented to provide an improved player experience. The results got with the tests show that Treme-Treme's new solution provides better engagement with children than the previous one, transmitting the intended pedagogical knowledge more effectively and representing a better tool for teachers.

Keywords

Serious game, Earthquake, Tsunamis, Preparedness, Children, Godot Engine

Resumo

Os terramotos continuam a ser um dos desastres naturais que maior destruição criam. Mesmo com os avanços tecnológicos que têm sido feitos, é impossível prever quando ocorrerá o próximo terramoto ou qual a magnitude que terá, pelo que é fundamental que toda a gente esteja preparada na eventualidade de ocorrer um terramoto. Para tal, em 2014 um jogo sério foi desenvolvido para poder ensinar as pessoas, nomeadamente crianças do ensino primário, sobre quais os comportamentos que deveriam ser adoptados antes, durante e após a ocorrência de um sismo. Este jogo, o Treme-Treme, teve um enorme sucesso e 5 anos depois ainda continua a ser usado por professores de escolas primárias como um complemento ao ensino. Para além disso, foi igualmente adoptado por escolas de crianças com dificuldades educativas, tendo um impacto maior que o inicialmente pensado. Surgiu agora a oportunidade de continuar o desenvolvimento deste jogo, sendo que este trabalho retrata todo o processo inerente à realização da nova versão. Com o passar dos anos, o código do Treme-Treme tornou-se obsoleto, pelo que o jogo foi refeito e melhorado com sucesso, usando uma nova plataforma, Godot, o que permitiu repensar todo o design do jogo para proporcionar uma experiência melhor ao jogador. Os resultados obtidos com os testes efectuados comprovam que a nova versão do jogo desperta o interesse das crianças de uma forma mais eficaz que a anterior, fazendo-se sentir a melhoria no jogo ao transmitir ainda melhor os conhecimentos pedagógicos pretendidos e representando uma melhor ferramenta ao ensino.

Palavras Chave

Jogo sério, Terramotos, Tsunamis, Preparação, Crianças, Godot Engine

Contents

List of Figures

Listings

Acronyms

API	Application Program Interface
CSV	Comma-Separated Values
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
GLES	OpenGL for Embedded Systems
VR	Virtual Reality
XML	Extensible Markup Language

1

Introduction

Contents

1.1 Motivation

Natural disasters are something common on Earth, differing only in the type of disaster depending on the area in which we are and the predictability of them. Scientists are able, for example, to identify the formation of a tornado as well as its trajectory and intensity with hours or even days in advance, there being specific times when the probability of a being generated is greater. In the case of earthquakes, it is still impossible to predict in good time when one will be triggered. The tectonic plates are in constant movement and can release the energy accumulated slowly, in the form of small earthquakes imperceptible to the population, something that happens daily, or abruptly, with earthquakes that cause mass destruction. There have been some advances in the predictability of an earthquake, especially in large ones, using statistical models and economics to predict earthquakes in the long, intermediate and short periods, but this is a problem that has been tried to be solved for over 100 years without an ideal solution [1–3].

Therefore, it is important for us all to be aware of the danger to which we are subjected by living in an area with seismic activity. It was in this philosophy that in 2014 the serious game Treme-Treme was developed [4]. It was born from a project on seismic preparedness of the Department of Civil Engineering of Higher Technical Institute (DCE) under the European project UPStrat-MAFA (Urban disaster Prevention Strategies using MAroseroseic Fields and FAult Sources). The department wanted to create a serious game that would teach children about 8-10 years of age the dangers of earthquakes, as well as teach the best practices to take before, during and after an earthquake. As the target audience was very young, the complex knowledge should be conveyed in a simple way, which they understood and retained. The use of serious games has proven beneficial for the transmission of complex knowledge over short periods of time when compared to traditional teaching methods. These types of games have been applied in several areas, both children and adults, with very positive results. [5–9]

Treme-Treme has been used several schools by elementary school teachers in, at least in Portugal, as another classroom teaching aid tool. Several sensitization sessions have also been held by the civil department in schools and events, such as the exhibition ”Quando Lisboa Treme. De 1755 à Cidade Resiliente” Palácio Pimenta in 2015, at the European Researchers’ Night in 2016 and at the Avante Festival in 2018. In addition, it is also known that the game has been applied in schools for children with special needs, something that had not been foreseen initially, but which portrays well how good this tool. In sum, there are already hundreds of children who have played the Treme-Treme and are now better protected against earthquakes, so it is fundamental that the game is maintained and continued so that many more have the opportunity to experience it as well.

The game was initially developed using Unity, a free engine (up to a certain volume of business) still very popular today, which allows to make games in 2D and 3D, as well as export to various platforms, used by major manufacturers of video games such as CD Projekt, for example. The first version of the

game was exported to 3 platforms, web, PC and macOS, all available for free and was the first integrated into a website specially dedicated to this game. Anyone could simply access http://www.treme-treme.pt/ and play directly on the browser, without having to download. However, at that time games exported to the web using Unity necessitated a special plugin, the Unity Webplayer, which became deprecated and no longer supported by browsers a few years ago. Therefore, the game is no longer available to play. In addition to this plugin, the source code itself is no longer available for reuse because Unity has evolved a lot in the last 5 years.

With the need to keep the game available to the public as another tool to raise awareness about the dangers of earthquakes and how we should prepare, this thesis proposes a new solution, rebuilt from the beginning and with several changes, to re-launch Treme- Treme for free and on more platforms than before, and is also available in a mobile version, as it is a market that continues to expand and already has more players than any other platform.

Having said this, it was established that the game should convey the following messages:

- It is essential to build an emergency kit containing only the items necessary for the survival of one or more persons for an indefinite period of time.
- The player must know what dangers to avoid in the house and what to do in any situation
- The player must know that it is essential to disconnect both the light meter and the gas valve before leaving the room
- Children should be taught the importance of leaving home and going to a meeting point previously defined with their relatives because this is a point of union between all
- It should be shown how dangerous the earthquakes are and how long a person takes to protect themselves

1.2 Project Goals

The main goal is to allow everyone, specially our target audience, to easily play Treme-Treme again. The game should be working on a browser, without having to download additional plugins. In addition, the game should further promote engagement with children and at the same time maintain the transmission of knowledge about the correct practices and procedures before, during and after earthquakes occur. The proposed improvements suggested in the previous work should be evaluated and, if suitable, implemented. The game should also be available on more platforms than previously supported, allowing the largest number of players to experience the game, regardless of the device they own. New languages should be added to allow players from different nationalities to play the game and be prepared for an earthquake.

1.3 Organization of the Document

This thesis is divided into 6 main parts. Chapter 1 presents a motivation for the work carried out, explaining why it is a necessity, accompanied by the definition of the fundamental objectives of the project. Chapter 2 begins by defining what serious games are because they are important, then the three major stages of earthquakes are analyzed, and finally, some related works are introduced, as well as a comparison between them. In Chapter 3 an overview of the first version of the Treme-Treme game is made, with an explanation of the game concept followed by the problems that were solved with the new solution. Chapter 4 details the whole new solution, starting with the explanation of choosing a new game engine, followed by the changes made, first in high level and then at a low level. In Chapter 5 an evaluation of the work is done, comparing both versions based on the results of practical experience in a school. The methods used are explained, as well as the procedure and the results, ending with a brief summary of what was obtained. Finally, in Chapter 6 a conclusion is made on the whole work and some proposals for future work are presented.

2

Related Work

Contents

This chapter contains work related to this project. It starts by defining what is a serious game and how players can learn while playing. Then it will briefly explain what is an earthquake. Finally some serious games focused on earthquakes will be presented and compared between each other.

2.1 Serious Game

The term *serious game* is a buzzword and there is no consensus on its definition. There are several definitions given to [10] has selected two main definitions from [11] and [12], highlighting six key characteristics of a serious game.

- A serious game is a learning process and teaching is a priority
- It should still entertainment the player because it continues to be a game
- It is an application that use video game technologies
- Multiple objectives should be address
- Almost all fields are covered, from business and defense to education
- Everyone can play it, no matter the age

Given this, we can define the term *serious game* as a video game, that can be played by every individual, independent of its age, and that has the main purpose of teaching one or multiple subjects, from a wide range of fields, while the player is entertained. However, it should not be played just for amusement, it is a pedagogical tool and should be treated as such.

2.2 Earthquake Preparedness

Natural disasters occur more and more frequently and sometimes leaving large tracks of destruction. One of these disasters is earthquakes, a release of energy at the level of the earth's crust caused by the sudden slip of the tectonic plates. The friction causes the edges of the plates to get stuck together, and as they move steadily, sometimes the energy is released quickly, which causes waves of energy to reach the surface where they do damage. If the earthquake happens on land, the damage to buildings can be extensive, with the formation of avalanches, new mountains or the opening of faults, such as the San Andreas Fault, which has more than 1046 kilometres in length and 16 kilometres in depth. On the other hand, if the earthquake happens under water, tsunamis can form, walls of water that move in all directions and epicentre the site of the earthquake. These walls can be tens of meters high when

the energy released is large, thus causing even more damage to the coastal populations to add to the earthquake occurred.

The damage does not stop there, taking, for example, the earthquake of magnitude 8.7 that hit Japan, where a tsunami struck the Fukushima nuclear power plant, causing the release of radioactive material into the atmosphere and contaminating rivers and soils to the return. There has been a reduction on marine products in Fukushima [13] and evidence has been found that wild monkeys in Japan [14]. More than 300,000 people were evacuated and no death was recorded from direct exposure to radiation, but it is estimated that the radiation released was so much that they arrived to be reported on radioactive material in California [15].

Seismic risk is a reality in Portugal, but most of the population is not aware of this problem. In Portugal, there is a great shortage of pedagogical tools for seismic risk education. Treme-Treme was designed to raise awareness among young people of changing attitudes and behaviors related to protection and prevention. The more informed we are about how to lessen or avoid damage, the better prepared we will be in case one day the earth shakes.

Since earthquakes can not be prevented or predicted, it is with initiatives like this that one can live better with the hypothesis that they occur, and one of the main objectives of the investigation is also fulfilled: the connection with society.

Having said that, it is very important that each of us know what to do before, during and after an earthquake. To this end, with the help of the Department of Civil Engineering of Instituto Superior Técnico, as well as information available online from UPSeis [16] and from institutions such as the Instituto Português do Mar e da Atmosfera[1], American Red Cross[2] and Ready[3], it was possible to list the main measures to be taken in three possible moments.

2.2.1 Before an earthquake

Preparation is the key to success. So it is at this stage that each of us must invest more time to prepare ourselves mentally as best we can for the disaster that may occur. There should be a concern to build an emergency kit with at least the following items:

- Water bottles (at least 1 litre/day/person, minimum, for at least 3 days)
- Canned food, easy to prepare for at least 3 days/person
- Medication for at least 7 days, specially if the person suffers from a chronic disease
- At least one battery-powered or hand-crank radio

[1] https://www.ipma.pt/pt/enciclopedia/geofisica/caso.sismo/index.html
[2] https://www.redcross.org/get-help/how-to-prepare-for-emergencies/types-of-emergencies/earthquake.html
[3] https://www.ready.gov/earthquakes

- Batteries
- Multi-purpose tool with a sharp knife with a can opener
- Sanitation and essential personal hygiene items
- Flashlight
- Matches in a waterproof container
- Whistle
- Map
- One change of clothes
- Signal flare
- Contacts of the emergency services (police, fire, nearest hospital) and family
- Personal identification and important documents, or at least a water proof copy thereof
- Credit card and cash
- Games and activities for children

This kit should be constantly updated and be on a location easily accessible by all family members. In addition, there must also be a ready-to-use extinguisher, the ways of switching off gas and electricity must be known, furniture should be secured to the floor and walls, especially the heavier ones and the building should be maintained and repaired when cracks arise. This applies to both home and workplace. A meeting place should also be set so that all family members can come together. The area should be well known to all in order to be able to guide themselves and make the best survival decisions, such as going up to high ground in the case of coastal areas. Finally, it is always essential to carry out simulations whenever possible to test the reaction capacity of each in an environment closer to what would be experienced in reality and to test if the preparation measures taken are the best, adjusting what is not correct.

2.2.2 During an earthquake

Drop, cover and hold on. This is more important during an earthquake and one should always remain calm. If the person is inside a building, it must look for shelter under a strong surface, that can withstand eventual objects that can fall. A table or a bed is often the best place to find shelter, and you should always protect yourself from glass shards from mirrors or windows. Only in extreme cases, if none of the

above exists, the person can shelter under a doorway, but this remains an extremely dangerous place to be. If gas is smelled, one should get out of the house and move as far away as possible.

In case of being on the street, it should take refuge in open places, without buildings, roads or poles nearby, since there is a greater risk of something heavy falling. If you are driving, the car must be stopped safely in an open and wide area. There is always the possibility of opening craters on the ground, so the person should take this into account.

2.2.3 After an earthquake

After the earthquake is over, you should leave home as soon as possible and always be safe, as the building has been weakened and there is a possibility of collapse. On the way, and only if it is safe, you should go get the emergency kit. Replicas will emerge, these will be less intense than the main earthquake but can be equally destructive, so it must be kept in a safe place when they occur. You should go to the place previously identified as a meeting point to meet with family members. If you live in a coastal area, you should go to the highest place possible, since there is a possibility of tsunami formation. The directions of the authorities should be followed. There is a possibility of a region being isolated for a few days, so you should always have the emergency kit with us. If you are sailing on the high seas, you should pay close attention to tsunamis, especially if the boat is small.

2.3 Other Serious Games

We will now describe some serious earthquake games. In 2014 there were also other games that have since ceased to be available online, such as Treme-Treme, so they will not be covered in this document. Others, like Section 2.3.1, have changed their location online, something that has been updated here to let the reader play it and learn more about the topic.

2.3.1 Beat the Quake!

Beat the Quake![4] is a serious game that teaches players how to prepare their homes for the occurrence of an earthquake, more specifically, how to hold different objects on solid surfaces so they are not knocked over. It was designed in 2006 by the Earthquake Country Alliance, a partnership of people, organizations, and regional alliances that work together to improve preparedness, mitigation and resiliency. The last available version is 2008 and the game has a scenario as a living room and the player has to find out which objects need to be fixed by clicking on them. A window is then asked to ask the players which methods and tools are most appropriate for fixing the clicked object, giving them three

[4] https://www.shakeout.org/dropcoverholdon/beatthequake/game/

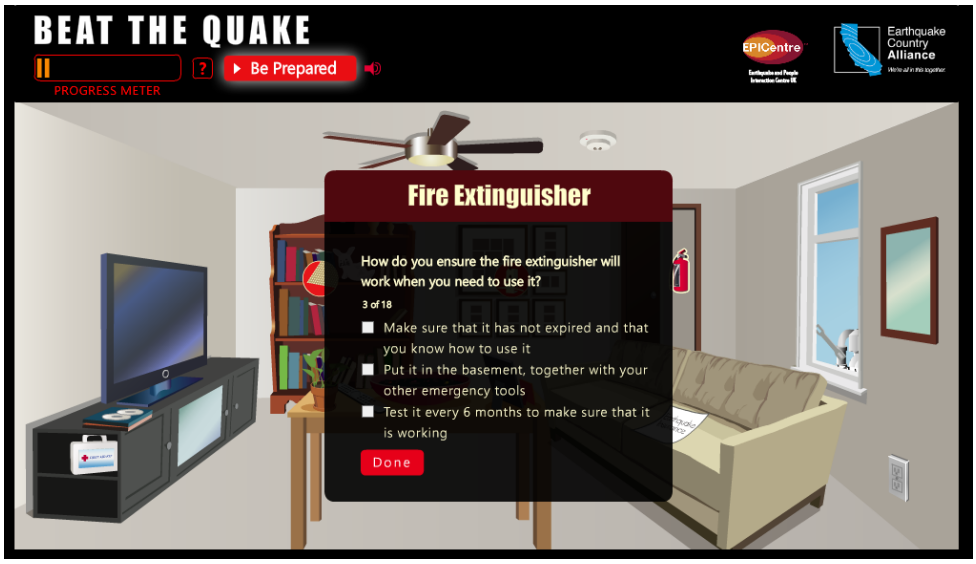


Figure 2.1: Beat the Quake!

options, as you can see in the Figure 2.1. In total there are 18 objects that must be identified and chosen before the earthquake occurs, with a progress bar in the upper left corner. The level of interaction of the player is measured and, if the player takes a long time on a given question, the earthquake happens more quickly and the opposite applies if the player is quick to answer the questions. The game also encourages the player to learn more about the theme, with a button, "Be prepared", which puts the game in pause mode. In this mode are shown preparedness tips that help answer questions about each of the objects, when the player clicks on them, and that works as a complement to learning. During the game, there is an auditory, visual and textual interaction to keep the player engaged and to instruct him. At the end of the game, the quake happens and the level of destruction of the living room is higher or lower depending on the player's choice. A score is given according to the player's performance, all objects are shown, both those that have been identified and those that are not. In addition, more resources are given on sites with content on the topic and it is possible to play again or send the score obtained to friends and invite them to play, requesting the name and email of the player, as well as that of up to 5 friends, being sent later an email in the name of the player for each of them.

This is a game very focused on the main objective, with simple mechanics and quite rich in content. The player has the freedom to choose the available objects in the order he wants, to make the decision he thinks best and to pause the game to learn more about them. The fact that the earthquake happens more quickly if the player takes more time to make his decisions passes the message that earthquakes can happen at any time and we must prepare as soon as possible. It is only available for the web, which is by itself not completely negative as it can be played on any computer provided there is an internet connection. Finally, the use of Flash in development will be quite negative from 2020, the year in which this technology will be discontinued, no longer supported by browsers.

2.3.2 Earthquake Simulator VR

Figure 2.2: Earthquake Simulator VR

Earthquake Simulator VR[5] is Virtual Reality (VR) serious game that puts players inside a house during an earthquake, followed by a fire. It was developed for HTC Vive and has as main objective to teach survival techniques that must be taken before and during the catastrophe. The game begins inside the room of a house, where it is possible to access all the divisions. The game's basic instructions are written on a wall, just like the button to start the game. By clicking on it, the television begins to give a realistic warning of the eminence of a catastrophe and, when leaving this division, the player moves to the kitchen, where is a dog that will later guide the player during the rest game, showing for example where one should take refuge during the earthquake, and a series of objects on the bench, as well as a white container with red crosses on the sides and a frame with a checklist on the wall. The first objective of the player is to place the items inside the container and scrape them from the checklist. Then it is shown to where the player should follow, by order arrows on the ground and the behaviour of the dog, which shows that the player should shelter under the dining room table. When the player is able to shelter, the earthquake begins, recreating what would happen in reality. The earthquake takes several seconds to end and around the player, all the items not attached to the walls start to fall. After the earthquake, the player must try to leave the house, an objective that is made difficult by the beginning of a fire inside the house.

This is a game that can recreate extremely well the reality and captivates the player very well, because it is played in VR, transmitting to the player each of the events. It is only available for PC, needing

[5] https://store.steampowered.com/app/607590/Earthquake_Simulator_VR/

the HTC Vive headset, a VR device with two screens that give different images and that is placed on the head, transmitting the sensation of 3 dimensions, and that detects the movements of the people, being accompanied by two commands, used to pick up kit items. This is a concept that makes sense in these games because it enables the player to be more involved with the game and feel more of their decisions and understand what cost they may have. As a negative, we have the fact that not everyone has available a headset like Live and the game is not free, so not reaching the target audience like other games.

2.3.3 Earthquake Survival

Figure 2.3: Earthquake Survival

Earthquake Survival[6] is a simulation/survival 2D serious game. The story takes place in Kapiti Coast, New Zealand, where the player lives with his family when, unexpectedly, at 8 AM a magnitude 7.0 earthquake occurs and causes widespread damage, causing isolation of the city for at least 3 days. The aim of the player is to help both his family and the community survive until external aid arrives. For this, the player must exchange supplies that he has with the community, as well as help them in some tasks, such as building bathrooms or cleaning debris. In the image, you can see the only game scenario. At the top, we have the six player stats, which decrease over time and increase when consuming the supplies. Still on the top and centre is the number of days after the earthquake, as well as the current time, which only increases in two ways: one hour while helping someone or three while sleeping. Below, we have the list of supplies as well as the indication of the quantity of each. Finally to the centre, we have the area controlled by the player. In the background, we have the player's home lit, from where

[6] https://tough-love-productions.itch.io/earthquake-survival

notifications appear that one of the player stats is low and foreground the various bots that circulate horizontally with notification icons indicating that it is possible to interact with them. To do this, the player must hover with the mouse on the bots to know if it is possible to help them and later click on them to confirm the decision. The game ends when help arrives or when the player dies.

This game was developed in GameMaker, a drag&drop style engine, which has a free component that allows exporting to PC and a paid that allows exporting to the remaining platforms. As for this particular game, it has been exported to the web and PC, can be played for free and remains just a prototype. It is played just by clicking on objects and is very static and the player has to control the resources available to keep itself alive as time goes by. It teaches that help may take some time to arrive and that resources should be shared and well managed with individuals around the player. On the other hand, there's not much graphics showing that a massive earthquake just strikes the city because all the buildings are still intact.

2.3.4 Disaster Hero

Figure 2.4: Disaster Hero

Disaster Hero[7] is a fairly complete and extensive game that teaches some practices that must be taken before, during and after a natural disaster to the player. In the beginning, the player can choose and personalize his character between two characters, then enter a television contest with the slogan

[7] http://www.disasterhero.com/

”Are you prepared to be the next Disaster Hero?”. In this, the player can choose to fight against one of four different bots: Tectonic, Tempest, Whirwind and Dr Deluge, which correspond to earthquakes, tornadoes, hurricanes, and floods, respectively. After this choice, the player and the bot play a series of mini-games to decide who the next disaster hero is, many of them with questions that reward the player after a given choice. The themes of these mini-games are related to what the opponent’s bot represents and, in the case of the earthquake, includes drawing up a plan and an emergency kit.

This is a simple game, clicker type, which teaches many concepts to the player through the questions asked. It is only available for web and was developed by the American College of Emergency Physicians, an organization founded in 1968 with the objective of educating and training physicians in emergency medicine to provide quality emergency care in the nation’s hospitals.

2.4 Other Serious Games Comparison

We will now compare the previous games.

They all have something common, they are serious games that have as the main objective to teach the player procedures to take in case of an earthquake. Each transmits this knowledge in his own way. Beat the Quake! focuses mainly on pre-earthquake, while Earthquake Survival focuses more on the post-earthquake. The remaining two cover all phases of an earthquake.

As for building an emergency kit, Earthquake Simulator VR is the only one that directly addresses the issue. We can consider that Earthquake Survival refers indirectly to the need to have a good emergency kit since the player has an inventory that he has to control in order not to die. The need to leave home is also addressed directly by Earthquake Simulator VR.

As for the interface, the poorest is Earthquake Survival, since the action takes place after a quake of great magnitude and the houses shown are completely intact. Also, as the action takes place on the street, it would be thought that one could walk a lot for a map, but we are confined to a single static screen. Beat the Quake !, is a similar game in this point of view, however, the player has the feedback that an earthquake occurs because the room shakes and the objects not secure to the wall and floor fall.

Finally, the game that has more realistic gameplay is without a doubt Earthquake Simulator VR, because to be able to make the player feel completely immersed in the event. Disaster Hero is in a separate category, as it addresses the issue of natural disasters, including earthquakes, in an indirect way. The player competes against the enemy ”earthquake” and tries to overcome it through minigames. This is very peculiar gameplay and quite well elaborated.

When to the covered platforms, Earthquake Simulator VR is only available for PC, while the remaining ones are available for the web. Earthquake Survival is the only multiplatform of the four, being available for both web and PC.

3

Treme-Treme v1

Contents

This chapter will point out the work done on Treme-Treme prior to this thesis and why was important to update it.

3.1 Previous Gameplay Design

The Treme-Treme is a serious game built around the theme of earthquakes and that takes the player to go through all the phases of an earthquake. It was developed for a master's thesis in 2014 by a master's degree student in computer engineering from Instituto Superior Técnico, in partnership with the Civil Department of the same faculty and under a European project. This simple 2D game, single player, intended to convey complex themes to children between the ages of 7 and 10, especially in the classroom context, where a teacher could put their students to play to complement the theme taught during a lesson.

The game has three levels conceptualized, and only the first two were developed in a production version. As it fits in a European context, the game supported three languages, Portuguese, English and Italian, which could be changed in the settings from the main menu. On almost every screen a pause menu is available, which allows you to turn the sound on or off or exit the game.

Figure 3.1: Treme-Treme v1 Kit Mission

The first level corresponds only to the preparation phase before an earthquake. It is first shown an explanation over several screens about what is an earthquake, how the player should prepare for one and what dangers are subject. After this phase, the player can choose one of two characters, the boy, Sunami, or the girl, Terramota, who will represent the player ahead. Lastly, this level ends with the player's first major mission, building an emergency kit that will be used after an earthquake. In this mission, as you can see in the Figure 3.1, the player must drag to the emergency kit the six items that he

found fundamental for his post-earthquake survival of nine possible choices and submit his kit to verify that everything is correct by clicking the button Okay, whenever you want. If the player does not make the right choice or does not fill the entire kit, a message is displayed regarding the problem.

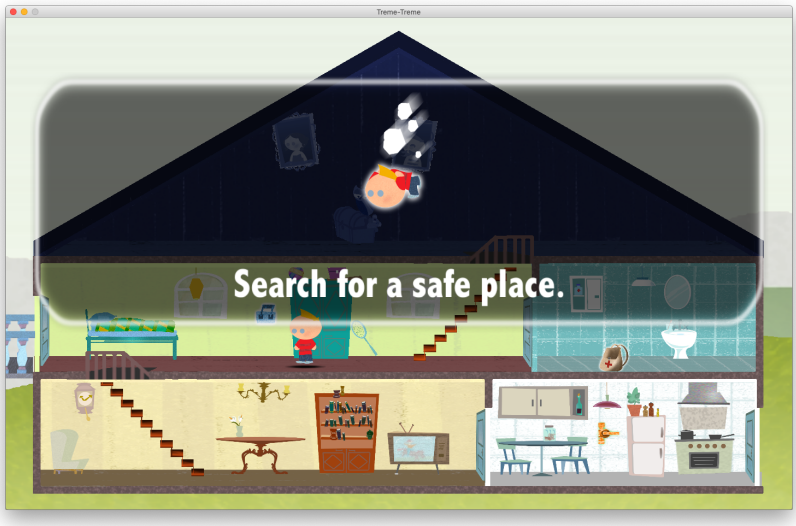


Figure 3.2: Treme-Treme v1 House Mission

The second level corresponds to all phases of an earthquake, before during and after. In this, the player begins by spawning in a random place of his house, and are given clues to the player about what will happen, as can be seen in Figure 3.2. An earthquake is about to happen and in just 10 seconds a safe place to take refuge in safety must be found. If you can not or if you decide to wander around the house during the earthquake, you will be subject to any of the dangers such as glass, furniture, electric wires, lamps, ceilings or the balcony. After the main earthquake, an aftershock sequence appears between intervals of 10 to 20 seconds, forcing the player to find a new refuge. To penalize the player for staying too long in the house, the character dies during the third aftershock, if it is still in the house. The player can optionally pick up the emergency kit, not requiring it to complete the level, such as turning off the electrical panel, which is located near the bed. On the contrary, before leaving the house the player must turn off the gas, otherwise, he will die when he is near the escape. After leaving home, the player should look for their relatives who are waiting for him at the meeting point.

As for the third level, it was developed but was never release to the public, as has been said previously, but would continue to give continuity to the story. The player would be with his/her family in a coastal zone and a tsunami would be approaching. He/she would have to use the features of the emergency kit he/she built earlier to save everyone. This level is thought to have two components, one by day and one by night, to show the player that the flashlight and batteries are two indispensable resources. All gameplay would be spent on the street, outside of dwellings, representing the days after the earthquake,

thus showing the player what other care should be taken abroad.

3.2 Identified Problems

The most relevant problems from the first version of the game that have been identified will now be listed.

Game doesn't run on the website This is the most serious problem, which originated from the need to carry out this master's thesis. The game must be made available for free and as easily as possible to the player. Therefore, one of the first things the player must have available when he or she accesses the Treme-Treme site should be the game itself, with no need to install anything, especially third-party plugins. It turns out that the game was made in a version of Unity that still needed the Webplayer plugin to run in browsers. Today this plugin has been discontinued and current browsers no longer support it. One of the solutions would go back to generating an export to the web using HTML5, for example, but since Unity has already evolved a lot in the last 5 years, it is almost impossible to import the existing code for a recent version of Unity, since most used libraries are deprecated. In explaining scenes, if the player clicks too many times on the button for a short period of time, sometimes results in the unexpected crash of the game. It is therefore very important that this problem is solved.

Level 1 is too big The first level, which corresponds to the construction of the emergency kit, is too large since it also encompasses all the explanation of what an earthquake is and the character's choice. Therefore, this level could be divided into at least two parts to compartmentalize the topics.

Game progress not recorded If the player, for any reason, closes the game and goes back to running, all progress will be lost, having to start all over again. From a pedagogical point of view, this is not really a problem, because the player going through the whole explanation could assimilate concepts that in the first gameplay had not been so clear. From the point of view of the game itself, this is a problem as it adversely affects the player-game interaction.

Sound disabled by default Sound is something that completely turns a game. To do so, it must be on by default unless the player wants to turn it off.

English was the default language In the context of the European project, it is important that the game has English as default language. But since this is a Portuguese game, our language should become the main language when the game is first started or, if it is added, detect the language of the device and set it accordingly.

Game crash if forward scene button is pressed too fast In explaining scenes, if the player clicks too many times on the button for a short period of time, sometimes results in the unexpected crash of the game.

Italian was not fully translated Some strings were detected to be in English when they actually

belonged to the set of Italian strings, such as ”audio on” or ”audio off”

Third level still not implemented The third level, although already conceptually done and already having all the necessary assets, is still not playable.

Poor version control The code of the first part of the project was placed in Bitbucket, the web-based version control repository hosting service owned by Atlassian and alternative to Github. However, this repository only has 4 commits with a very descriptive message of what was done in each commit.

4

Treme-Treme v2

Contents

This chapter describes the work done in the updated version of Treme-Treme. It begins with an explanation of the tool used for the development, as well as a brief comparison with the one used in the previous version of the game. Then the biggest changes made to the game at the architectural level are described, followed by a more detailed explanation on the implementation of each level. Finally, it is detailed how the progress of the game has been saved, as well as the new system of translations used and the addition of a new language.

4.1 Choosing a new development tool

As stated in the last chapter, Treme-Treme's first version has been developed using version 3.x of Unity3D, which represented a major challenge to expand the previous work. Given this, there was the option to change the game engine to a better suited one for Treme-Treme. An alternative open source game development engine was considered, Godot [1]. Its documentation provides a quick writeup that goes as follows:

Godot Engine is a cross-platform game engine to create 2D and 3D games from a single interface. Games can easily be exported to multiple platforms, including the major desktop platforms (Linux, macOS, Windows) as well as mobile (Android, iOS) and web-based (HyperText Markup Language (HTML)5) platforms. It is free and open source and it will continue to be like that, driven and developed by the community.

Unity is a proprietary and free software with restrictions on both usage and revenue. The software is directly supported on Windows and macOS but not on Linux, supports development for all platforms, uses Visual Studio as an editor, contains endless resources available online ready to use, and the scene system works with GameObjects and Prefabs. Godot, on the other hand, is completely free, with no limits to the use or revenue generated, is directly supported on all operating systems, only not offering export to TV and Console when compared to Unity. Contains a powerful animation creator, supports the usage of shaders, with a dedicated editor, allows debugging directly on the device where it is intended to export and uses the system of scene nodes and trees instead of prefabs. Using this system, it is possible to divide a large scene into several smaller ones, re-using them whenever possible, as we will see later. An example of this system is a scene that contains a character, enemies and background. It is possible to create a scene with only the character and one with one or more enemies, thus dealing with all the details in these scenes. Then we can create the main scene with the background and its obstacles and link the scenes of the character and enemies with this, creating the level. In Unity, these subscenes would be the Prefabs, they would always have the same properties.

The editor is mostly divided into 6 parts, which can be seen in the Figure 4.1. On top, represented

[1] https://godotengine.org/

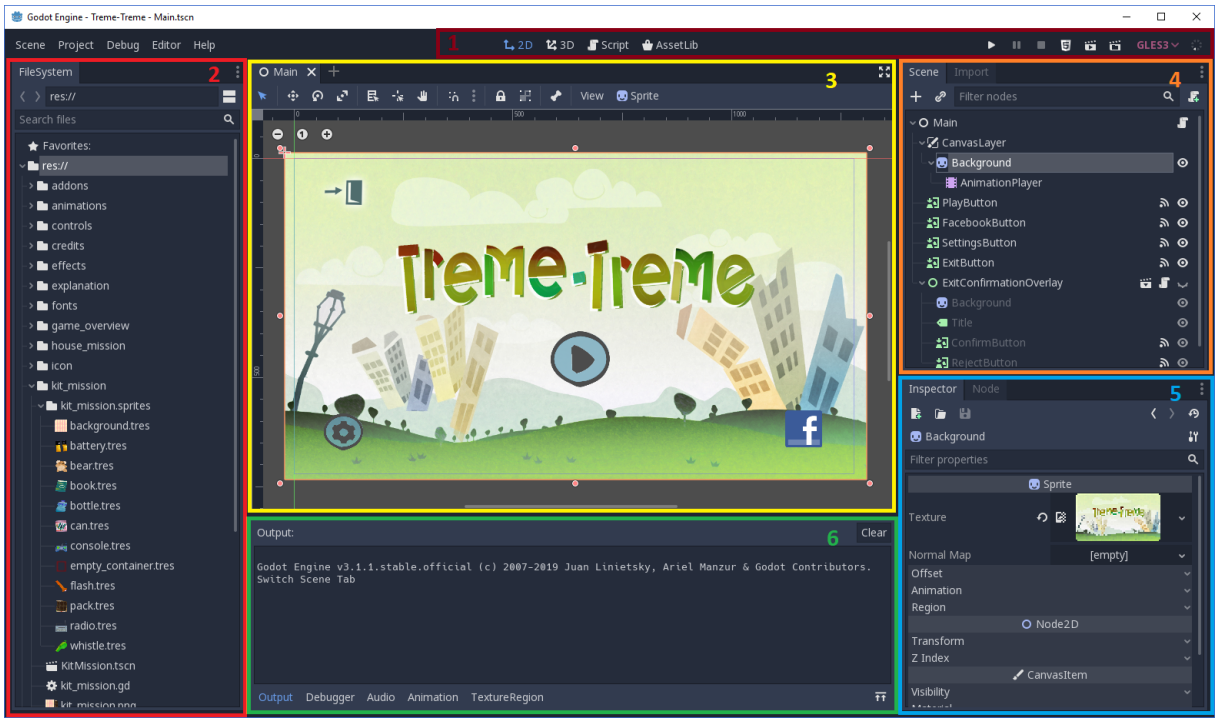


Figure 4.1: Godot engine's editor

by *1*, we have the toolbar, where you can move between the two types of scenes, 2D or 3D, as well as Scripts and AssetLib, where you can obtain assets available online, or integrations with other programs. For example, in the case of Treme-Treme, an addon was included to easily import atlases of images generated by the program TexturePacker. In addition to that described, the toolbar also includes buttons to run the game from the beginning in native format to the operating system, web and mobile, from the scene that is being shown or another customized, pause and stop the playback and, finally, to change between OpenGL for Embedded Systems (GLES)2 and GLES3, the two rendering modes used by Godot. Treme-Treme was developed using version 3 as it supports the particle system, used at level 2, but in doing so, some old hardware may not be supported.

Shown with *2*, on the left, is the file system, where all the project files are located, with the main folder being `res://`. In the case of Treme-Treme, the files were organized into folders first by levels and subdivided into scenes. Each subfolder contains all the textures that will be used in the respective scene. As for common files several scenes, such as global scripts of which we will speak the subsection, translations and settings sounds, are in specific folders, to be easily accessed by the system. In *3*, at the centre of the image, we have the Scene, which as the name implies, is where the scene is constructed with all nodes instantiated in the Scene Tree, represented in *4*. In *5*, the Inspector, we can change the properties of these objects, such as their size, colour, texture, etc. Finally, in *6* we have the output and debug windows, where you can test the game and find out problems. Godot offers a debug system with

breakpoints that are not ideal but allows you to know the basics about variables and functions called.

In short, the engine transition from Unity to Godot was due to the following major reasons:

- Lightweight, Godot's executable file has less than 40MB).
- Available on Steam, with "Overwhelmingly Positive" reviews, keeping me always updated with the latest version (currently 3.1.1).
- Ability to easily export the game to a large number of platforms with just a few clicks.
- It is all there, build the scene and program the script in the same editor, with autocomplete and sintax highlight, no need to install third party heavyweight programs like Visual Studio.
- Easily customizable on the project settings.
- Ability to see the source code and talk to the developers to report an issue (or even solve it myself).
- Enough online resources, both in the documentation, which is extremely complete, as in the community, that teach to use the tool and to solve any problems that may arise.
- The experience of experimenting with and learning a new programming tool and language, with which I had never worked, thus representing an extra challenge on a personal level.

4.2 Redesigned Game Interaction Architecture

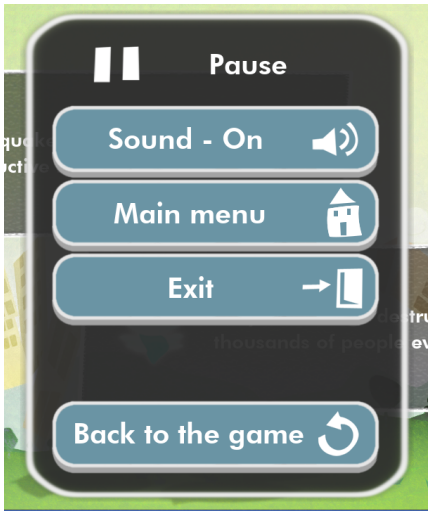
Chapter 3 described how Treme-Treme was initially designed, a simple three-level game that taught earthquake preparedness to young children. The first level is the base of the rest of the game. It explains what an earthquake is, the destruction that may cause to the population either through the destruction of buildings or through tsunamis. This is also where the player choose the character, Sunami or Terramota, and where the first mission takes place, building an emergency kit. Level two had the second mission, leaving a house slowly destroyed by a major earthquake struck and some aftershocks. Finally, the third level consisted of trying to reach high ground before a tsunami caused by the previous earthquakes reached land. Game progressed was not recorded, which meant that if the player exited the game, would have to redo all levels again. Audio was also the only game setting easily modifiable during game play, accessible through the pause menu. To switch language the player would have to press the back button through all scenes to reach the main menu. Once level two was reached, going back was impossible. In addition, the default language was English and sound was disabled. As the main menu do not have any text or indication that audio should be playing, players would have to start the first level and then realise that they did not understand the language. As for the sound, players could play the entire game without realising that sound was available but turned off.

To address all these problems, Treme-Treme's design had to be rethought and the following changes should be made.

- Going to the main menu should be an easy and straightforward process and should be possible during gameplay
- Game progress should be recorded
- Sound should be enable by default
- Language should be set according to the device running the game

The player can now go to the main menu at any time during the game, using the designated button on the pause overlay menu. The title of this overlay window, consisting of a sprite and a label, has also been centred instead of having the sprite in the middle and the label on the right side. The background image is also darker, giving the more significance to this menu. Figure 4.2(a) shows the previous implementation, with the pause menu mixed with the text and images on the background. On the other hand, Figure 4.2(b) illustrates the newst version, with the black background blocking almost completely the text behind, forcing the user to focus on the pause menu.

(a) Pause menu v1



(b) Pause menu v2

Figure 4.2: Pause menu comparison

Some effects have been added to improve the game feel, highlighting the animation of shaking the main screen, which gives the beginning to the theme of the game, earthquakes. The splash screen of Unity was also removed, something that is automatically placed by this game engine in the free version. When the game is started, a splash screen is displayed that disappears after 3 seconds to make room

for the main menu. In the old version, this was the second splash screen and the player had to click to disappear and go to the main menu.

4.3 Development Process

This subsection will describe at a more technical level all the development of the new solution to the problem. It begins by describing three global scripts, which are used to control the core of the game. Then, each of the main levels and screens of the game will be addressed, both the new ones (Section 4.3.2, Section 4.3.3 and Section 4.3.6), and the old ones (Section 4.3.4 and Section 4.3.5). Finally, the persistence of data and the multilingual system will be addressed.

4.3.1 Godot's Global Scripts

In Godot once a scene is instantiated, it loads all its nodes starting from the root to the children with all associated properties, which includes the scripts to control the node's behaviour. Once the next scene is instantiated, the previous one is removed from memory, deleting all the information from that scene which, for example, may include how many stars the player won building the emergency kit on the first try. That being the case, just as in the first version of the game, if the player closed the window, all progress would be lost and would have to restart the game, going through all the levels he had already done to get to where he was before. This can be frustrating if the game has multiple levels with many screens, such as at level 0, and if the player inadvertently closes the game window many times. If the player is frustrated with the game, he will stop playing and the concepts of this serious game will not be assimilated. That said, there are two approaches to keep a record of the game:

Persistence: Information may be physically stored on one or multiple files before changing scenes and loaded to memory again at `_ready()`. This is a clear choice when the game is closed, however frequently opening a file, writing or appending information and a few moments later opening a file to load, parse the information and fill the node's parameters has a cumbersome cost and massively affects game performance.

Master Scene: The game may consist of one special scene that controls all the other ones. The major drawback of this approach is that those scenes could no longer run on their own, they would run inside another scene (the master). It is expected that the behaviour could be unstable and not fully predictable.

Godot supports a master scene alike approach called AutoLoad [2]. This concept is based on the usage of the Singleton design pattern. Multiple scripts and scenes can be added to the AutoLoad list at

[2] https://docs.godotengine.org/en/3.1/getting_started/step_by_step/singletons_autoload.html#autoload

`Project -> Project Settings -> AutoLoad` with a specific *name* which can be called at any given time by all scenes during run-time. This list is prioritised, meaning that at the top is the script/scene first to be loaded and at the bottom the last one. It is worth noting that a new and single *Node* instance is created for each script on the list, added to the root viewport before loading all other scenes.

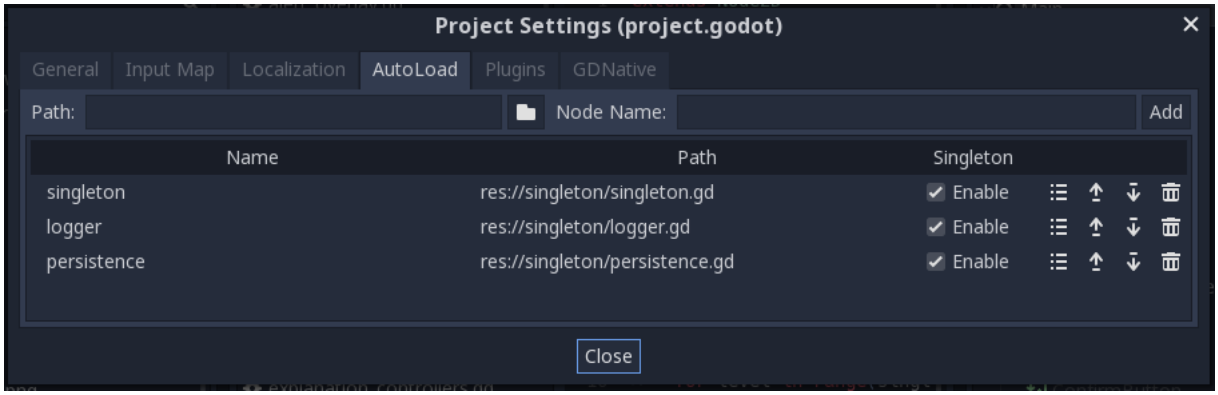


Figure 4.3: Treme-Treme Autoload scripts

Figure 4.3 presents the list of AutoLoad scripts used on Treme-Treme. Starting from the bottom is *persistence*, containing the functions to save/load game status to/from the disk. `save_data(data)` receives a dictionary as input and generates a JavaScript Object Notation (JSON) file, `treme_treme_data.json`, that can be encrypted with a password using the function `open_encrypted_with_pass` to prevent file tampering. The password used varies from the type of the device. Mobile devices use its unique id, obtained with `get_unique_id()` from the `OS` class, to prevent file sharing, however tampering the file to unlock new levels without playing the game cannot be avoided. For all other exports, the password is a SHA256 hash hardcoded on the script and both file sharing and changing are a possibility, however, unless the attacker perform reverse engineer to the code, it is unlikely that file gets changed. `load_data()` does the opposite of the previous function, as it loads the JSON file, deciphers the content and returns a dictionary. If the file is not found, the event is logged and `null` is returned. The file is store on the `user://` directory, which is a read-only folder, used for user resource files and unique on for mobile and consoles.

Then follows *logger*, which provides an easy to use logger Application Program Interface (API) [3]. It contains all the usual levels `VERBOSE, DEBUG, INFO, WARN, ERROR`, follows the format:

```
{LEVEL} {MODULE} {MESSAGE}
```

It also allows saving logs to a file, useful to keep a copy of the stack trace that led to an unexpected error.

[3] https://github.com/KOBUGE-Games/godot-logger

Figure 4.4: Treme-Treme logger message of file load successfully

Figure 4.4 shows the message displayed when the game loaded the information from the JSON file correctly.

At the top of the AutoLoad list is *singleton*, the main game script containing all global variables and functions that act like an interface, wrapping the control of the game's basic behaviour. Specific functions from this script will be explained on the following subsections.

4.3.2 Game Level Selection

This is where the player sees their progress in the game and choose the next level to be played. Figure 4.5 shows this new scene, which gave the player some information about each of the levels before they are played. To do this, all levels have an image and an animation that represents them.

Level 0: image of destroyed buildings that increase and decrease in size.

Level 1: image of the kit with all the possible items to be carried, spinning around it.

Level 2: image of the house from where the player will have to flee, shaking periodically since an earthquake is followed by aftershocks.

Level 3: image of the tsunami from, permanently locked and without animation.

Quiz: question mark rotating horizontally since these are levels with unknown questions.

This new scene, introduced as a new feature of the second version of Treme-Treme has all levels distributed sequentially and linearly so that even without explicitly numbering them it is noticeable that a story is being told, with a beginning (knowing what is and how to be prepared for an earthquake), middle (facing all encountered challenges) and end (reach the meeting point safe sound and find the parents). Each main level has an associated quiz level, which is represented near the level it concerns. As previously stated, the gamification factor, represented in this scene by stars under each level. Three stars can be obtained by the player and show the best performance achieved.

Levels that are still blocked have the property `modulate` with the value `#757575`. This visually makes the button node and its children with a grey tone easily associated with the intended state. A message informing you that the level is locked and how to make it playable is shown if the player clicks the button. Analogously to levels, stars that have not been unlocked have the same property grey.

The level of the tsunami is permanently blocked since it has not yet been built. We have included the level so that the player knows that he can continue playing new levels soon. A message informing you

Figure 4.5: Treme-Treme's game level selection

that this level will soon be available is displayed if the player clicks the button to differentiate from the other levels that may only be blocked.

The game level selection is not shown when the player unlocks a new level. The two current ways to show this scene are:

- The player is on the main menu and clicks the play button
- The player spends the 3 life-level lives at the house mission
- The player completed a quiz level

This scene comes to solve the difficulty of switching between levels without having to go through all the scenes that separate them. The game play is not broken by the transition between levels and reality is represented in a better way. The child should always be prepared for the eventuality of a catastrophe to occur, building an emergency kit that is put to the test soon after. It's worth adding that when the player starts playing the levels, you will only return to this scene if you click the "Main Menu" button of the pause menu, if you complete the game in full on the non-Quiz levels (which at this point corresponds to completing level 2) or whenever a Quiz ends. When a non-level Quiz is finished, the next level is unlocked and the game proceeds there (at the end of the kit's mission, level 1, the player immediately starts playing level 2, for example).

4.3.3 Level 0 - Introduction

This level corresponds to the initial part of the old level 1. Since it introduces essential concepts about earthquakes and, in turn, of the game, it will be approached as level 0. Like the old version, it consists

of 4 scenes, each of the first 3 contains 2 boxes with a sprite to the left followed by text. In the last screen the player can choose one of the two possible characters, Sunami and Terramota, boy and girl respectively.

The most noticeable change between the old version and the current was the removal of the scene that explained the controls because it is no longer suitable for the current times. It consisted of a single text box with the text explanation that the player would have to click the mouse at the desired position for the character to move, accompanied by the image of a mouse. Given that mobile devices nowadays represent most of the devices used to play these types of games, with touch being used as input instead of the mouse and that the way of playing is transversal to many other current games, it was found that this scene could be removed from this level. If the player still wants to know which controls to be used, the same scene can always be accessed from the settings, clicking the *Controls* button.

To speed up the construction of the scenes, some have been constructed to work as templates that can be imported, or linked as Godot calls this process, from others. One of these scenes, *ControlsExplanation*, consists of a Sprite with the background, 3 buttons and *PauseMenu* previously seen in the Figure 4.2(b). This template is quite used by other scenes since it contains the basic controls for the scene change. The main template for this level is *MainExplanation*. Contains *ControlsExplanation*, 2 *Containers* and an *AnimationPlayer* that controls the animation of the containers. Each container corresponding to a box containing the image (*Sprite*) and the text (*Label*) with the explanation previously described. Since in some cases the second Container has text with bullet points, it was necessary to readjust the whole box. Thus, this node contains 3 Label child nodes and the readjustment is done according to the strings given to the Labels. If only Label0 contains text, then the text is centred in the container in the same way as the other container. Otherwise, if Label1 and Label2 contain text, then the container is set to support all text and have the Sprite aligned.

To link scenes, we click the button with a chain icon (Figure 4.1), adjacent to the + and import the template scene as a child node of the currently selected node. Changes made directly to these scenes are immediately reflected in the nodes that were linked. On the nodes that were instantiated with these scenes, it is possible to land properties, but it is not possible to add or delete nodes. That is, the Sprite image with the background of the *ControlsExplanation* node in the *MainExplanation* scene can be changed, but the Sprite cannot be removed.

This level does not reward the player with stars, as it is only an introduction to the game and represents no challenge to it.

4.3.4 Level 1 - Emergency Kit Mission

This level has as main objective the construction of an emergency kit. After a scene with an explanation of what needs to be done, the player enters the main level scene, *KitMission*. This scene, as Figure 4.6

shows, contains ten *Sprite* nodes, one for the emergency kit the remaining nine for each item that the player can choose to take. In addition, it also contains the *ControlsExplanation* with some modifications. The background *Sprite* has a new texture and both the previous and next buttons are not visible and in place of the second a new button with the text *OK* has been inserted, since at this level the player has to confirm the choice made to proceed to the next scene.

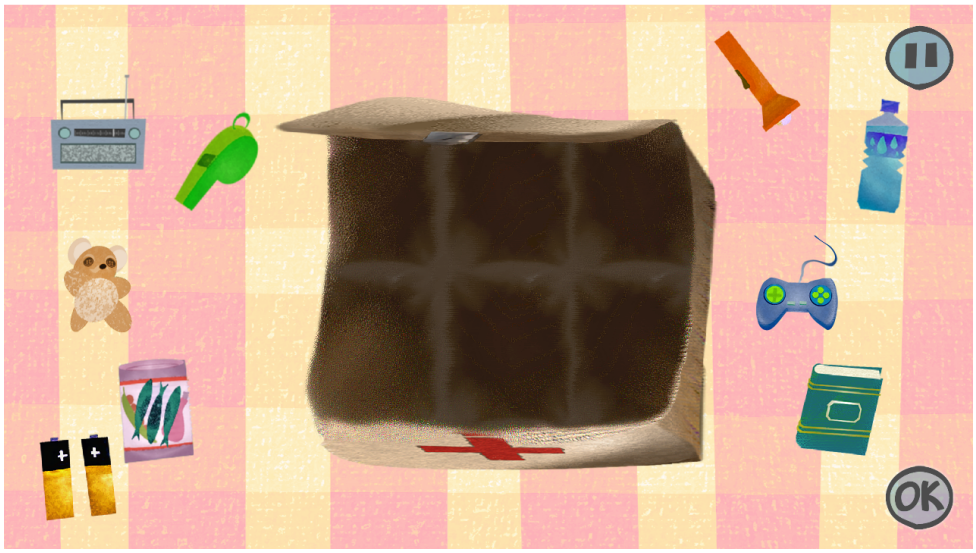


Figure 4.6: Level 1 - Kit mission

The `_ready` function starts by initializing an initial circular transition effect, so that the screen goes black as soon as the scene is loaded and hidden. For this purpose, the *ColorRect* node is used together with the *Tween* node. In the first one, three fundamental properties are specified, the *color*, which has in this case the value `#000000` (equivalent to black), the *shader*, which provides the effect of the transition taking into account the last property, *filter*, Figure 4.7 with a gradient from white to black. Describing at a high level, the effect used with shader works by varying a variable, called *cutoff*, between values from 0 to 1, which correspond to the extremes of the gradient. Since the image pixels range from white to center and black to corners, it is possible to give successive cutoff values over a period of time, which in turn are mapped from white to fade at the beginning of the level, Figure 4.8) or the opposite (fade in before the scene transition). This mapping (from white to black) is defined as one of the shader script parameters (`hint_albedo`), available at Listing A.1.

In order to vary this variable, the Tween node was used, since this node makes vary one or several properties of a given node during a certain time, through the function `interpolate_property()` that receives as arguments the object, the property to vary, the initial value, the final value, the duration, the type of transition (linear, sinusoidal, among others), and the time of the interpolation, (whether it must be faster in the end, at first, both or constant). A delay can optionally be passed, which is 0 if not given.

Then, while the effect is on, a message is displayed in a text box at the center of the screen. This

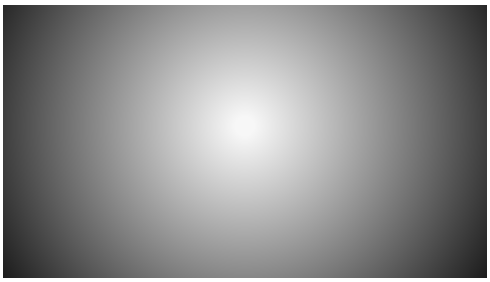


Figure 4.7: Circular gradient texture used for opening and closing some major scenes

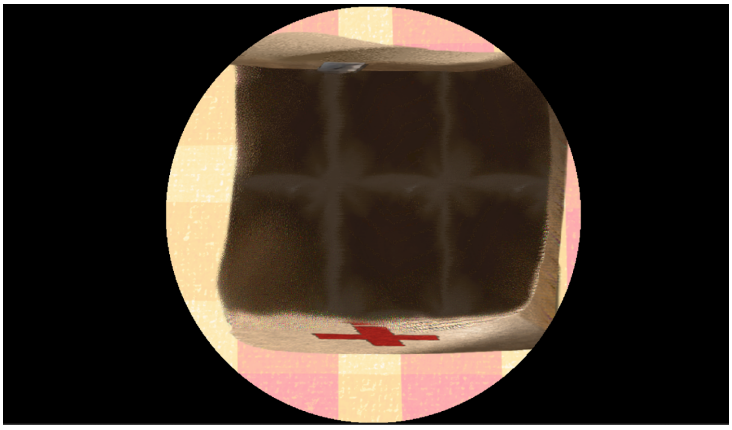


Figure 4.8: Circular gradient effect opening the Kit mission

message is used several times throughout the game, so a template scene was made to speed up the process, called *AlertOverlay*. This scene should be linked by the scene that you want to use and initially should have the property *visible* to `false` and is composed of 2 Sprites, one box for the background and another for an image associated with the text, 1 Label, with the text to be displayed, and a Timer, which by default has the attributes `wait_time` to 2 seconds, `one_shot` to `true`, so that it only runs once, and *autostart* `false`. To call this overlay, it has been established that a function called `show_alert` must be created in the scene code that calls the overlay. In this function, both the Sprite of the image associated with the text and the text itself must be updated, the Timer must be started and finally the *AlertOverlay* node should be shown. `can_skip` is an optional parameter of this function, which is `false` by default and is used to explicitly enable the player to disappear the message by clicking the mouse. When the Timer reaches 0, *AlertOverlay*'s `_on_Timer_timeout` function is automatically called, which hides the message, stops the Timer and triggers a signal for a second function, `alert_dialog_timer_ended`, which must, like the previous one, be implemented in the code of the scene that calls *AlertOverlay*. This is used to keep the game going after the message disappears. For example, when we can change the scene after the level score is shown, starting the fade in effect. Figure 4.9 shows an AlertOverlay example.

As for the item positions of the kit, they are chosen randomly from among nine existing predefined *Vector2D* points, still in the `_ready` function. There are two lists, one with the strings of the names

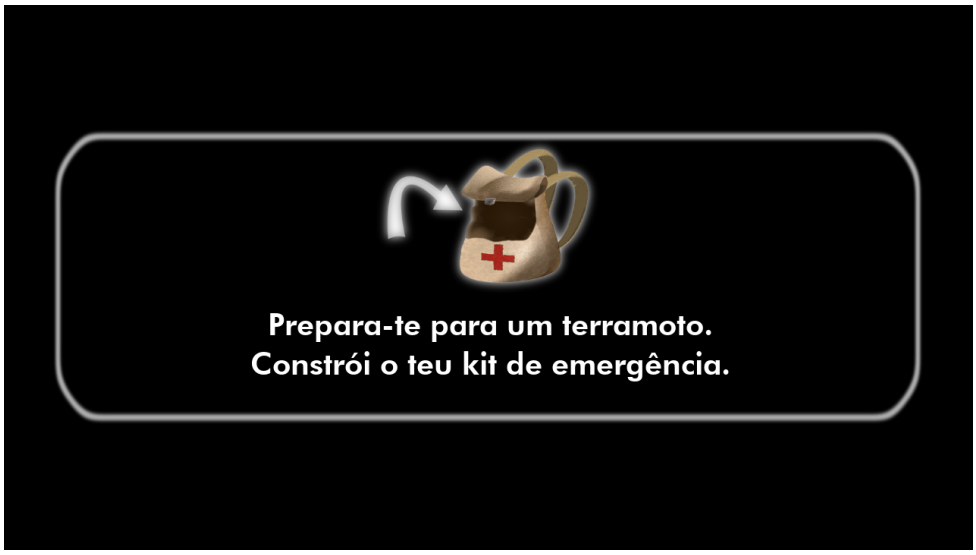


Figure 4.9: Example of an AlertOverlay message with a fade out effect behind

of all the items and another with the possible positions, which is randomized through the function `singleton.shuffle_array()`. A for loop that starts at 0 and ends in the length of the items, assigns the position value of the list of possible positions to the *position* attribute of the object returned by the `get_node(items[i])` function.

As for the logic of the level, the player has to choose which items to take with him in an emergency kit. To do this, you must drag them into the kit, which is divided into 6 parts, or containers, the equivalent of a 2x3 array. To control which objects are dragged into the kit, we used the *CollisionShape2D* node, which detects collisions within a predefined area. In Figure 4.10 it is possible to observe all these areas, blue rectangles around the objects and that are a little larger than these. It also shows three types of collisions resulting from the player's input, in type 1, we have the correct items (*Batteries*, *Bottle*, *Can*, *Flash*, *Radio* and *Whistle*) that only have an Area2D node, allowing the player to drag the item this process will be explained later) and a *CollisionShape2D*. Type 2 is used on items that should not be chosen (*Bear*, *Book* and *Console*) and is similar to 1 with the difference of including an animation that is played when the item was included in the kit, giving visual feedback to the player of your error. Finally, type 3 is included all 6 containers of the kit, which cannot be clicked or dragged by the player, only to identify where an item was placed. There is also a Sprite with the image of a square screen that is shown for 1.6 seconds in containers without item when the player clicks on the *Ok* button, thus warning that something is wrong.

During the movement, the `_process` function, which runs automatically once on each frame, checks if there is any item to be moved and, if it is the case, the distance between the center of the item and the mouse is calculated and multiplied by a factor of 20% to smooth the movement and find the new position to the center of the item. Before assigning this new position, it is checked if it is outside of the viewport

Figure 4.10: Tree of nodes used for the 9 items and 6 containers and the respective collision shapes

and if it is, the point closest to the mouse is found and assigned and the item is completely visible.

Mouse click detection is done through the signal `input_event` of the *Area2D* node of the item under the mouse position, which triggers the `_on_Area2D_input_event` function with the custom argument `node_name`. The function verifies that the event was generated by the left mouse button and that there is still no object to be moved. This second check is necessary to ensure that the signal is not triggered while one item is dragged over another. If the requirements are met, the move is started by invoking the function `start_movement(get_node(node_name))`, which in turn saves the node being moved, its starting position and, if it does, the container where it was placed.

When the object is dropped, with the player stopped clicking the left mouse button, a new event is triggered, instantiating the overridden `_input` function. If the event's `pressed` attribute is set to `false` and the `moving` variable is set to `true`, the `stop_moving` function. In it, the first *Area2D* nodes are selected, and items that intersect the item that is being moved. If one or both do not exist, the corresponding variables are set to `null`. Taking into account these variables, as well as those recorded at the beginning of the movement, a decision is made as to which action to take, taking into account the following 9 possible cases:

Case 1: was not on container -> is still not on container: the item is in the position where it was dropped

Case 2: was not on container -> is still not on container but replace position of another one: the item is in the position where it was dropped and what was there going to the initial position of the one that was moved

Case 3: was not on container -> is on empty container: the item is in the container that was empty

Case 4: was not on container -> is on container but replace position of another one: the item is in that container and what was there going to the initial position of the one that was moved

Case 5: was on container -> is not on container: the item is in the position where it was dropped and the container is empty

Case 6: was on container -> is not on container but replace position of another one: the item is in the position where it was dropped and what was there passes to the container where what was moved was

Case 7: was on container -> is on container and is the same: the item resumes the starting position

Case 8: was on container -> is on container and is another: the item is in the new container and the initial container is empty

Case 9: was on container -> is on container and is another but replace position of another one: the item is in the new container and the initial container is with the item that was changed

In cases where it is necessary to change positions between items, they are added to a list, which is iterated in the `_process` function so that the transition occurs over several frames in order to be smooth. This type of movement is achieved by calculating the distance between the current and the final position on both components and multiplying it by a factor of 30% so that it slows down as it nears its end. Since this movement only tends to 0, when a threshold distance of 1 unit is reached, the movement is stopped.

By completing this level, the player can receive between 0 and 3 stars, inclusive. Whenever the scene begins, it is assumed that the player has 3 stars and for each time the ”OK” button is pressed while the kit is incomplete but with at least 1 item already placed or poorly constructed, the number of stars is decremented. The player can try this level as many times as he wants, and after 3 attempts will always receive 0 stars. In addition, if the player places the wrong items in the kit and click on the *OK* button, the generic message will be shown saying that something is not correct. On the second try, if any of the items are still correct, an animation will be shown to give a hint to the player of what items should be removed.

4.3.5 Level 2 - House Mission

This is the most complex level of the game, both in terms of implementation and gameplay. At this level the player must escape from home and meet the parents while an earthquake is taking place, followed by several aftershocks. To reach the goal, you need to find hiding places to protect yourself from rocks, glass and furniture, turn off gas and electricity, and pick up a survival kit.

This level consists of 6 main parts: canvas layer, camera, background, house, character and navigation path, organized as shown on Figure 4.11.

Figure 4.11: Level 2 viewport

4.3.5.A Canvas

The canvas layer contains all the elements that are fixed to the window, that is, that always keep the same position for the player, even if the window moves. Godot, by default, assigns a canvas to each scene. If a *CanvasLayer* node is defined, it overrides the default. The child nodes of the *CanvasLayer* are: *ControlsExplanation*, an *AlertOverlay*, a *Tween* along with a *ColorRect*, to perform the same fade in and out effect as Section 4.3.4, when entering and exiting the scene, respectively, a *Sprite* with dozens of rocks that drop and fill the screen after the third replica has strike with the player still indoors, and last but not least a *TimerOverlay*. This last one is a scene that contains a countdown of 10 seconds with a clock in the lower right corner, as seen on Chapter 3.

4.3.5.B Camera

Like the canvas, Godot associates a camera with each scene by default. For this reason, none of the other scenes has a *Camera2D* node. However, in this scene, we want to have more control over the camera, to zoom in or out of the character, to follow it when it moves, to shake when an earthquake or a replica takes place. The camera's movement is dictated by the point, defined by the *Position2D* node, which is following at that moment. When the mission begins, the camera begins by following a central *Position2D* to the house and, after a message is displayed, the *Position2D* to be followed becomes the centre of mass of the *Player*. With this change, a zoom in of 1.4 is made to the new point, for 5 seconds, using a *Tween*. As for the shake, this is done by adding a random value, between -1 and 1, to both components of the current camera position, to each frame for 15 seconds, and also to sound corresponding to the earthquake. Whenever the shake is started, and the character is indoors, the camera zooms out to the middle position of the house.

4.3.5.C House

The house, like the background, has a root node of type *Node2D*, in order to easily aggregate each of its elements. Inside the house node, there was a second split between each of the divisions, similarly made with *Node2D*s, which finally contain *Sprite*s and *AnimationPlayer*s. As for the animations, there is only one per division of the house that controls the rotation, scale and position of all *Sprite*s in that division. As the earth can shake up to three times, while the character is indoors, before the player loses, three division animations have been made, following the template `[division's name]_shake[1:3]`, except for the balcony, which contains only the animation of the player's death. As for the textures used, and since this is the scene that has the most textures, sprite sheets were made with all the images, thus combining them in only 1 image that can be loaded quickly once and cut in the different sections. This cut-off is done automatically by the TexturePacker Importer addon when importing the *.png* and *.tpsheet* files, which contains the compiled image and the information where each texture begins and ends respectively. The images appear differently in a generated folder, always with the name *.sprites* appended, facilitating the work of the programmer.

There are two special types of animations where the *Particles2D* node was used instead of *AnimationPlayer*: on the rocks falling from the ceilings during earthquakes and sparks coming out of the electricity cable. As it is intended that several instances of the same texture be emitted from a single position during a certain time, this node was chosen, this being the main reason for the project to be rendered using GLES3, as mentioned in Section 4.1. In the case of stones, 3 *Particles2D* have been created that emit 12 stones in descending order, interspersed in random time intervals for 5.1 seconds, with a linear velocity of 50 units per second and velocities of 1.7. In this way, rocks always fall twice per earthquake. As for the electricity cable, it consists of a white *Polygon2D*, subdivided into 10 equal parts, each with an associated *Bone2D*, that bring the movement of the cable closer to reality.

4.3.5.D Character

The character can spawn in 6 different positions, being chosen a random position by the function `_ready()` and has as main node a *KinematicBody2D*, which contains specific properties for this type of objects, such as gravity, collisions and physics. The body of the character is composed of 12 different *Sprites* and interconnected with each other, daughters of the *Position2D* node, which represents the centre of mass and is used by *Camera2D*. As there are two characters, Sunami and Terramota, not all textures are the same, so at the beginning of the level is made the correspondence between the nodes and their textures. To facilitate this process, the texture files follow the `[name_of_the_character]_[name_of_the_texture].png` template. In addition to these nodes, there is also an *AnimationPlayer* where the idle, walk, cover and jump animations are defined. The first 3 are used, respectively when the character is standing still waiting for player input, moving and sheltering

under the bed or one of the tables. As for the fourth, it is used when the character meets the parents, showing the player how happy they are to achieve the goal.

When the character is in the immediate vicinity of the emergency light or gas tap and has not yet deactivated them, a balloon with the respective symbols is shown, indicating to the player what should be done. In addition, the character may die in various ways, depending on where it lies during the time the quake occurs. This detection is done through the *CollisionShape2D* nodes. Several of these nodes have been spread through the navigation path so that the player is always in contact with at least one of them, triggering events when entering and leaving the collision area that modifies global variables and, when necessary, these are read and the decision is made of how the game should behave, either by showing a message that the player took refuge in a place, or died on the stairs, proceeding to the animation of the stairs to break in half, or until when the player leaves and re-enters at home.

4.3.5.E Navigation path

Navigation is related to the house and the player. In Figure 4.12 it is possible to see, in red, the path that the character can follow, a polygon of type *NavigationPolygonInstance*. This path was defined taking into account the distance from the centre of mass of the character to the floor of each room of the house, as well as its lateral distances to never hit the walls go down/up the stairs in the most natural way possible. The position of the mouse is obtained whenever the player clicks on a point on the screen, passing it as a second argument to the function `get_simple_path` of the *Navigation2D* node. The first argument of this function is the initial position, which corresponds to the mass centre of the character, and the third argument is a boolean to optimize the path choice, returning a *PoolVector2Array*. The initial *Vector2D* is excluded since it is the current position of the character, iterated over the remaining ones. At each iteration, the character moves in a straight line between the position it has at that given moment and the current iteration's *Vector2D*. If the player clicks the screen before the character reaches the end of the iterations, the previously calculated path is deleted, and a new path is immediately obtained.

This level differs from the previous one as to the reward that gives the player on the form of stars. Upon completion of this level, the player may also receive between 0 and 3 stars, inclusive, but is limited to 3 attempts. After these 3 attempts, a message is shown that the player is not yet prepared to face an earthquake and is forwarded to the level selection scene. In addition, whenever the house scene begins, it is assumed that the player has 3 stars. Each time an earthquake occurs, the number of stars is decremented by one unit and it is certain that at least one earthquake will occur at all times. The player can increase the number of stars by turning off the gas and electricity. As it is necessarily necessary to turn off the gas to complete the level, the loss of a star with the first earthquake is ignored. In case

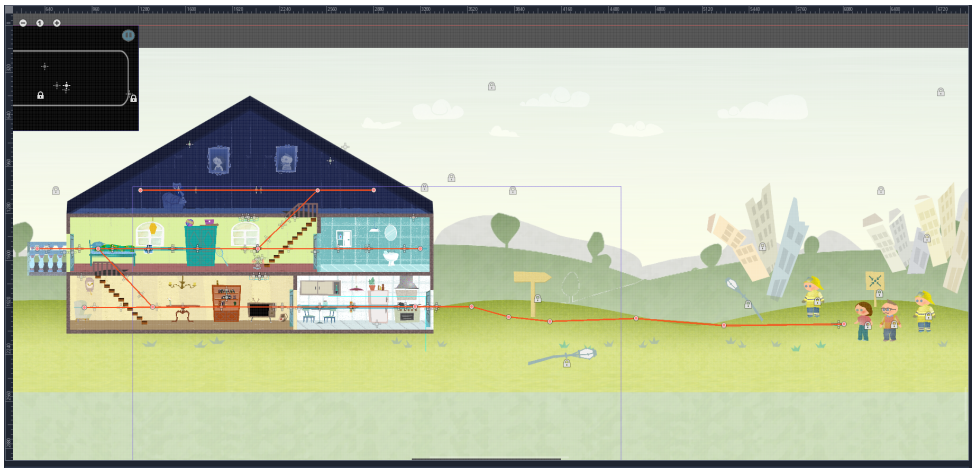


Figure 4.12: Level 2 player navigation path, in red

the player turns off both gas and electricity, completing the level with only 1 earthquake occurred, the number of stars will be 3, never exceeding this number.

4.3.6 Quiz

Figure 4.13: Level 0 quiz - wrong answer

These levels were introduced as yet another act of gamification to test the player's knowledge about what they have just learned at each level. Thus, if the player does not receive all the stars of a *Quiz* as a reward, they will either go back to the *Quiz* or re-play the level to make the concepts clearer. This promotes the player's engagement with the game. As this game was intended to be played in a classroom, these small levels are optional, so that the teacher may not include them in his class, asking

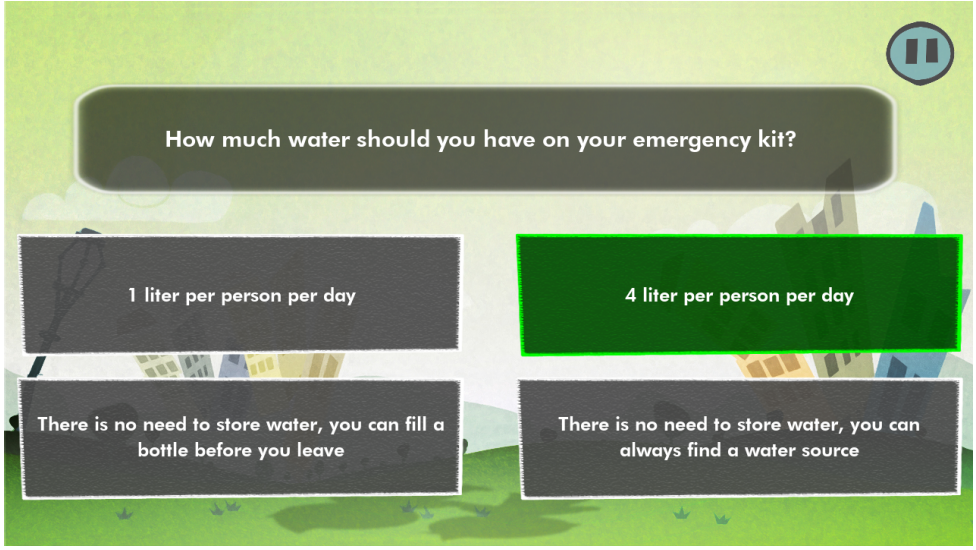


Figure 4.14: Level 1 quiz - correct answer

questions or evaluating himself to his students.

In terms of development, the objective is to present 3 questions that test the knowledge acquired during the accomplishment of the associated level. The questions are randomly selected from a pool of questions, using `singleton.shuffle_array()`, so that the player is required to read both the questions and the answers and does not redo the level mechanically.

The game scene consists of a label at the top with the question to ask and buttons with the answers to occupy the bottom of the screen. The number of responses can vary between three and four, so two of the buttons do not have fixed positions and the rest do not. Figures 4.13 and 4.14 show how this level is in the two possible variants.

When the player responds incorrectly, the `modulate` property of the clicked button changes to red and a deep sound is played, penalizing the player's choice both visually and audibly (Figure 4.13). On the other hand, when the player chooses the correct answer `modulate` is changed to green and a sound saying *yeah* is played. After a 5 second delay the next question is loaded.

After the third question answered, a text message is shown to report their performance along with the number of stars won.

4.3.7 Game persistence

Something missing in the previous version and that becomes more significant as the number of levels increases is the ability to save game progress. If a player accidentally closed the game or could not continue playing both the progress made and some changes to the game settings would be lost. Nowadays, virtually any game keeps some information, either locally or online through the creation of an account.

Therefore, this functionality was implemented by saving the following global variables from the script `singleton`:

- **Sound**: **boolean** where *true* represents sound on and *false* sound off
- **Language**: **string** with the language locale to be used
- **Character**: **string** with the character name
- **Current scene**: **string** with the scene identifier where the game should be continued
- **Statistics**: **dictionary** where the keys are strings with the identifiers of the existing levels and the values an integer with the number of stars got, in the closed range of 0 to 3

This solution was initially designed, so that progress was saved whenever a scene change occurred, by calling the `singleton.scene_changer` function. In this first prototype, the concept of the game overview did not yet exist, so by pressing the play button on the main menu, the player would resume the game in the last scene he was in before leaving. By introducing the game overview and after observing the logs produced, it turned out that too many unnecessary calls were made and the solution was redesigned to save data only when warranted. Therefore, data is saved whenever one of the following events occurs:

- The game is closed
- Some changes in sound or language settings are made
- A level change occurs
- A new character is chosen

To save the data, it is called singleton's `save_game()` function that creates a dictionary, with the states of the previously mentioned variables, that is passed as an argument to the `persistence.save_data()` function. This dictionary is then transformed to JSON and saved to the disk as described at section 4.3.1. Similarly, to load the data is called the `singleton.load_game()` function that generates a variable with the return of `persistence.load_data()` and, if not null, parse the data and changes the global variables.

The `_notification()` function, overridden in the `singleton` script, automatically detects if the request was made to close the game. This is a function of the `Object` class that notifies the object, in this case, the *singleton* node, that something has happened. This *something* is an integer passed as an argument and the overridden function tests if it has the value `MainLoop.NOTIFICATION_WM_QUIT_REQUEST` (which seeing the documentation has a value of 6). If so, the game is saved and then closed.

4.3.8 Multi language support

Treme-Treme is a game planned to be played by children from different nationalities, therefore the first version had support for three languages, Portuguese, English and Italian. Translations were stored on Extensible Markup Language (XML) files, handled by a Message Manager class that parsed files using a XML Reader. XML is a very human readable format that supports hierarchical data structures. It is an extremely verbose file format, up to three times as large as other formats like Comma-Separated Values (CSV), used by Godot. This is similar to a table and may be harder to read by humans, however it is easily parsed and imported and exported to common editors used by translators, like Microsoft Excel.

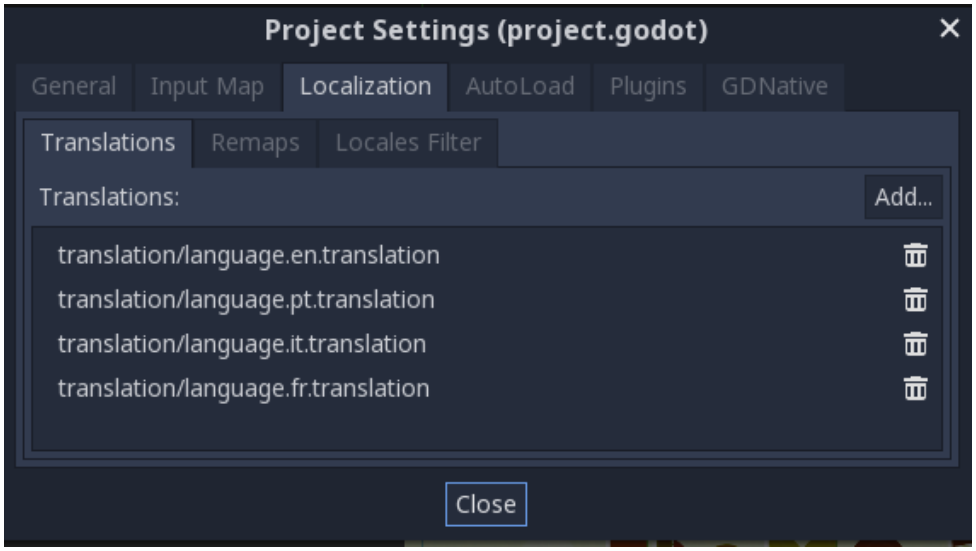


Figure 4.15: Treme-Treme's second version available translations

Godot imports CSV files as translations by default. It generates a `[lang].translation` resource file on the same location as the imported file and based on the CSV header. `[lang]` is the column header and a valid locale. The list of Godot's valid locals follow the Unix standard locale strings [4]. This resource files contain a key/value table, where key is the first CSV column string, in capital letters, and the value is correspondent the translated string. Figure 4.15 shows the list of available Treme-Treme translations and the resource files created after importing the CSV file. In addition to the previous three languages, both French and Spanish are now available. Translations are managed by the core object TranslationServer [5]. New Translations are set and removed from it at run-time by the `singleton`'s function `next_language()` and FIFO `languages`. When the player request to change the current language, this function is called and starts to pop the first value, sets the locale to it and pushes back to the FIFO. This means that the last value is the current used translation. When the game starts and if a record from a previous play session exists, this function is called on a loop until the loaded locale is reached.

[4] https://docs.godotengine.org/en/3.1/tutorials/i18n/locales.html
[5] https://docs.godotengine.org/en/3.1/classes/class_translationserver.html

All Label node's `text` property is filled with the key that identifies a specific string. Then, through the function `singleton.get_parsed_translation(key)`, the correct value is return containing the translated string. This function first calls TranslateServer `translate(key)`. Then any `\\n` is replaced by `\n` due to limitation parsing the CSV file. `\n`s present on the file are interpreted as characters, so the backslash is escaped by the parser and `\n` is printed instead of generating a new line. This action is required because the strings are not made to fill a single line but an entire textbox and depending on the language it may not be visually reasonable to always split the sentence after a fixed number of characters. This way, the translator can specify where the string should be splitted.

4.4 Deploy

The main objective of this work is to put the game back online so it can be played. As a complement, the game must be published on multiple channels and available for several platforms. Currently the market for mobile games has increase almost exponentially, due to the rapid increase in the number of carriers of a smartphone, so this is an area where the game should be available.

Therefore, the game was exported to the 3 main platforms:

- Web, using HTML5
- Android
- Windows, MacOS and Linux

The game can be played directly in the browser by accessing http://treme-treme.pt/, and it is also possible to download it through this site.

As for Android, the game can be downloaded through Google Play (https://play.google.com/store/apps/details?id=pt.treme_treme.game) or Aptoide (https://treme-treme.en.aptoide.com/). Since apps for children under the age of 13 need to meet more stringent requirements, a Privacy Policy section has been added to the official game site, otherwise Google Play would not allow uploading of the game.

A cross-platform problem, but mainly affecting the web and mobile, is the size of the assets used. Although some steps were taken to reduce the size of images without losing quality, that was not enough. On the web the game takes some time to load, even with a good connection speed from the client browser and the mobile, the application turned out to be 77MB.

The main objectives proposed on Section 1.2 have been accomplished and improved. The game has been rebuilt successfully and everyone can play it without having to install any extra software other

than the game itself. When exporting to the web, it is possible to play directly in the browser, accessing the site where it is housed, further facilitating access to the game, which is published on the existing site. In addition, the game has been exported to Android, Windows, macOS and Linux. The game offers support for French and Spanish, two new languages, beyond those already supported, and gamification was introduced to make the game more interesting, challenging and better convey concepts about earthquake preparedness.

As for this chapter, it was described the entire process of implementing the new version of Treme-Treme. We started by describing the tool used and the reasons for the change made. Changes were then described in the interaction architecture of the game, which will be evaluated in the next chapter. Finally, the most relevant technical parts of each level, the persistence of game progress and the translation system were detailed in order to make known the practical work done.

5

Evaluation

Contents

This chapter presents the evaluation of the new Treme-Treme version. The evaluation activity aims to observe and measure how well the newest version conveys procedures that should be taken during an earthquake to the players in comparison to the previous version. The activity also intends to determine how the game may be used as a multimedia tool during classes that address the topic of earthquakes and tsunamis preparedness so that students can better assimilate the subject.

5.1 Methodology

Three methods were used in order to compare how both Treme-Treme versions transmitted the seismic knowledge to the children. Observation, interviews and data collection.

5.1.1 Observation

The children's behaviour and engagement were observed throughout the activity and some of the comments made were noted. This method is the best way to compare both of the game's clues since the former does not contain data persistence. By looking at, for example, which items to take in the emergency kit or how players die on the second level, we have a perception of how the knowledge is being passed and how they are learning. The constructive discussions that the children were having during the course of the game proved to be another way of obtaining relevant data to justify their choices. Teachers have also been observed to have a perception of the reaction to the game since the game should be made available by them in the classroom and is intended to know if it is a good tool.

5.1.2 Interview

It was not possible to gather formal consent to conduct individual surveys in time, so questions were made to each group before and after each session. Besides children, professors have been requested to answer some questions and provide some feedback from the changes made and how suited the game was to be used as a teaching resource. These questions have been made while the students played and after all groups completed the activity.

5.1.3 Data collection

The latest Treme-Treme version has the ability to keep log files of the activities, as detailed on Section 4.3.7, allowing us to have more groups trying the game at the same time and complementing the research with the time spent on each scene, of the retries on both Kit and House missions and the reason why the player died. Unfortunately, we could not extract any data from the previous version of

the game since the code was not reusable or modifiable and my colleague only used the previous two methodologies on his evaluation.

5.2 Procedure

As Treme-Treme's target audience are young children, specially students with ages between eight and ten years old, the activity was conducted at Casa Pia de Lisboa with the third and fourth grade students on April 23rd, 2019. A total of 44 children participated, 20 from the third grade and 24 from the fourth grade, testing both versions of the game in different configurations. All students had already studied earthquakes at school and knew some of the basic procedures to take in case of an emergency. None of the students played Treme-Treme prior to that day, but some already knew the existence of the game. From the 44 students, 14 were girls and 30 boys.

Figure 5.1: Group of children testing Treme-Treme at Casa Pia de Lisboa

Students were split into groups of four, as shown in Figure 5.1, in a total of 11 groups, due to a limited number of available devices and time for the entire session. Each group session started with an introduction to the game and an interactive explanation about earthquakes preparedness, given by my supervisor Mónica Amaral Ferreira, an earthquake specialist. Comments were collected during the explanation about the knowledge on the subject. To simulate a class environment, the children played the game with a professor guiding them, especially during the *Quiz* levels. After they walked through all levels, they switched places with the next group and some questions were asked to understand what they thought of the game if the new quiz levels were a good addition and to establish a comparison between both versions. The quiz levels also tested the specific knowledge addressed on the game like, for example, what items should be on the kit or what should a person take before exiting the building.

The groups took on average 15 minutes each due to the short time and limited number of computers

available for the number. Since both versions are similar, to compare them better and try to remove the pre-knowledge factor of the game, the groups were divided into the three following combinations:

- Two groups played first version 1, then version 2
- Two groups played first version 2, then version 1
- Seven groups only played version 2

During gameplay, six randomly selected groups have been requested to execute some special tasks, which will be addressed on Section 5.3.2, in order to get feedback and analyse the behaviour on some features of the game. Two different groups per task were chosen, one from each grade, in order to eliminate previous knowledge of the location of the menus and buttons and to get feedback from children of different ages.

5.3 Results

The results presented are divided into two components. In the first, a comparison is made between the two versions based on observation data, questionnaires and game logs. The second component portrays the special tasks that were asked of the children to evaluate the architectural changes made to the game. In these is first reported what flow expected to perform, followed by an explanation of what children actually did, based on behaviour observation.

5.3.1 Versions Comparison

During this evaluation, 4 groups of children were asked to test both versions in different orders to try to establish a method of comparison between the two. The biggest concern in doing this was that the children were biased because they had already gone through all the scenes, not paying as much attention as they would have if they were seeing them for the first time and completeness. But this was not revealed. The children gave as much importance to the first version they tried as to the second version, reading the whole theoretical part aloud in turn and debating which items should be placed in the kit or where they should be hidden inside the house. Because the results obtained did not prove to be biased, they were included in the graphs that followed on both versions. Our evaluation also include the data obtained in 2014 to assess whether the game still remains a current-day tool.

5.3.1.A Goals

The work previous done established 5 main goals for Treme-Treme, which are still fundamental for us and are used to evaluate the accomplished work.

Goal 1 - The player must know how to build a simple emergency kit and which objects are the most important for his survival

This was the easiest and fastest mission level, where it was intended that only the strictly necessary objects be placed in the kit. All groups managed to pass this level, taking, on average, 72 seconds to do so. 64% of the groups collected 3 stars, 27% 2 stars and only one star in 1 of the groups, which corresponds to 9%. None of the groups received 0 stars, that is, everyone clicked the *OK* button with the kit partially or incorrectly filled in a maximum of 2 times. Compared with the results obtained in the version 1 evaluation, there was a huge discrepancy in the data, since previously 80% of the children failed 2 or more times. At that time the children did not know what objects to put in the emergency kit, but all had managed to complete this level, which indicated that while playing there was an apprenticeship of what was essential to transport to survive an earthquake.

It was expected that some of the objects were not identified by children, such as the radio that nowadays fell into disuse, with the emergence of smartphones, but this was one of the first objects to be placed inside the kit, revealing that the game continues to be acquired, even having spent half a decade on its idealization.

Goal 2 - The player should know where the dangerous and safe places are and what to do in each situation

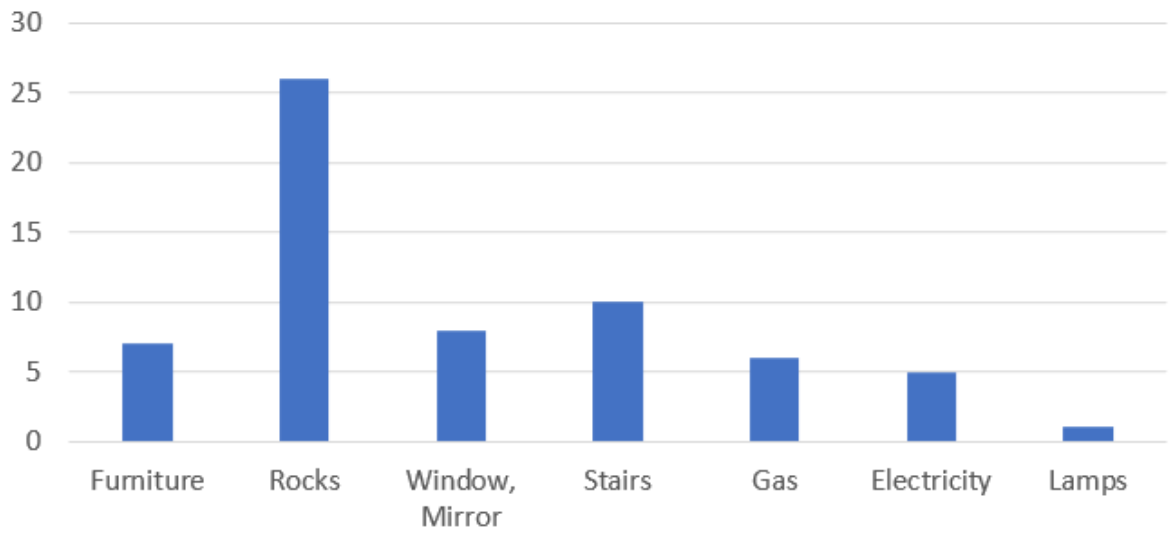


Figure 5.2: Sum of deaths, by type, from all groups

This group of players, like the one who carried out the tests to the first version, showed to have a piece of good knowledge about where they should hide during an earthquake. As such, it became easier for them to identify the best places to protect themselves when the countdown reached 0. However, there were a large number of failed home-level attempts and recorded deaths are shown in the Figure 5.2.

The rocks were the source of the highest number of deaths since it covers most of the area of the house. Whether permanently, in all places where there is only one roof of the house, or sporadic, like the rocks that spawn randomly in the 6 possible sites, even if they are already covered by stairs, for example. In addition, the players decided to explore the various divisions of the house, even knowing that an earthquake was imminent, further increasing the number of deaths. This is not directly reflected in the player's lack of knowledge about the occurrence of an earthquake, but rather that it is a game and that the exploratory factor is always present, especially in children. Another problem that also contributed to some deaths was that the path was not perfect and when players clicked under the bed, the character began to descend the stairs as it was the closest point to the given place as input.

As for the identification of safe places where the player should be housed in case of an earthquake, most groups were able to quickly identify these sites. Only two groups were killed at first by the facts (exploratory and bug of not being able to hide under the bed) previously stated.

With the exception of 1 group, everyone has gone through at least one replicate in at least 1 of their attempts. Something relevant was the fact that all the groups left the house before the first replica happened, in the attempt in which they managed to pass the level. This may demonstrate that the children have been discovering the ideal actions by trial and error, perfecting their knowledge the attempt until they feel perfectly at ease to face an earthquake.

Goal 3 - The player must be aware that gas valves and electrical switches must be turned off before leaving home

In the earlier assessment of the development of the first version of the game, players were forced to turn off both electricity and gas. The results showed that 92.50% players did it and they were able to understand the importance of doing it in the real world. In the second version, as it was reflected in the first one after the evaluation, to complete the level, the player is only obliged to turn off the gas to complete the level, being the electricity optional. It was observed that most players failed to immediately understand what was supposed to be done, even with the balloon containing the image of the gas tap/electricity switch near the character was shown, with 55% and 45% of the groups dying at least one for gas and electricity, respectively. One group only completed the second level on the third attempt, finding shelter correctly but dying 2 times in the process, one for electricity and one for gas. On that attempt we observed that the group turn off both the optional electricity and the mandatory gas, proving that they learned with the mistake.

Goals 4 and 5 - The player must understand how dangerous earthquakes can be, how much time a person takes to protect himself and the importance of leaving home after an earthquake to his family emergency meeting point far away from buildings

The players showed they knew they had to leave the house as soon as possible, but out of curiosity, they were more interested in exploring the various divisions of the house. On average, players took 48

seconds to complete the second level and the fact that they did it once did not represent a significant reduction in time from a later attempt. As a rule, attempts, where players completed the level, were the longest, even knowing the map. It was observed that players in the first phase were disoriented when what to do to complete the level, as well as what was the countdown. After they knew what the second was, as soon as he appeared, the first instinct was to find shelter, even while already in the kitchen with the gas off and near the exit door. After leaving home, all groups understood that they were supposed to find the meeting point and none returned home, having the arrow with the text that indicated the way to play an important role in this decision.

5.3.1.B Gameplay

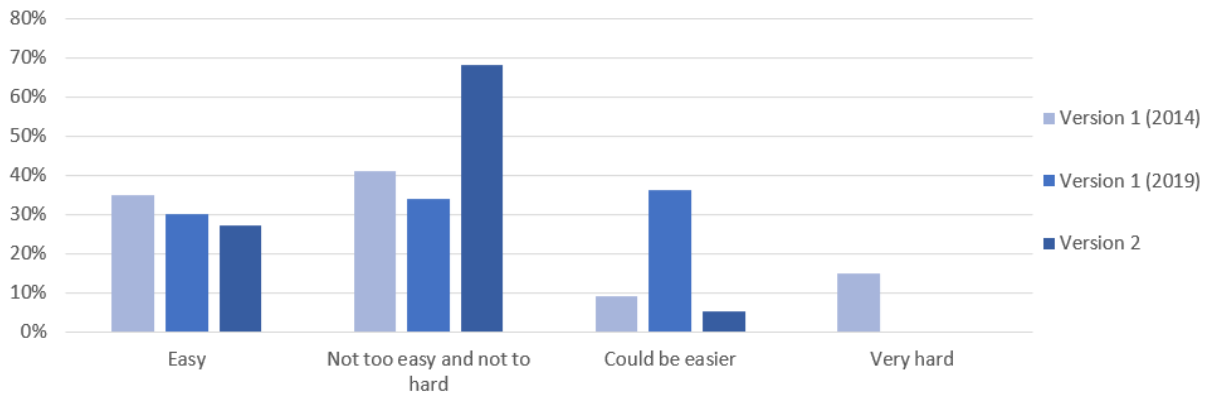


Figure 5.3: Game difficulty - comparison between both versions, with data from the previous evaluation

There was a noticeable improvement in the difficulty of the game, and the vast majority of participants now think the game is well balanced, as seen on Figure 5.3. This reveals that even with the introduction of questions, which by and large make the game more difficult and boring, it remains something that children are interested in playing.

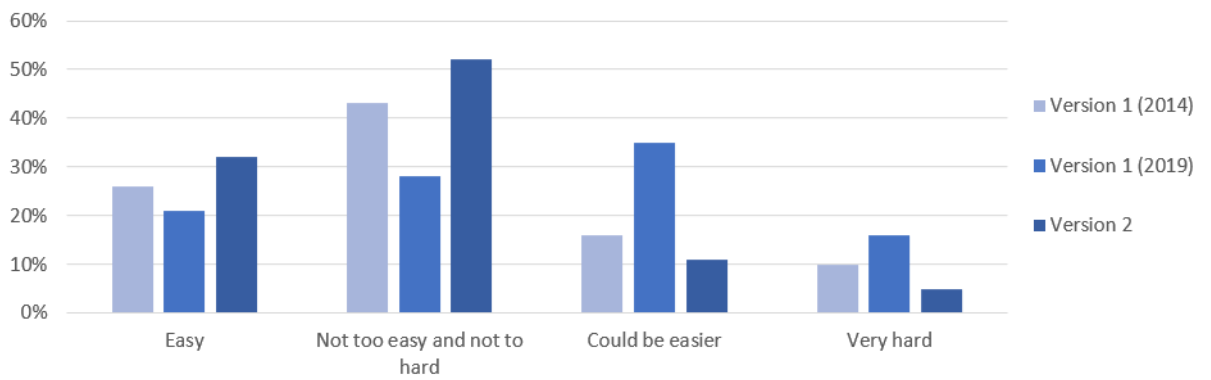


Figure 5.4: Controls difficulty - comparison between both versions, with data from the previous evaluation

Regarding game controls, there is a slight improvement from the previous experiment to the current, as seen on Figure 5.4. However, some participants still think on some occasions the controls should be better. This fact was noticeable by observing their behaviour at the house mission when trying to hide under the bed. As the stairs are very close to the bed, the navigation mesh is not ideal and when the player clicks under the bed, the closest detected point is halfway down the stairs, inducing confusion to the player. Nevertheless, this better result may relate to the fact that the participants now use smartphones to play this type of games instead of a PC with a mouse, which have similar controls to Treme-Treme. This means that both the initial choice of controls, back in 2014, and the decision to maintain them were the best option. Unfortunately, it was not possible to test the gameplay on a mobile device to prove this hypothesis.

5.3.2 Special tasks

These special tasks have been asked of the children as a way of verifying that the changes made to the game with the development of the new version contributed positively to the same. As it was previously impossible to return to the main menu after starting the game and being already at the kit level, making it impossible to go to settings, for example. Thus, it was initially the children who turned off the sound of the game, something that was already possible on the previous version. In doing so, the children had to open the pause menu, which had the option of returning to the main menu. Then they were asked to change the language of the game, testing the way to the main menu. It was finally asked to change the character, something that was also impossible to do on the first version if the player was already on the menu of the house. The latter required the player to think about all the game architecture since there is no direct path to character change from the level selection scene.

5.3.2.A Disable sound

In this case, the objective was to turn off the sound while playing a level, the expected flow would be:

1. Click on the pause button on the top right corner

2. Click on the sound button

This flow was easily achieved by the group from the fourth grade, however similarly to the previous task, the group from third grade was not able to associate that the pause button would allow them to change some game settings.

5.3.2.B Change language

The objective was to change the language from Portuguese to English while playing a level, the expected flow would be as follows:

1. Click on the pause button on the top right corner
2. Click on ”Main menu”
3. Click on the settings button on the bottom left corner
4. Click on the language button until ”English” appears

The children demonstrated confusion and none of the groups was not able to achieve the objective. After getting feedback, the root was the lack of association that the pause button would take them to the main menu which would allow them to switch the language.

5.3.2.C Choose a different character

On this final request, the objective was to choose a different character than the one previously selected before playing the house mission and two main flows could have been taken. The first would be:

1. Click on the pause button on the top right corner
2. Click on ”Main menu”
3. Click on Play
4. Choose the first level
5. Click on the next button on the bottom right corner until the player choice scene appears

The second possible flow was:

1. Click on the pause button on the top right corner
2. Click on ”Main menu”
3. Click on Play
4. Choose the second level
5. Click on back button on the bottom left corner

Both groups were able to complete the task, choosing flow number one, as they did not remember that the second level has a previous button. While executing the task we could observe that the current implementation is not ideal and changes should be done so that player can easily switch the chosen character. This was also reinforced by observing the behaviour of other groups which did not perform this task and were composed by children from both genders, playing the game in turns.

5.4 Final Remarks

In this chapter, we evaluated and compared both Treme-Treme versions. The main objective was achieved as the children demonstrated to gain more knowledge about earthquakes preparedness after playing the game. The objective of the newest additions was also achieved. Gamification proved to be the key element to provide higher engagement between the children and the game. To get all possible stars, they replayed the levels where their performance was lower over and over again. Quiz levels were also good to foster knowledge, generating healthy debates to choose the correct answer. The interviewed professors also gave good feedback and considered Treme-Treme's second version a good addition to the classroom when they addressed the subject of the earthquake.

However, this is still a tool under development, as some issues remain and should be fixed. As seen, most of the players choose the bed to hide from debris, clicking under this furniture. It turns out that the position where players provide as input is closer to a point in the middle of the stairs and the children do not understand why the character has this unexpected behaviour. Constructive criticism was taken to improve the game and will be addressed in the next chapter.

6

Conclusion

Contents

6.1 Conclusions

This project proved to require a greater effort than originally envisaged. Although the game has already been made available for several years, the whole code has become obsolete by using old libraries and programming languages that are no longer supported by Unity, making it impossible to work on what already existed. It was necessary to redo the game in full, without the possibility of reusing code that already existed. All the animations had to be also redone from scratch and the location of the objects was approximated of the original version. It was thus impossible to perform the third level on the available time.

Gamification was introduced as the game was rebuilt, which made it even more appealing and aroused much more curiosity in children. It was quite gratifying to see that the children loved the game and were able to learn from the work that was developed. One child from one of the test groups went so far as to say that he did not want to leave when one of the colleagues remembered that they had to deliver a report until noon, so the four left the room so sad and with heads down, after giving their feedback about the game. In addition, all the children said they would play the game again at home and it was exactly the intended reaction upon deciding to dedicate time to rebuild Treme-Treme. During development, I suffered a lot with bugs that took days to be fixed, which sometimes led me to think that maybe I should not have chosen this project. However, the look of those children playing something I did, made all the hard work worth. So, today, I know that I made the right choice.

The challenge of using a tool that I had never used or even heard about, the Godot Engine, was also a good choice compared to Unity. The number of resources available online, as well as the ease of working on any platform, to export on any platform, is one of its best features, taking into account that it is an opensource project, which depends on the community to be able to hire full-time developers. In my opinion, currently, Godot is undoubtedly the best game engine for 2D games. For the resources it has is fascinating to see that with work, dedication and liking for what is done, everything can be achieved.

Still, on Treme-Treme, I hope it continues to grow and has even more players than there are currently. As the site had no statistics associated with the number of downloads, it is impossible to know for sure how many children (and even adults) have already played it, but we realize it has already gone further than we ever expected. The first version had already in mind the internationalization and this new version follows the same way, with the addition of French and Spanish, two of the most spoken languages worldwide. In this way, more children will learn how to be prepared and how to proceed when one of nature's greatest dangers strikes, saving their lives and the ones around them.

It is noteworthy that after the evaluation the game was already in exhibition on Dia do Técnico, on May 23, 2019, in order to promote the game laboratory of the Instituto Superior Técnico, where it received excellent feedback from those who played it. The game was available to be played along with other games developed by the masters of games, being one of these the first version of Treme-Treme. The

activity lasted 3 hours and was visited by children and parents.

6.2 Future Work

Treme-Treme continues to be a work in progress. As the primary objective has been achieved - the code is not obsolete - the following changes can be performed to improve the game and deliver new knowledge to children all over the world:

- **Expand the game with new levels.** The conceptualisation of a tsunami level has already been performed and there is a working prototype, however, it was built using Unity, which means that the code cannot be reused. Nevertheless, as the rest of the game was rebuilt, the development should be smooth and faster than before.

- **Possibility to explore new ways to engage the players.** This topic is directly connected to the previous. With the introduction of the third level, perhaps the kit mission should not both require players to fulfil all available slots and take essential items to survive a natural disaster. Players could take what they felt to be the best items and then, on the third level, realise that the decision previously made was not the best. Then they would replay the kit mission and take the correct items to continue unlocking levels. By doing this, children would discover the reason why, for example, canned food is important than a teddy bear on their own and not because they are forced to do it by the game.

- **Increase the number of questions per level.** During experiments, children replayed the quiz levels several times to win all stars. As the pool of questions on both levels 0 and 1 match the number of available questions, only randomly selecting the order that they are shown is not the ideal procedure. The pool of questions should be bigger to transmit even more knowledge and to keep the player focused.

- **Adapt the game to child with disabilities.** As we learned children with special needs play Treme-Treme in the classroom to learn about earthquake preparedness. Having this in mind, the game could also be done thinking in, for example, people who suffer from colour blindness.

- **Create one or more instructional videos and tutorials.** Videos are a great way to pass on complicated concepts to kids. To this end, videos with the best practices that should be taken before, during and after an earthquake, which would be reproduced at the beginning of each level to complement it. In addition, they could be played when the player dies, giving hints on alternate ways to pass the level.

- **Increase the number of questions per level.** During experiments, children replayed the quiz levels several times to win all stars. As the pool of questions on both levels 0 and 1 match the number of available questions, only randomly selecting the order that they are shown is not the ideal procedure. The pool of questions should be bigger to transmit even more knowledge and to keep the player focused.
- **Improve the multi language support.** New strings were introduced and Italian continue to be a language not fully translated. New languages, like Japanese, could also be introduced to reach even more players worldwide.
- **Assessment of the passage of knowledge** An assessment can be made to understand the impact of the game on children's knowledge. As this is a serious game, we want to know if the behaviours have changed after they have been played, so this can be assessed using a drill to test the children in an event as close to reality as possible. Do they shelter in the right places during the earthquake? Or is it trying to run away from home is subject to something falling on them? Does the switchboard turn off? And the gas? And they will remember to take the emergency kit with them?

Bibliography

[1] G. M. Molchan, “Earthquake prediction as a decision-making problem,” *Pure and Applied Geophysics*, vol. 149, no. 1, pp. 233–247, 1997.

[2] R. J. Geller, “Earthquake prediction: a critical review,” *Geophysical Journal International*, vol. 131, no. 3, pp. 425–450, 1997.

[3] R. E. Wallace, J. F. Davis, and K. C. McNally, “Terms for expressing earthquake potential, prediction, and probability,” *Bulletin of the Seismological Society of America*, vol. 74, no. 5, pp. 1819–1825, 1984.

[4] Pedro Barreto. (2014) ”Treme-treme - A serious game to teach children earthquake preparedness”, IST. Last Accessed on 20-June-2019. [Online]. Available: https://fenix.tecnico.ulisboa.pt/downloadFile/563345090413387/dissertacao.pdf

[5] R. Lovreglio, V. Gonzalez, R. Amor, M. Spearpoint, J. Thomas, M. Trotter, and R. Sacks, “The need for enhancing earthquake evacuee safety by using virtual reality serious games,” in *Lean & Computing in Construction Congress*, 2017, pp. 1–9.

[6] R. Lovreglio, V. Gonzalez, Z. Feng, R. Amor, M. Spearpoint, J. Thomas, M. Trotter, and R. Sacks, “Prototyping virtual reality serious games for building earthquake preparedness: The auckland city hospital case study,” *Advanced Engineering Informatics*, vol. 38, pp. 670–682, 2018.

[7] J. Cain and P. Piascik, “Are serious games a good strategy for pharmacy education?” *American journal of pharmaceutical education*, vol. 79, no. 4, p. 47, 2015.

[8] J. W. Burke, M. McNeill, D. K. Charles, P. J. Morrow, J. H. Crosbie, and S. M. McDonough, “Optimising engagement for stroke rehabilitation using serious games,” *The Visual Computer*, vol. 25, no. 12, p. 1085, 2009.

[9] A. J. Stapleton, “Serious games: Serious opportunities,” in *Australian Game Developers Conference, Academic Summit, Melbourne*, 2004.

[10] H. Mouaheb, A. Fahli, M. Moussetad, and S. Eljamali, “The serious game: What educational benefits?” *Procedia - Social and Behavioral Sciences*, vol. 46, pp. 5502 – 5508, 2012, 4th WORLD CONFERENCE ON EDUCATIONAL SCIENCES (WCES-2012) 02-05 February 2012 Barcelona, Spain. Last Accessed on 29-May-2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187704281202201X

[11] J. Alvarez, “Du jeu vidéo au serious game: approches culturelle, pragmatique et formelle,” Ph.D. dissertation, Toulouse 2, 2007.

[12] E. A. Amato, “Vers une instrumentalisation communicationnelle des jeux vidéo: quelles formes de séduction idéologique ou publicitaire?” in *Colloque international EUTIC 2007:” Enjeux et Usages des TIC”, 7-10 novembre 2007*, 2007.

[13] T. Wada, Y. Nemoto, S. Shimamura, T. Fujita, T. Mizuno, T. Sohtome, K. Kamiyama, T. Morita, and S. Igarashi, “Effects of the nuclear disaster on marine products in fukushima,” *Journal of environmental radioactivity*, vol. 124, pp. 246–254, 2013.

[14] K. Ochiai, S.-i. Hayama, S. Nakiri, S. Nakanishi, N. Ishii, T. Uno, T. Kato, F. Konno, Y. Kawamoto, S. Tsuchida *et al.*, “Low blood cell counts in wild japanese monkeys after the fukushima daiichi nuclear disaster,” *Scientific reports*, vol. 4, p. 5793, 2014.

[15] A. Priyadarshi, G. Dominguez, and M. H. Thiemens, “Evidence of neutron leakage at the fukushima nuclear plant from measurements of radioactive 35s in california,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 35, pp. 14 422–14 425, 2011.

[16] Michele M. Wood, Carol Asiala, Wayne Pennington. ”UPseis - an Educational site for budding seismologists”, Geological and Mining Engineering and Sciences, Michigan Technological University. Last Accessed on 29-May-2019. [Online]. Available: http://www.geo.mtu.edu/UPSeis/index.html

A

Code of Project

Listing A.1: Shader used for fade in/fade out effects

```
// for 2D rendering
shader_type canvas_item;
// result is just albedo (color). No lighting/shading happens in material
render_mode unshaded;

// variable for the effect, between the range 0 and 1 with steps of 0
uniform float cutoff : hint_range(0,1.0);
// used as albedo color, default white
uniform sampler2D filter : hint_albedo;

void fragment()
{
    COLOR = texture(SCREEN_TEXTURE, SCREEN_UV);
```

```
    vec4 tex = texture(filter, SCREEN_UV);
    if (tex.r < cutoff || cutoff > 0.999)
    {
        COLOR.rgb = vec3(0.0);
    }
}
```

B

Interview Questions

B.1 Some of the questions to students

1. Which version did you like the most?
2. Why did you like that one the most?
3. What should you always take before you leave home?
4. What should I put in an emergency kit?
5. What should you do when you feel an earthquake?
6. Did you play Treme-Treme again?
7. Did you have difficulty with the game controls?
8. If so, on what level did you feel the most difficulty?

B.2 Some of the questions to teachers

1. Which version did you like the most? Why?
2. Do you think the questions in the new version are appropriate?
3. Questions are optional for the game and are just a bonus to reward the player, do you think they should be mandatory to the game progress?
4. Would you use the game as a multimedia resource complementary to the teaching of earthquakes as well as behaviours to have?