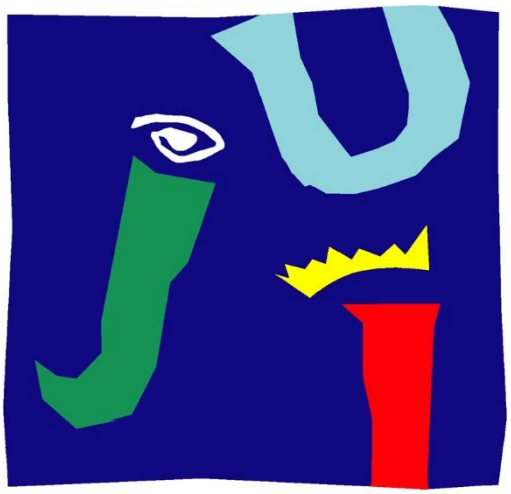

UNIVERSITAT JAUME·I

Final Degree Project's Technical Report

Game Design and Development Degree

STRESS TREATMENT THROUGH VIRTUAL REALITY

AUTHOR: NEREA ALAMAR PORTER

TUTOR: RAÚL MARÍN PRADES

Summary

This technical report presents the design and implementation details involved in the development of an interactive environment using the Unity3D game engine.
The main objective of this project is to decrease the user's stress level. To achieve this objective the project consists of a virtual reality application that includes different relaxation techniques like audios, images and graphic elements.
Moreover, the application obtains and uses user data from an external sensor, eHealth. These dates are indices of skin temperature and conductance and are used to calculate the stress level of the user during the virtual experience.

The application is for Android devices, using Cardboard glasses and a bluetooth gamepad to move inside the environment.
The virtual experience consists in an interactive environment in third-person in which the user can walk through the stage, exploring the scene while a relaxing background music is played and accesing to other scenes consisting in relaxation audio guides and relaxing places (taking advantage of the realism of virtual reality).

As said before, this project was developed using mainly Unity3D, although I also used software like MonoDeveloped and Visual Studio 2017 for developing the codebase, 3ds Max for modeling, animating and texturizing, Adobe Photoshop CS6 for texture customizing, Arduino for the eHealth Sensor implementation and Word 2016 for the creation of this report.

Keywords: Interactive environment, stress, Unity3D, relaxing

Index

Figure Index

Introduction

In this introductory section, we'll cover different important aspects of this project, from the context and motivation, it addresses the situation of society in terms of stress, the objectives I set myself up when facing this project, the planning and expected results, as well as the justification of how my project enhance the skills I learned during my years in UJI.

Context and Motivation

Stress is one of the most serious health problems today, which not only affects health workers, causing them physical or mental disability, but also employees of different areas, as well as government officials.

According to the World Health Organization stress is defined as the "set of physiological reactions that prepare the body for action."

The impact of stressful situations on the individual's physical and mental well-being may be substantial or extreme, such as suicide or cardiac arrhythmias. Behavioral changes that may occur due to increased levels of stress include states of nervousness, body decay, physical tension, and heart problems. [1]

Unfortunately, too much stress has become an increasingly common feature in contemporary life. When stressors unbalance the nervous system, relaxation techniques can help us to restore steady state by producing the relaxation response, a state of deep calmness that is the polar opposite of the stress response.

For many people to relax is synonymous with sitting in front of the TV without doing anything after a stressful day. But this does not help reduce the effects of stress. To combat stress effectively, we need to activate the body's natural relaxation response, which is achieved by practicing relaxation techniques such as deep breathing, meditation. [2]

There are different relaxation techniques, some more active or passive than others, to help people relax, but there is a lot of general ignorance on the subject. [3]

The aim of my project is to create an interactive environment that has different relaxation techniques. It is a way to get people to different relaxation techniques but in a more enjoyable or entertaining way, allowing them not to have to leave the technology aside but to use it to achieve this goal.

Objectives

When I decided to go ahead with this project, I set myself a series of goals:

- Create an application that uses different relaxation techniques so that it can be adapted to the user.
- The user enjoys a complete immersive virtual experience.
- The user feels more relaxed after using the application, and in turn learns and discovers different methods of relaxation that he can apply at the moment he needs them.

Planning

In this section, we'll go in detail about all the different tasks that I planned to do during the development of the project.

The tasks planned to be implemented were:

Task 1: Design the distribution of scenes and elements (10 h):
Decide the scenes, and concrete scene elements that model, texturize, animate and implement.

Task 2: Design, model, texturise and animate character (40 h):
It includes both the initial sketches and modeling, texturizing and animation in 3ds Max, as well as the research and use of specific tools for this, with the aim of improving the results.

Task 3: Movement of the character (5 h):
Implementation in Unity3D of the movement of the player and the position of the camera.

Task 4: Use and implementation of virtual reality (20 h):
It includes both research on the tool and its proper use and implementation.

Task 5: Design, model and texturing main scene (40 h):
It includes both the initial sketches and the modeling, texturizing and animation in 3ds Max, organization and distribution of the geometry, as well as the research and use of specific tools for this, with the aim of improving the results.

Task 6: Design, model and texturing the other scenes (30 h):

It includes both the initial sketches and the modeling, texturizing and animation in 3ds Max, organization and distribution of the geometry, as well as the research and use of specific tools for this, with the aim of improving the results.

Task 7: Implementation of the scenes (10 h):

Everything related to the Unity3D implementation of the scenes, both user interaction with the environment and corresponding animations within Unity3D.

Task 8: Change of scenes and saving data between scenes (10 h):

Unity3D implementation of the change of scenes, saving the data shared between scenes, and design and implementation of a load screen between scenes.

Task 9: Implementation of the code needed to operate the sensor (20 h):

It includes both the code for reading data and for the use and later visualization of this data in the application.

Task 10: Implement database access to store received data (20 h):

Creation of web page where the database is stored and code that allows access to it and send data.

Task 11: Write and record texts of relaxation techniques (10 h):

Research, writing and later recording of audios using relaxation techniques.

Task 12: Menus and user interface (5 h):

Design and implementation of menus that allow the user to move between scenes, as well as user interface, which includes the on-screen display of the data received by the sensor.

Task 13: Effects and details to give realism (5 h):

Once the environments are created, add extra effects and details to enhance the look and help give realism.

Task 14: Test and make changes based on results (30 h)

Test the application on the desired device by checking that everything works properly and without errors.

Task 15: Final report (45 h):

Research, documentation and drafting work on all project development.

	Febrero	*Marzo*	*Abril*	*Mayo*	*Junio*
Task 1	10 h				
Task 2	20 h	20 h			
Task 3		5 h			
Task 4		20 h			
Task 5		20 h	20 h		
Task 6			30 h		
Task 7			10 h		
Task 8			10 h		
Task 9				20 h	
Task 10				20 h	
Task 11				20 h	
Task 12				5 h	
Task 13					5 h
Task 14					30 h
Task 15					45 h

Figure 1: Project's initial planning

Expected results

With this project the user is expected to significantly reduce their levels of stress or anxiety through an interactive experience, as well as superficially introducing different relaxation techniques, so that the user is able to use them on their own account. It is also expected to make users aware of the importance of relaxation, and thus gradually be able to turn it into a healthy habit in their daily life.

Justification

In this section, the relation between the proposed project and the courses taught in the degree will be stated.

VJ1212 – Graphic Expression & VJ1216 – 3D design & VJ1226 – Character design and animation

These are the three subjects present in the degree related with 3D design and animation. The learning outcomes achieved in these subjects were:

- *E02 - Understand the different methods of three-dimensional representation on a flat surface.*
- *E04 - Use three-dimensional modeling applications for video games.*
- *E04, IR07, IR13 - Explain the principles that allow the definition of three-dimensional graphic elements.*
- *E06 - Explain the basics of computer animation.*
- *E06 - Explain the principles of character creation.*
- *E06, IR07, IR13 - Use character modeling and animation applications in the context of video games.*

VJ1204 – Artistic Expression & VJ1223 – Video Game Art

The learning outcomes achieved in these subjects were:

- *E01 - Master the different graphic tools and their expressive resources.*
- *E01 - Produce freehand sketches of the elements of video games.*
- *E09, G09 - Control the graphical tools to create representations with high level of finishes.*
- *E09, G10 - Create environments with definition of lights, shadows, textures and environmental effects.*
- *E09, G10 - Create characters characterized and textured.*

VJ1227 – Game Engines

Some of the subject's concepts were applied to Unity, which has helped me and qualified to carry out this project. Autodidactic capacity, which was enhanced through the subject of Game Engines, is an essential generic skill which has been specifically applied in the learning process of Arduino or virtual reality.

The learning outcomes achieved in these subjects were:

- *E08, IR06, IR07 - Explain the basic principles of creating interactive images.*
- *E12, G09 - Evaluate the technical characteristics of game engines as technology for the creation of video games.*
- *E12, IR07, IR14 - Use game engines to create video games.*

VJ1208 – Programming II

In this subject are provided knowledge of programming in C # language that are necessary for the creation of this project.

The learning outcomes achieved in these subjects were:

- *IB04 - Implement libraries that allow reuse of code in simple problems.*
- *IB04 - Use existing libraries in program development.*
- *IB04 - Use standard implementations of the following data structures: stack, co-la, linked list, dictionary.*

Virtual reality

Virtual reality is the term used to describe a three-dimensional, computer generated environment which can be explored and interacted with by a person. That person becomes part of this virtual world or is immersed within this environment and whilst there, is able to manipulate objects or perform a series of actions.

Although we talk about a few historical early forms of virtual reality elsewhere on the site, today virtual reality is usually implemented using computer technology. There are a range of systems that are used for this purpose, such as headsets, omni-directional treadmills and special gloves. These are used to actually stimulate our senses together in order to create the illusion of reality.

This is more difficult than it sounds, since our senses and brains are evolved to provide us with a finely synchronized and mediated experience. If anything is even a little off we can usually tell. This is where you'll hear terms such asimmersiveness and realism enter the conversation. These issues that divide convincing or enjoyable virtual reality experiences from jarring or unpleasant ones are partly technical and partly conceptual. Virtual reality technology needs to take our physiology into account.
For example, the human visual field does not look like a video frame. We have (more or less) 180 degrees of vision and although you are not always consciously aware of your peripheral vision, if it were gone you'd notice. Similarly when what your eyes and the vestibular system in your ears tell you are in conflict it can cause motion sickness. Which is what happens to some people on boats or when they read while in a car.

If an implementation of virtual reality manages to get the combination of hardware, software and sensory synchronicity just right it achieves something known as a sense of presence. Where the subject really feels like they are present in that environment. [4]

Why virtual reality is important for my project

Virtual reality allows the user to completely immerse himself in a different world. Disconnect from the real world full of stress, and forget about all the problems.
Using virtual reality in my project helps people relax, feel inside the environment, visualize and walk through the scenes in the first person.

“Some virtual reality experiences can help control breathing, even there are devices that can be used in situations of tension with images and audio that help mitigate the adverse effects on the body due to stress.” - Marco DeMiroz, one of the key virtual reality investors in Silicon Valley.

There are studies that advocate that the visualization of relaxing landscapes such as a beach coupled with environmental sounds such as sea waves help to focus and reduce stress. [5]
It is also proven that virtual reality can help treat both stress and anxiety situations. It’s already working on similar projects to address these problems, it is a matter of time for virtual reality to become a commonly used therapeutic method. [6][7][8]

Types of Virtual Reality Devices

On the one hand are the devices of higher quality and more expensive, which are used in a computer that meets specific requirements to operate. These are:

HTC Vive

The HTC Vive [9] is a virtual reality device for computer made by HTC and Valve. This device is designed to use space in a room and immerse itself in a virtual world in which the user is allowed to walk and use controllers to interact with virtual objects.
The Vive has a regresh rate of 90 Hz. The device uses two screens, one per eye, each having a display resolution of 1080x1200. The device uses 70 infrared sensors: 32 on the headset and 19 for each controller. In addition, it also uses a MEMS (Microelectromechanical systems) gyroscope and accelerometer.
It is said to operate in a 15-by-15-foot (4.6 by 4.6 m) tracking space if used with both "Light-house" base stations that track the user's movement with sub-millimeter precision. The Light-house system uses simple photosensors on any object that needs to be captured; to avoid occlusion problems this is combined with two lighthouse stations that sweep structured light lasers within a space.

The front-facing camera allows the software to identify any moving or static objects in a room; this functionality can be used as part of a "Chaperone" safety system, which will automatically display a feed from the camera to the user to safely guide users from obstacles.

Figure 2: HTC Vive devices

Oculus Rift

The Oculus Rift [10] is a virtual reality headset developed and manufactured by Oculus VR, a division of Facebook Inc.

The Rift has a stereoscopic OLED display, 1080x1200 resolution per eye, a 90 Hz refresh rate, and 110° field of view. It has integrated headphones which provide a 3D audio effect, rotational and positional tracking. The positional tracking system, called "Constellation", is performed by a USB stationary infrared sensor that is picking up light that is emitted by IR LEDs that are integrated into the head-mounted display. The sensor normally sits on the user's desk. This

creates 3D space, allowing for the user to use the Rift while sitting, standing, or walking around the same room.

Figure 354: Oculus rift devices

On the other hand we have two alternatives for portable devices (smartphones) that allow us to enjoy a virtual reality experience cheaper and affordable.

Samsung Gear VR

The Samsung Gear VR [11] is a mobile virtual reality headset developed by Samsung Electronics, in collaboration with Oculus, and manufactured by Samsung.
When in use, a compatible Samsung Galaxy devide acts as the headset's display and processor, while the Gear VR unity itself acts as the controller, which contains the field of view, as

well as a custom inertial measurement unit, or IMU, for rotational tracking, which connects to the smartphone via micro-USB. The Gear VR headset also includes a touchpad and back button on the side, as well as a proximity sensor to detect when the headset is on.
The quality is very good despite being for a portable platform.

Figure 4: Samsung Gear VR

Google Cardboard

Google Cardboard [12] is a virtual reality (VR) platform developed by Google for use with a head mount for a Smartphone. Named for its fold-out cardboard viewer, the plaform is intended as a low-cost system to encourage interest and development in VR applications. Users can either build their own viewer from simple, low-cost components using specifications published by Google, or purchase a pre-manufactured one. To use the platform, users run Cardboard-compatible applications on their phone, place the phone into the back of the viewer, and view content through the lenses.

Figure 5: Cardboard glasses

Figure 6: Plastic Cardboard devices

Which one do I use and why

After spinning around the matter, I decided to use the simplest version of virtual reality, Google Cardboard.

It is the easiest and simplest, and most open to the public. Since there is currently little people who have or can afford to have one of the other virtual reality devices, and this device is affordable for everyone. [13]

On the other hand, it has been the easiest device to perform tests, since I tried to use the Oculus DK2 glasses and I could not make them work because my computer did not meet the minimum requirements.

The application is developed for Google Cardboard, but it is not a difficult task to use it for Oculus.

To convert a Cardboard application to an Oculus application, simply change a line of code from the GazeInputModule script included inside the Cardboard asset to correctly detect the position of the pointer and change the Cardboard mode from double screen to single screen:

Line of code used to export a Cardboard application:

```
pointerData.position = new Vec-
tor2(hotspot.x * Screen.width, hotspot.y * Screen.height);
```

Line of code used to export an application to Oculus:

```
pointerData.position = new Vector2(VRSettings.eyeTextureWidth/2, );
```

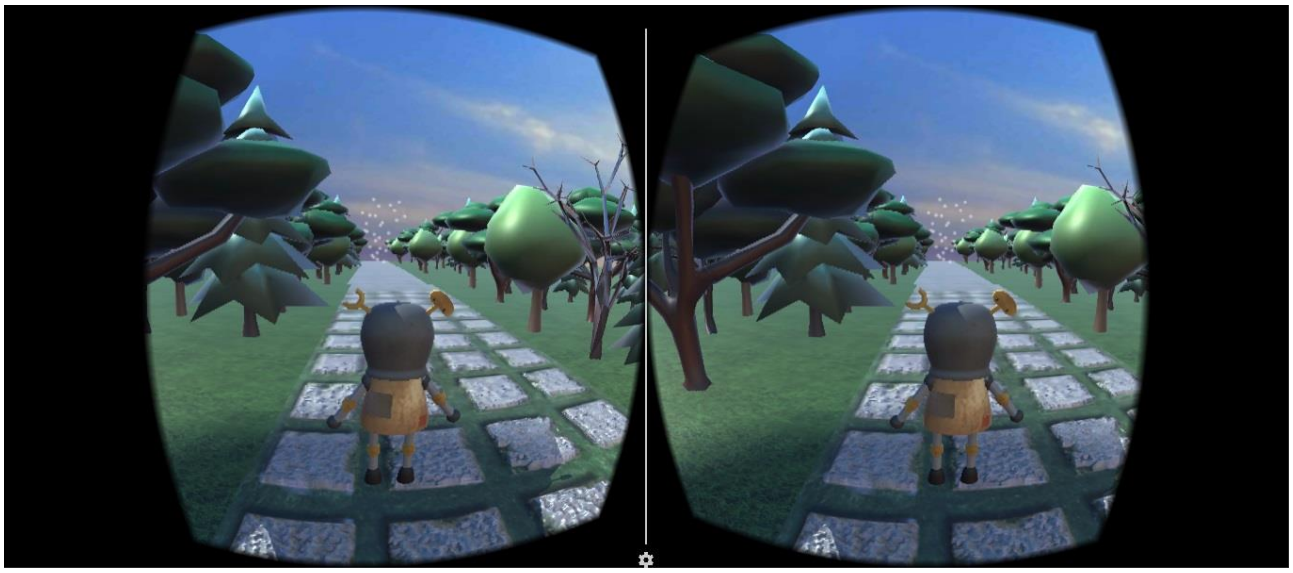

Figure 55: Cardboard double screen

Figure 8: Cardboard single screen

It is true that the quality is worse in Google Cardboard than in the rest of applications but it is still good and I had very much in mind while carrying out the project the reduction of computational information and that the application can execute correctly in any platform without having to slow down too much .

Specifically I have used the Woxter Neo VR 1 device, which includes virtual reality glasses and Bluetooth control.
The Bluetooth control is necessary and is also used, since to move around the scene it's needed a joystick.

Figure 9: Woxter Neo VR 1 device

Biofeedback

Biofeedback is a technique that is used to control the physiological functions of the human organism, through the use of a feedback system that informs the subject of the state of the function to be controlled on a voluntary basis. [14]

It is a process that allows a person to learn how to change physiological activity for the purpose of improving health and performance. Accurate instruments measure physiological activity such as brain waves, cardiac function, respiration, muscle activity and skin temperature. These instruments quickly and accurately 'feed back' information to the user. The presentation of this information - often in combination with changes in thinking, emotions and behavior - supports desired physiological changes. Over time, these changes can be sustained without the continuous use of an instrument.

Now biofeedback is being used in sport (to reduce pre-competitive anxiety, measured through heart rate), therapeutic support in cancer treatments, performance optimization, etc. [15]

E-Health Sensor

The e-Health Sensor Shield V2.0 allows Arduino and Raspberry Pi users to perform biometric and medical applications where body monitoring is needed by using 10 different sensors: pulse, oxygen in blood (SPO2), airflow (breathing), body temperature, electrocardiogram (ECG), glucometer, galvanic skin response (GSR - sweating), blood pressure (sphygmomanometer), patient position (accelerometer) and muscle/eletromyography sensor (EMG).

This information can be used to monitor in real time the state of a patient or to get sensitive data in order to be subsequently analysed for medical diagnosis. Biometric information gathered can be wirelessly sent using any of the 6 connectivity options available: Wi-Fi, 3G, GPRS, Bluetooth, 802.15.4 and ZigBee depending on the application.

If real time image diagnosis is needed a camera can be attached to the 3G module in order to send photos and videos of the patient to a medical diagnosis center.

Data can be sent to the Cloud in order to perform permanent storage or visualized in real time by sending the data directly to a laptop or Smartphone. iPhone and Android applications have been designed in order to easily see the patient's information. [16]

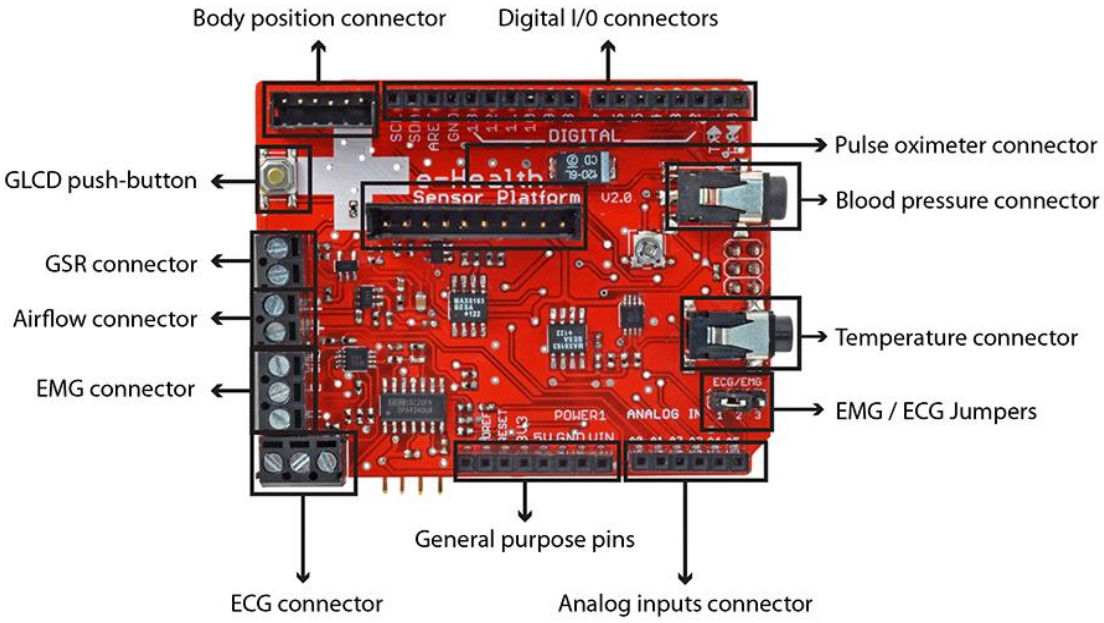

Figure 10: E-Health Sensor Shield V2.0

The e-Health library

The e-health Sensor Platform counts with a C++ library that lets you read easily all the sensors and send the information by using any of the available radio interfaces. This library offers a simple-to-use open source system.

The eHealth sensor platform includes a high level library functions for an easy manage of the board. All the files needed are in two separated folders, “eHealth” and “PinChangeInt”.

How to Use Data to Measure Stress

Electrodermal activity

The electrodermal activity (EDA) is one of the variables with the greatest psychophysiological tradition. It is proven that by passing a small electric current through the participant through two electrodes placed on the surface of the skin, its electrical resistance can change according to psychological variables.

The biological basis of the electrical activity of the skin is found in the activity of the eccrine sweat glands, which discharge sweat to the outside of the body through an excretory tube. The eccrine suduripar glands are distributed along the surface of the whole body, but their density is greater in the palm of the hand, a preferred place for EDA registration as a psychophysiological variable.

These glands respond fundamentally to two types of conditions: thermal and psychological. The second ones are the predominant ones under normal conditions and are the ones that give psychophysiological value to this type of variable.

The most commonly used electrodermographic procedures in psychophysiology are those of dermal resistance and conductance, based on the exogenous application of a direct current. Both have to do with the same physical factor, the degree of difficulty with which electricity crosses the skin. This difficulty depends, among other things, on sweating. Sweat is basically an electrolyte that impregnates the stratum corneum (dry and insulating layer of the epidermis) and fills the secretory ducts of the eccrine glands facilitates a pathway from the outside of the skin to the hypodermis.

What sensors do I use

In this project is taken into account and is going to focus on skin conductance data, the more sweating the more conductance.

And the data will be contrasted with the temperature data of the person.

It is verified that at higher temperature and conductance of the skin, more stress. [17]

Process

For the endothermic activity calculation the participant should wash his hands only with water, rubbing the palms well. Any product could alter the conductive properties of the skin.

Adhere the two electrodes to the participant's thenar and hypothenar eminences (usually in the non-dominant hand) and hook the cables.

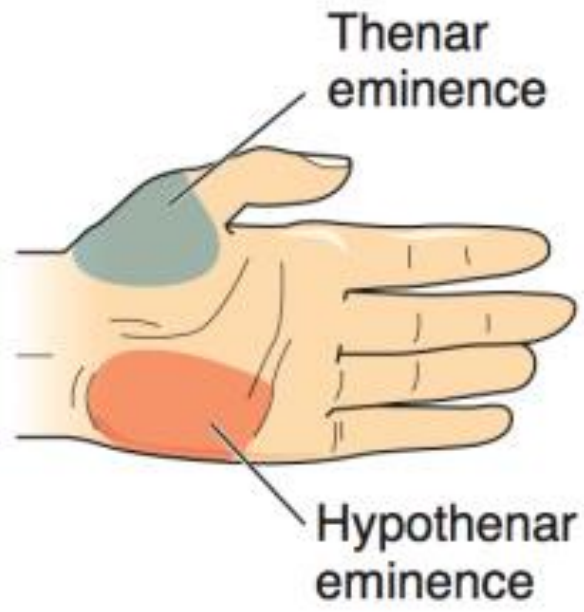

Figure 56: Thenar and hypothenar eminences

For taking measures of temperature, make contact between the metallic part and the skin using a piece of adhesive tape to hold the sensor attached to the skin.

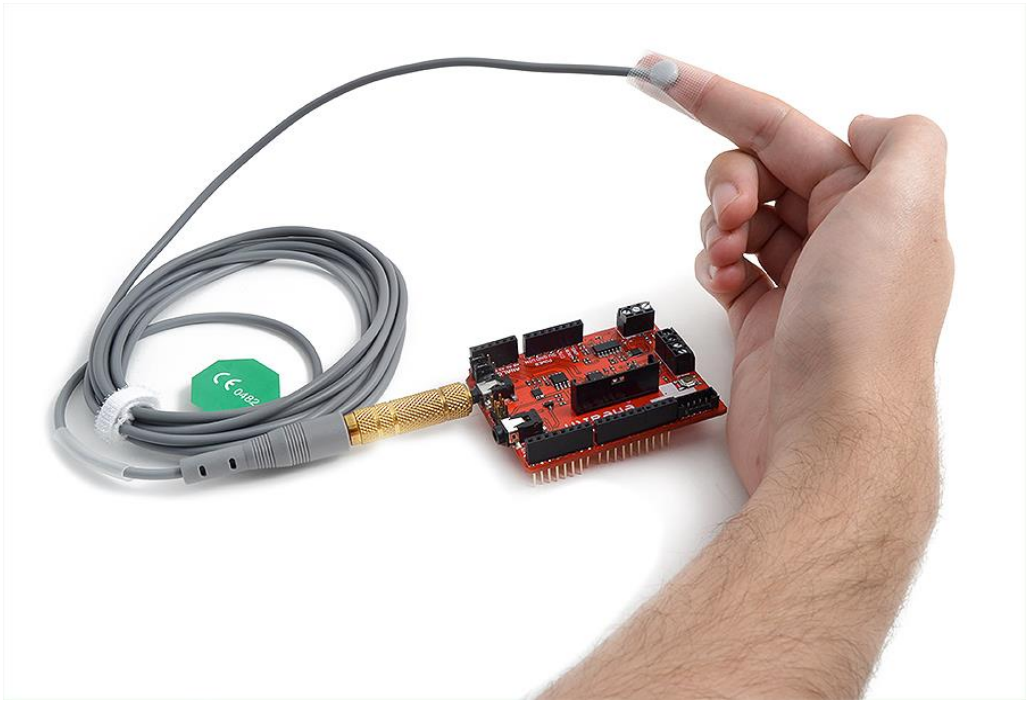

Figure 1257: Temperature calculation

Relaxation

The practice of relaxation is fundamental to face a multitude of circumstances and personal situations of daily life. On the one hand, it helps us to deal with the anxiety states that generate the daily task, be it pressures for work, noise, traffic of the cities, personal problems, stress. On the other hand, it relieves and prevents physical situations such as headaches, general indisposition, hypertension, stomach pains, discomfort of any kind ...

The practice of relaxation is recommendable by itself, there are no special causes to perform it, it may have a lot to do with the quality that we can give to our lives, in the sense that it is preventive, it is a form of hygiene and education.

Relaxation tries to harmonize corporal tendencies with psychic ones. That is, it is based on the principle that through psychological training, body awareness, from mental observation, we harmonize the tension levels of the body, so that we meet again with a body in harmony.

What is relaxation

Relaxation is fundamentally a physical and psychic state that is radically opposed to any state of tension. The state of relaxation is a subjective experience of calm, of hypoactivity, or in other words, this experience is defined by the lack of activity that seeks the most effective rest.

There are different methods of relaxation, but in this project we will use two methods in particular: autogenic training and visualization of relaxing spaces.

Autogenic training

Autogenic training is one of the most well-known classic techniques of relaxation and one of the most used. It consists mainly of a series of sentences elaborated in order to induce states of relaxation through suggestions on feelings of heat, weight, freshness, etc. And passive concentration in respiration.

In each of the exercises, the subject should only think about the formula that will be indicated, which will be different depending on the exercise you are performing. After concentrating on each of the different phrases of this training, you will gradually feel the sensation sought.

The method can be divided into two cycles:

The lower cycle

It consists of six exercises that must be dominated in order to move to the next cycle (upper cycle).

- The attitude towards relaxation and the atmosphere.
- Weight experience.
- Cardiac experience.
- Breathing experience.
- Warm experience.
- Head experience.

Each of these experiences influence different physiological levels, on the muscular system, the respiratory system, the heart, the vascular system, the abdomen, the head.

The upper cycle

In order to access the effectiveness of the upper cycle, the experiences of the lower cycle must be dominated, which is a sign of the great difficulty of achieving an effective and rapid relaxation.

The top grade is a self-concentrative relaxation that proceeds to other deepening exercises such as:

- The look towards the center of the forehead.
- Meeting the personal color.
- Other colors visualization.
- Objectes visualization.
- Encounter of personal feeling.
- Other people visualization.

[18] [19]

During this project, only the relaxation techniques of the lower cycle will be considered, since it is designed to introduce simple relaxation techniques to people with a good experience in the subject.

Visualization of relaxing spaces

Visualization techniques provide positive and pleasant images that allow us to live in relaxation experiences of happiness. They are perhaps the simplest and most attractive exercises to perform. Always after the relaxation exercises we can see how we seem to be involved in a cloud of positivity, calm and joy.

We can understand visualization as a technique of attraction to our mind of relaxing images, some can be something as simple as a sunny day, a landscape in spring, or anything complex we want.

Choose an image, a scene, any place that is very pleasant and relaxing. For example, being on a beach with the sound of the sea, in a forest by a river…

Feel that you are within that image, feel part of it and release all the tensions.

Then remain calm for a few minutes, and gradually slowly open the consciousness to reality, to the room, to the sensations of the body.

Methods of relaxation used

Autogenic lower cycle training

The attitude towars relaxation and the environment

Just 5 minutes of audio where it helps to disconnect and relax taking into account the breathing and the sensations of the body.

Weight experience

10-minute audio that focuses on imagining and feeling experiences weight in the body, slowly relaxing all areas of the body.

Warm experience

10-minute audio that focuses on imagining and feeling experiences the body's warm, slowly relaxing all areas of the body.

In the Appendix section are the transcripts of relaxation audios.

Visualization of relaxing spaces

Beach environment

An environment with a beach where the user can get carried away and rest next to the sound of the waves, the beach is one of the most relaxing and most used environments for this type of techniques.

Figure 58: Beach environment

Boat trip

A tour of the water in which the user can visualize a pleasant environment ro-deed of nature, accompanied by sounds of water and relaxing music.

Water is a very used and very efficient element to help relaxation.

Figure 14: Boat trip

Art

The art is a very important part of the project, because it depends on that, if the player is feeling relaxed, inmersed in the virtual world, and enjoys his virtual experience. It is important that the environment is pleasant, as well as not having many polygons so the application can execute correctly.
In virtual reality, if an application does not work quickly and optimally can make the player dizzy, so it is very important to design a nice but inexpensive environment.
One of the tasks that slows the application down is rendering, so it's so important to reduce the number of polygons to render.

What I have designed

Character

It is a robot, modelled, animated and textured in 3ds Max. The textures edited and created in Photoshop to give them more realism and metallic appearance.

Modeling

All pieces are modelled separately for the subsequent animation. I have used mostly extruded cylinders to model them.

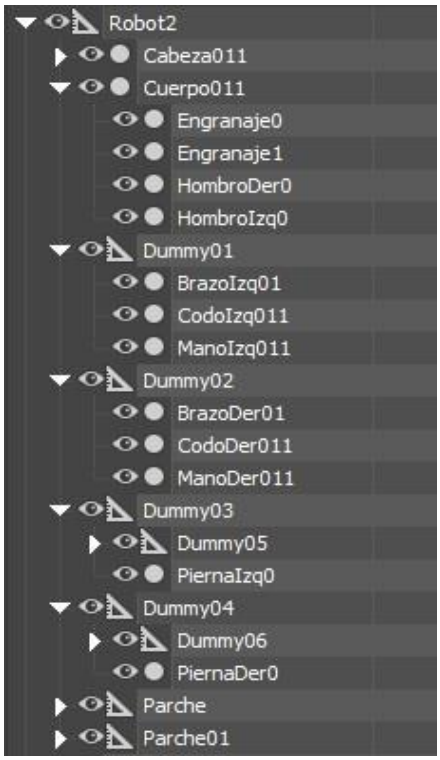

Figure 15: Character geometry

Animation

To animate it, I used 3ds Max, I didn't use a skeleton, because in a robot wasn't realistic an animation with a skeleton because it's not formed by organic material.

I did the animation manually frame by frame using dummies. Dummies are invisible objects that allow the geometry to be grouped in parts which has made the animation much easier for me.

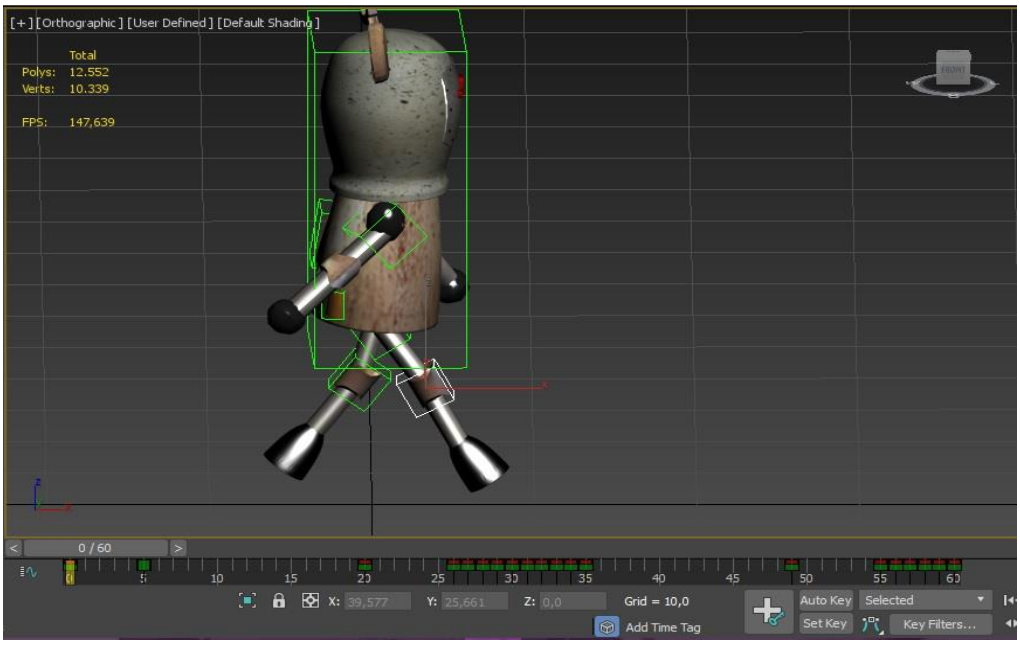

Figure 16: Animation with dummies

During the process of animating the character, I've been moving one by one the dummies to generate the animation of breathing and walking. It has been the most effective way to animate it since I was looking for sudden movements and little fluid to give more credibility to the character.

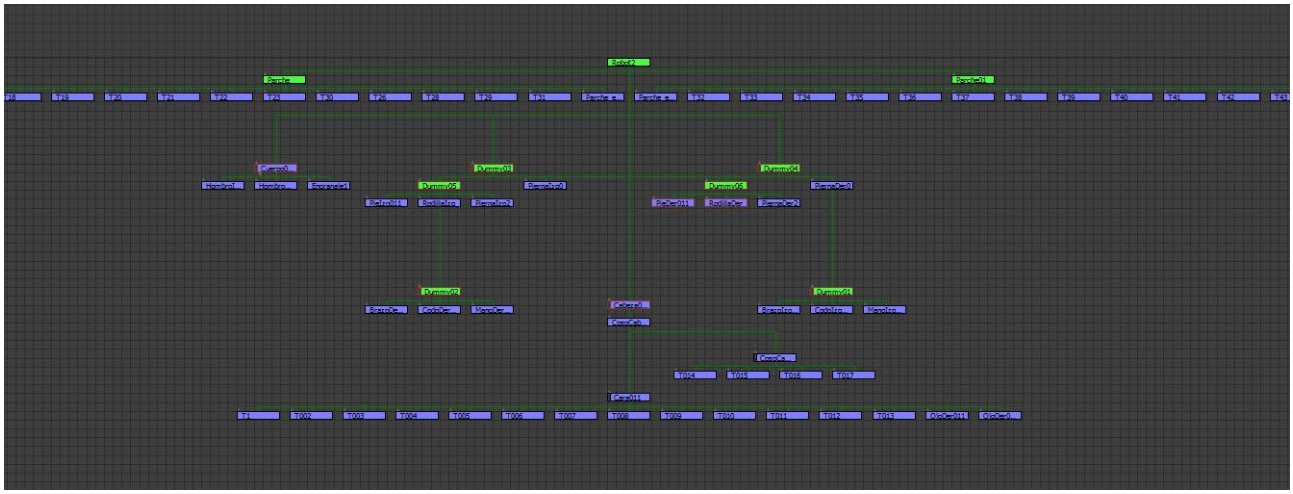

Figure 17: Hierarchy of character geometry with dummies

Textures

I used Photoshop CS6 to create, edit and adjust the textures, so that it could give a more realistic appearance to the character, giving it a metallic look.

I have used texture maps for most of the character textures. In order to apply these textures, I have used the 3ds Max UVW Map tool by modifying the mapping, length, width, and U, V and W tiles.

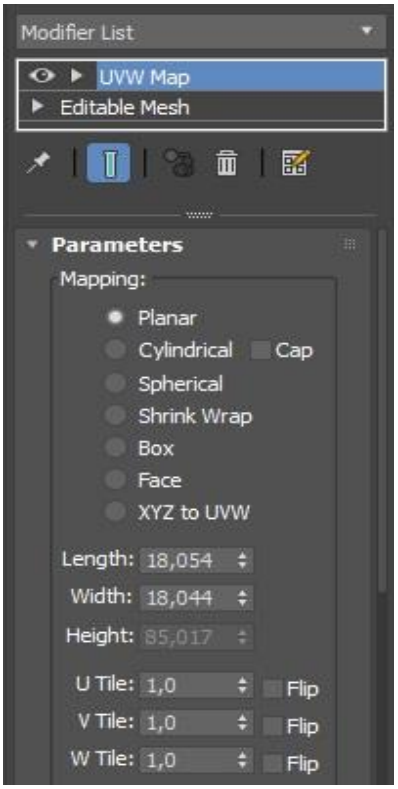

Figure 18: UVW Map tool

Thanks to this tool I have been able to perfectly couple the texture to each part of the character's body, obtaining a very good and realistic result.

Figure 19: Textured character

Scenes

There are 4 scenes designed and developed by me that includes art (not including scenes of relaxing audios).

For the scene elements modeling, some are completely made by me, starting from 0, also other models are made starting from the base of an existing model of the web site archive3d.org, editing and modifying it to achieve a good look.

In order to design the elements I have taken into account graphic optimization techniques, since, as mentioned before, for virtual reality applications it is very important that it works quickly and optimally to avoid problems.

To reduce the number of polygons of the geometry I used the ProOptimize tool of 3ds Max, thanks to this I converted a scene that could perfectly have 1.000.000 polygons in one of 100,000.

Figure 20: 3D element with 23124 polygons

Figure 21: 3D element with 4608 polygons

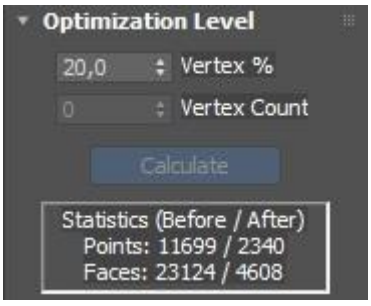

Figure 59: ProOptimizer tool parameters

Textures

To improve the visual quality of geometry as well as give realism and reduce the unnecessary calculation of geometry, lights or shadows, I have used different types of textures depending on the elements of the scenes.

Baked textures

On the one hand, I've put together different objects, reducing the number of objects to be rendered using the 3ds Max Attach tool. I have also used the technique of baked textures, that is, create a unique texture for the whole set, which also has already calculated lighting and shadows. This avoids an extra shade calculation cost in Unity, which greatly helps to improve the performance of the application.
To get a bake texture of an object, there is a tool in 3ds Max that calculates it, Render to texture. The first step is to open the Render to texture menu with the desired object selected.

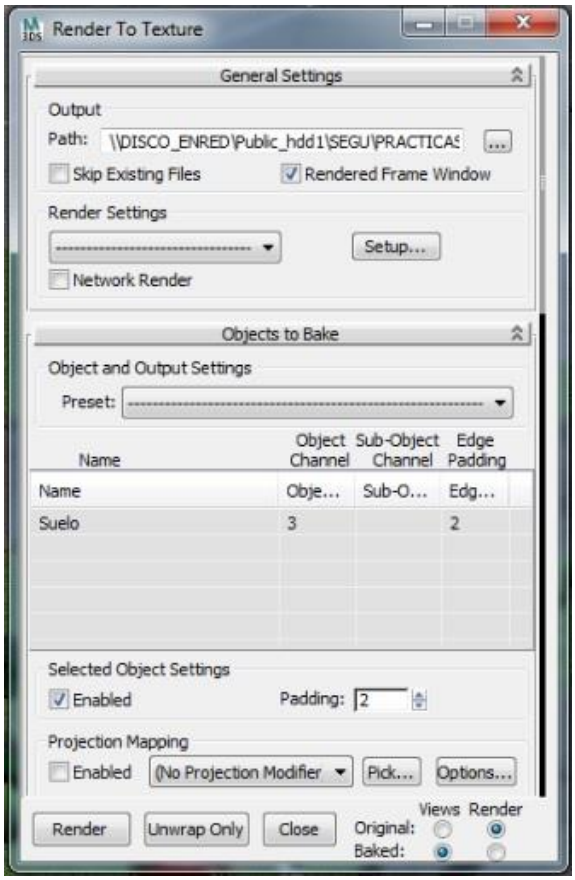

Figure 23: Generating baked textures part 1

Then, in the Output section, click on Add and select Complete Map, which consists of a map of both the texture and the lights and shadows projected by the scene. And then in Target Map Slot select Diffuse color, to display the color of the texture.

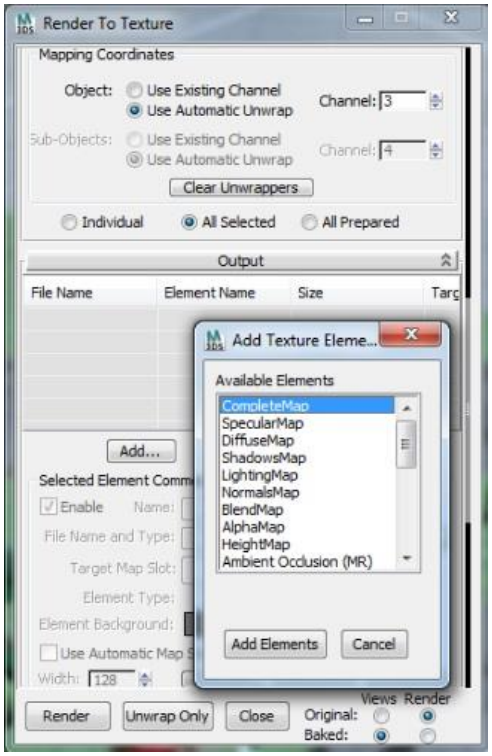

Figure 2460: Generating baked textures part 2

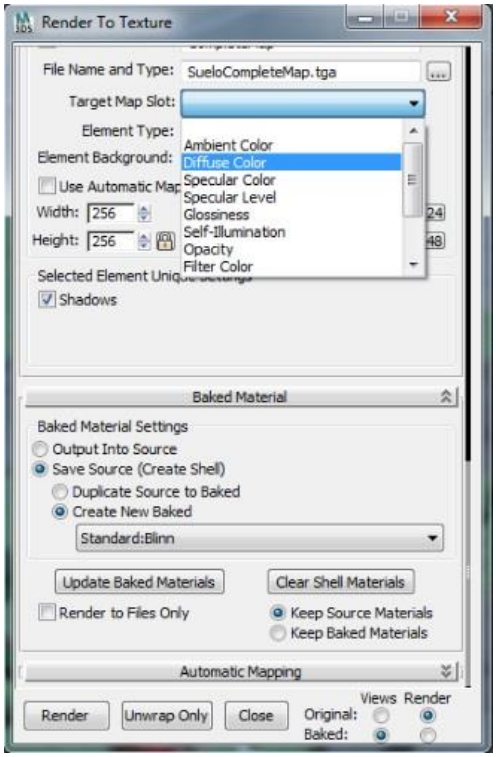

Figure 2561: Generating baked textures part 3

Finally click on Render and wait while the UV map is calculated and then export the texture.

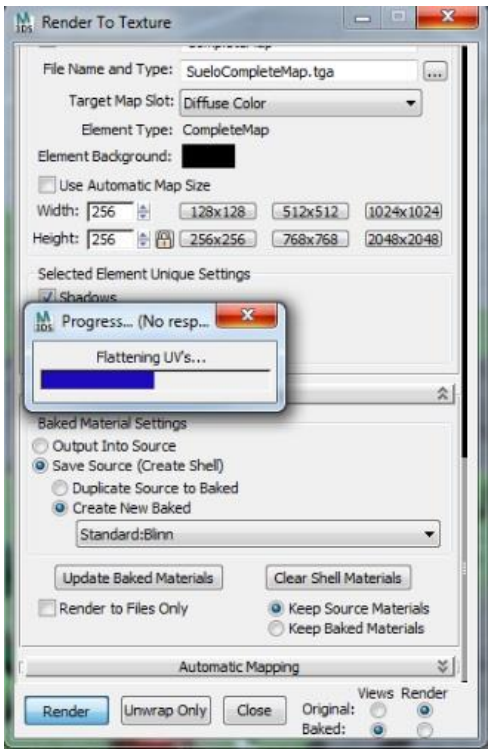

Figure 26: Generating baked textures part 1

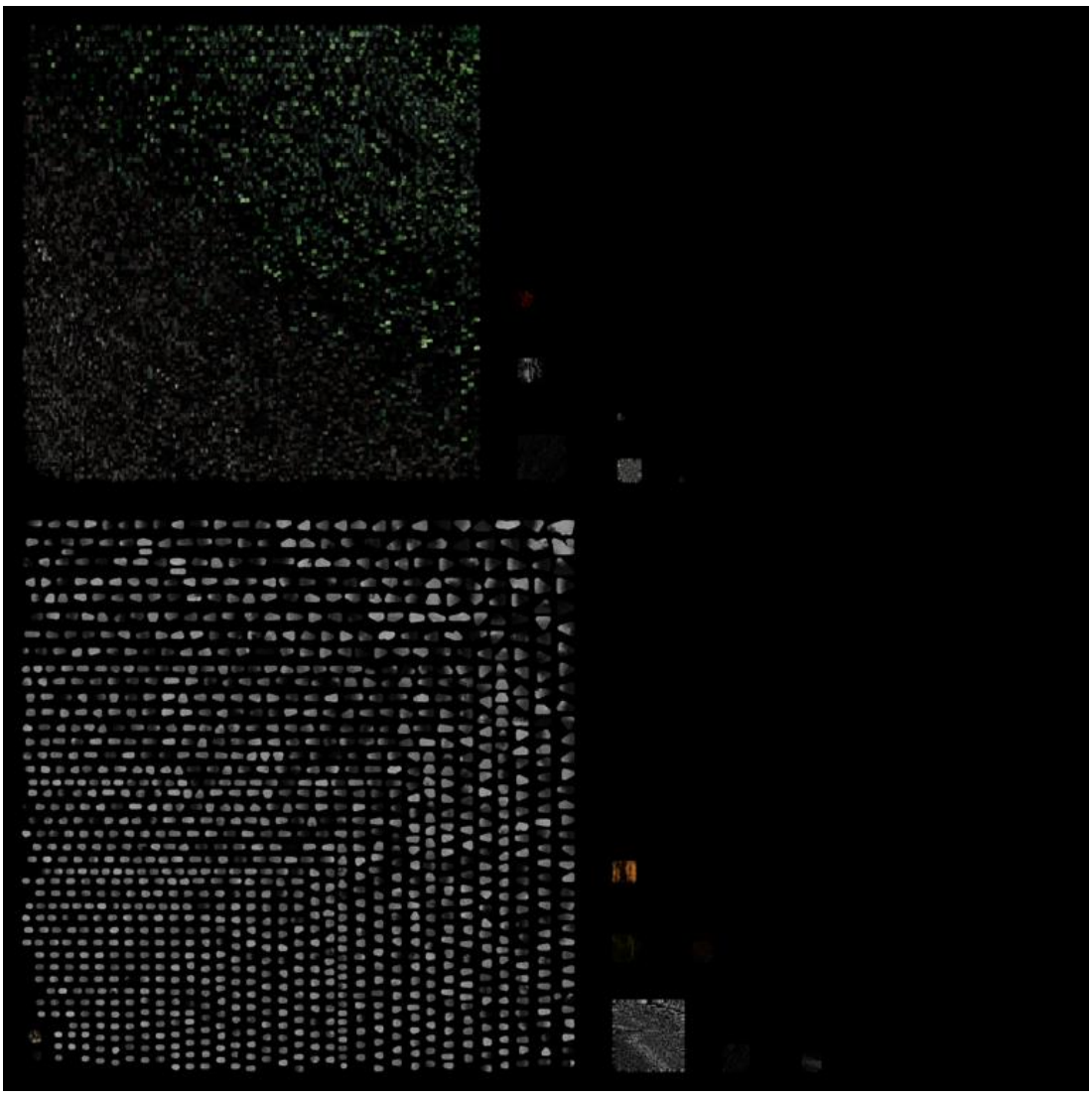

Figure 27: Baked texture map

Normal mapping

In 3D computer graphics, normal mapping is a technique used for faking the lighting of bumps and dents. It is used to add details without using more polygons. A common use of this technique is to greatly enhamce the appearance and details of a low polygon model.

Normal maps are commonly stored as a regular RGB images where the RGB components correspond to the X, Y and X coordinates, respectively, of the surface normal.

Specifically I used the tool of normal mapping in the first scene, to give more realism to the ground and give the feeling of relief on a plane playing with the illumination of the scene.

The steps to obtain and use the normal map are the following:

First, open in Photoshop the texture of the map with the desired surface image and select the NormalMapFilter tool.

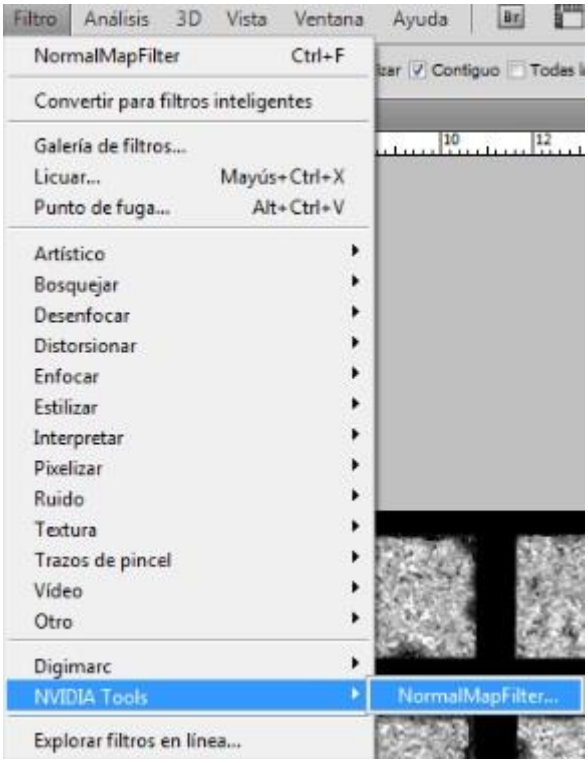

Figure 28: Normal map calculation

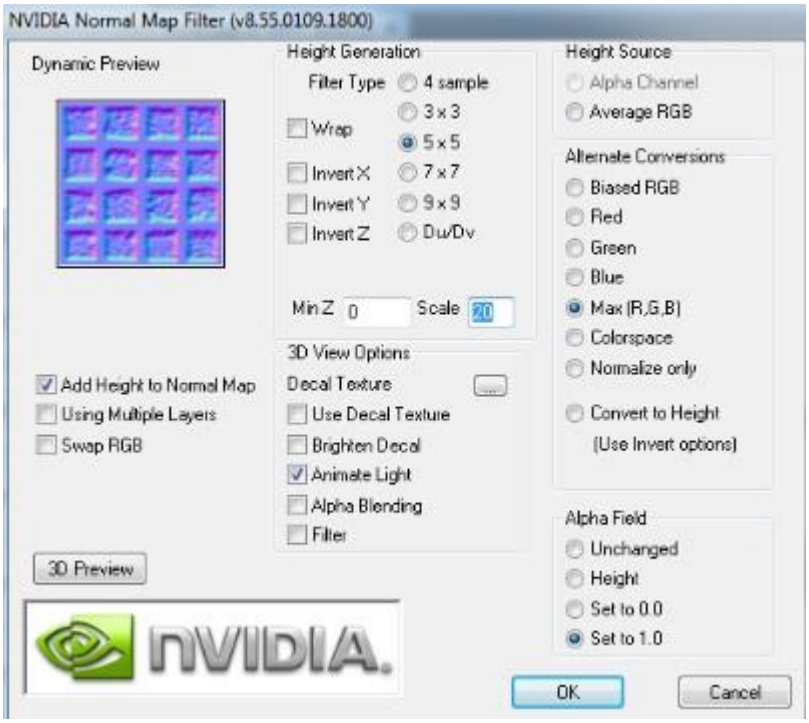

Figure 29: Normal map parameters

Only with this and adjusting the configuration variables of the tool it's obtained the calculated normal map.

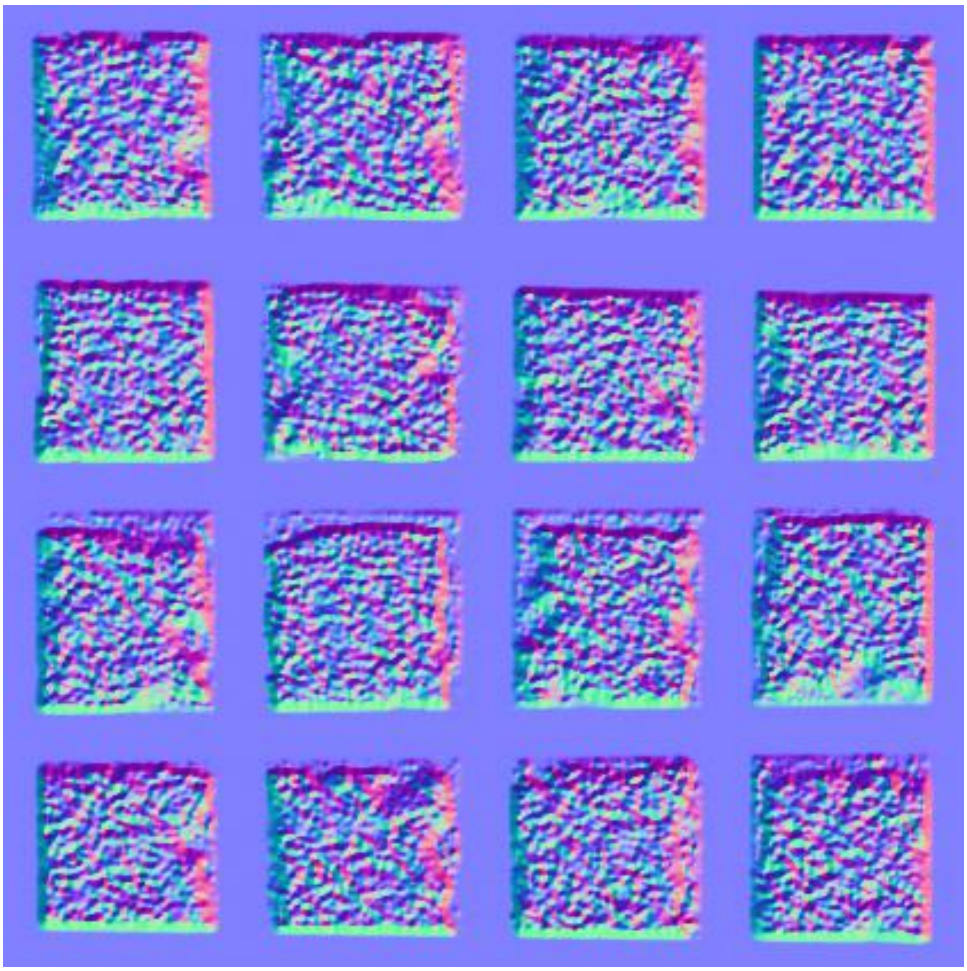

Figure 30: Normal map result

Once the normal map is obtained, to give a little more realism, we add shadows in the original texture.

For this we divide the image into two layers, one with the area of the texture that remains flat and another with the area of the texture with relief.

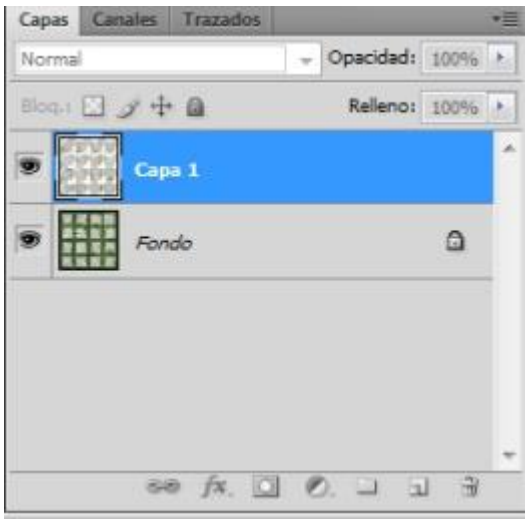

Figure 31: Texture layers

After that, access the options of the layer with relief to add shade, in the section Shadow parallel the configuration values are modified according to the desired shadow.

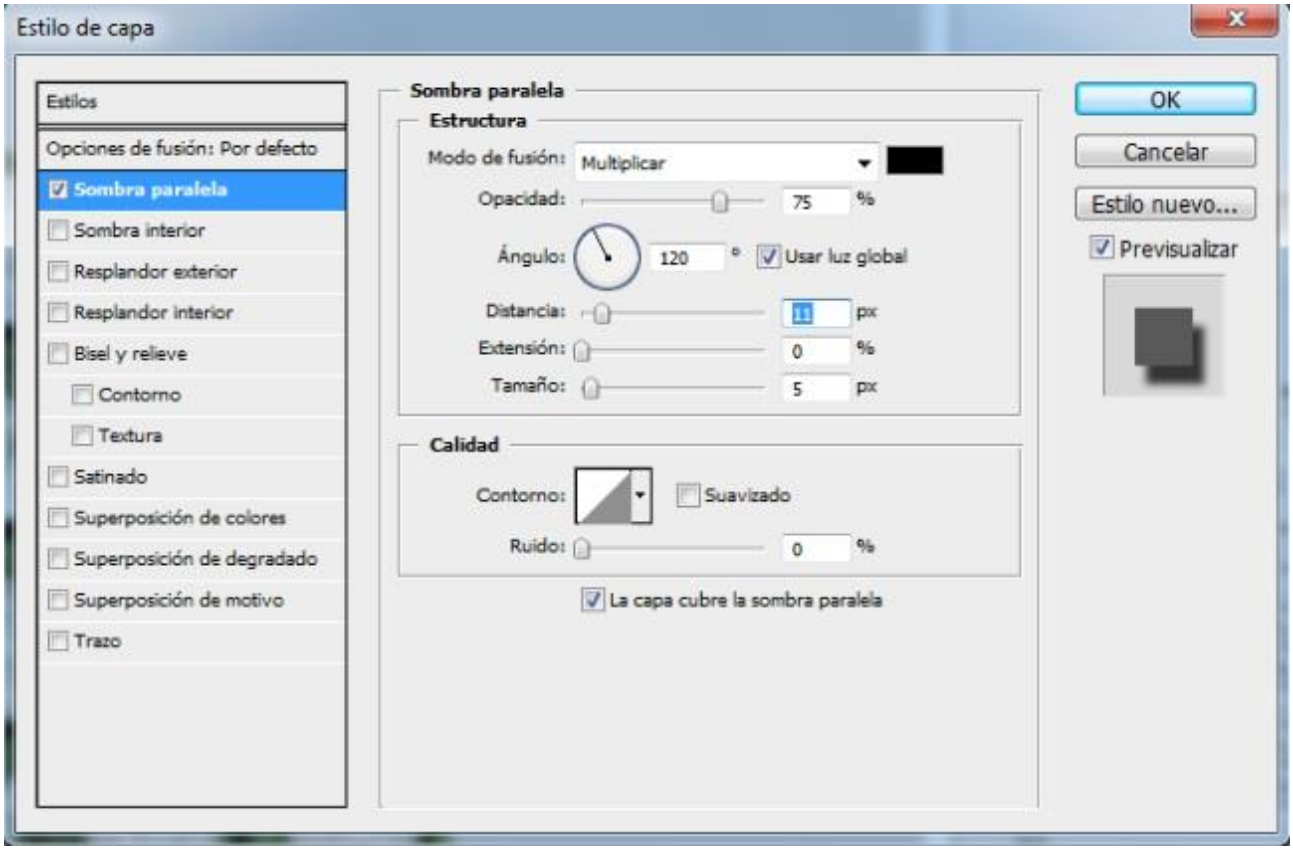

Figure 32: Texture shading parameters

Figure 33: Layer with shading

Figure 34: Full texture with shading

Texture map

For objects that have a concrete texture map, for example, to simulate the texture of wood, rock or specific images, have been used independent textures for each object.

These textures are calculated using the 3ds Max UVW Map tool as well as creating the textures for the character.

Objects with this type of texture have to be exported individually and placed their invidual texture in the Albedo parameter of the material of the object.

To put a texture in unity, in the object material parameters the texture, both an individual texture or a baked texture, is placed in the parameter Albedo.

There are other parameters, for example, the Normal parameter is used in the proyect for putting the Normal maps.

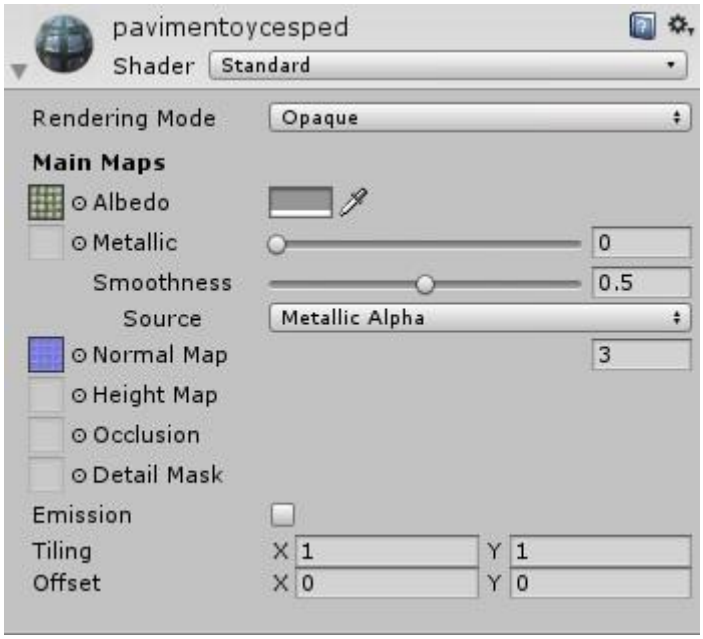

Figure 35: Texture in Unity

Figure 36: Scene composed of different types of texture

Implementation

E-Health Sensor

Arduino

In order to correctly use the Arduino device and obtain the required sensor data, implement an Arduino program that collects the desired data and sends it to the device.

Once it is sent, it is only by connecting the device to the computer that the sensor data can be accessed without implementing any external program.

The program used to collect these data is as follows:

Figure 37: Arduino sensor program

It uses the GetTemperature and GetSkinConductance functions implemented in the eHealth library to obtain the temperature and conductance values.

It also uses a Delay (10000) that causes the values to be read and printed every 10 seconds, so that you can reduce the computational load later in Unity by reading those values.

```
float eHealthClass::getTemperature(void)
{
    //Local variables
    float Temperature; //Corporal Temperature
    float Resistance;  //Resistance of sensor.
    float ganancia=5.0;
    float Vcc=3.3;
    float RefTension=3.0; // Voltage Reference of Wheatstone bridge.
    float Ra=4700.0; //Wheatstone bridge resistance.
    float Rc=4700.0; //Wheatstone bridge resistance.
    float Rb=821.0; //Wheatstone bridge resistance.
    int sensorValue = analogRead(A3);

    float voltage2=((float)sensorValue*Vcc)/1023; // binary to voltage conversion

    // Wheatstone bridge output voltage.
    voltage2=voltage2/ganancia;
    // Resistance sensor calculate
    float aux=(voltage2/RefTension)+Rb/(Rb+Ra);
    Resistance=Rc*aux/(1-aux);
    if (Resistance >=1822.8) {
        // if temperature between 25ºC and 29.9ºC. R(tª)=6638.20457*(0.95768)^t
        Temperature=log(Resistance/6638.20457)/log(0.95768);
    } else {
        if (Resistance >=1477.1){
                // if temperature between 30ºC and 34.9ºC. R(tª)=6403.49306*(0.95883)^t
                Temperature=log(Resistance/6403.49306)/log(0.95883);
        } else {
            if (Resistance >=1204.8){
                // if temperature between 35ºC and 39.9ºC. R(tª)=6118.01620*(0.96008)^t
                Temperature=log(Resistance/6118.01620)/log(0.96008);
            }
            else{
                if (Resistance >=988.1){
                    // if temperature between 40ºC and 44.9ºC. R(tª)=5859.06368*(0.96112)^t
                    Temperature=log(Resistance/5859.06368)/log(0.96112);
                }
                else {
                    if (Resistance >=811.7){
                        // if temperature between 45ºC and 50ºC. R(tª)=5575.94572*(0.96218)^t
                        Temperature=log(Resistance/5575.94572)/log(0.96218);
                    }
                }
            }
        }
    }

    return Temperature;
}
```

Figure 38: GetTemperature function

```
float eHealthClass::getSkinConductance(void)
{
    // Local variable declaration.
    float resistance;
    float conductance;
    delay(1);

    // Read an analogic value from analogic2 pin.
    float sensorValue = analogRead(A2);
    float voltage = sensorValue*5.0/1023;

    conductance = 2*((voltage - 0.5) / 100000);

    // Conductance calculation
    resistance = 1 / conductance;
    conductance = conductance * 1000000;
    delay(1);

    if (conductance > 1.0)  return conductance;
    else return -1.0;
}
```

Figure 39: GetSkinConductance function

GetTemperature function

`int sensorValue = analogRead(A3);` → Stores the values received by pin A3, analog port 3 in the variable sensorValue.

```
float voltage2=((float)sensorValue*Vcc)/1023;

voltage2=voltage2/ganancia;

float aux=(voltage2/RefTension)+Rb/(Rb+Ra);

Resistance=Rc*aux/(1-aux);
```

Mathematical calculations to calculate resistance.

```
if (Resistance >=1477.1){
Temperature=log(Resistance/6403.49306)/log(0.95883);
}
```

From the resistance value it obtains the temperature data.

GetSkinConductance function

`float sensorValue = analogRead(A2);` → Stores the values received by pin A2, analog port 2 in the variable sensorValue.

`float voltage = sensorValue*5.0/1023;` → Calculates the voltage from the value of sensorValue.

`conductance = 2*((voltage - 0.5) / 100000);` → Calculate the conductance from the voltage.

Unity

To read the data received by the sensor from Unity it is necessary, on the one hand, to import the IO.Ports library, as well as in the Unity configuration, to change the Api Compability Level to .NET 2.0 instead of NET 2.0 Subset.

Figure 40: Parameters to change API

Once configured, it is necessary to create a SerialPort object:

```
serialPort = new SerialPort ("COM3",115200);
```

And start by specifying port (COM3) and baud (115200).

Specify the maximum waiting time to read the data:

```
serialPort.ReadTimeout = 1000;
```

And open the serial port:

```
serialPort.Open();
```

Once everything is ready to be able to obtain the data, this function is used:

```
void CalcularDatos(){
    if (calculo) {
        if (tiempoSinDecimales % 10 != 0)
            calculo = false;
    }else if (tiempoSinDecimales % 10 == 0) {
        calculo = true;
        temperatura = "";
        conductancia = "";
        for (i = 0; i < 5; i++) {
            temperatura += (char)serialPort.ReadChar();
        }
        conductancia += (char)serialPort.ReadChar ();
        dato = (char)serialPort.ReadChar ();
        if (dato == '.') {
            conductancia += dato;
            for(i=0;i<3;i++)
                conductancia += (char)serialPort.ReadChar ();
        } else {
            conductancia += dato;
            for(i=0;i<4;i++)
            conductancia += (char)serialPort.ReadChar ();
        }
    }
}
```

Figure 41: Function that obtains temperature and conductance data in Unity

The first lines check that the sensor data is read every 10 seconds, this is because if it is calculated for each second the application slowed down a lot and it worked badly, so I had to make this change (next to the change in the program Of Arduino using the delay) so that the application works better.

The following lines store the values received from the sensor character by character in the variables temperature and conductance using the instruction:

```
serialPort.ReadChar ();
```

Save data between scenes

To save data between scenes I implemented the SaveLoad class, this class creates a file where each variable is stored and then when it wants to access these variables it opens and reads the variable of the file.

This allows you to save the id, temperature and conductance data, and the time the user is playing, so that the data will not be altered or reset when the screen changes. This way it is easier to follow up on the part of the user, as well as to correctly perform a biofeedback with the user.

```
public static void Load() {
    if(File.Exists(Application.persistentDataPath + "/tiempo.int")) {
        BinaryFormatter bf = new BinaryFormatter();
        FileStream file = File.Open(Application.persistentDataPath + "/tiempo.int", FileMode.Open);
        tiempo = (int)bf.Deserialize(file);
        file.Close();
    }
    if(File.Exists(Application.persistentDataPath + "/nombre.string")) {
        BinaryFormatter bf = new BinaryFormatter();
        FileStream file = File.Open(Application.persistentDataPath + "/nombre.string", FileMode.Open);
        nombre = (string)bf.Deserialize(file);
        file.Close();
    }
    if(File.Exists(Application.persistentDataPath + "/temperatura.string")) {
        BinaryFormatter bf = new BinaryFormatter();
        FileStream file = File.Open(Application.persistentDataPath + "/temperatura.string", FileMode.Open);
        temperatura = (string)bf.Deserialize(file);
        file.Close();
    }
    if(File.Exists(Application.persistentDataPath + "/conductancia.string")) {
        BinaryFormatter bf = new BinaryFormatter();
        FileStream file = File.Open(Application.persistentDataPath + "/conductancia.string", FileMode.Open);
        conductancia = (string)bf.Deserialize(file);
        file.Close();
    }
}
```

Figure 42: Function that loads the data of a file in Unity

```
public static void Save() {
    BinaryFormatter bf = new BinaryFormatter();
    //Application.persistentDataPath is a string, so if you wanted you can put that into debug.log if you want to know where save games are located
    FileStream file = File.Create (Application.persistentDataPath + "/tiempo.int"); //you can call it anything you want
    bf.Serialize(file, tiempo);
    file.Close();

    FileStream file2 = File.Create (Application.persistentDataPath + "/nombre.string"); //you can call it anything you want
    bf.Serialize(file2, nombre);
    file2.Close();

    FileStream file3 = File.Create (Application.persistentDataPath + "/temperatura.string"); //you can call it anything you want
    bf.Serialize(file3, temperatura);
    file3.Close();

    FileStream file4 = File.Create (Application.persistentDataPath + "/conductancia.string"); //you can call it anything you want
    bf.Serialize(file4, conductancia);
    file4.Close();
}
```

Figure 43: Function that stores data in a file in Unity

The data is sent to an online database using Google Script. [20]

This works as follows:

- A Google Spreadsheet is created.
- From this Spreadsheet a script is created google, in tools / editor of scripts.

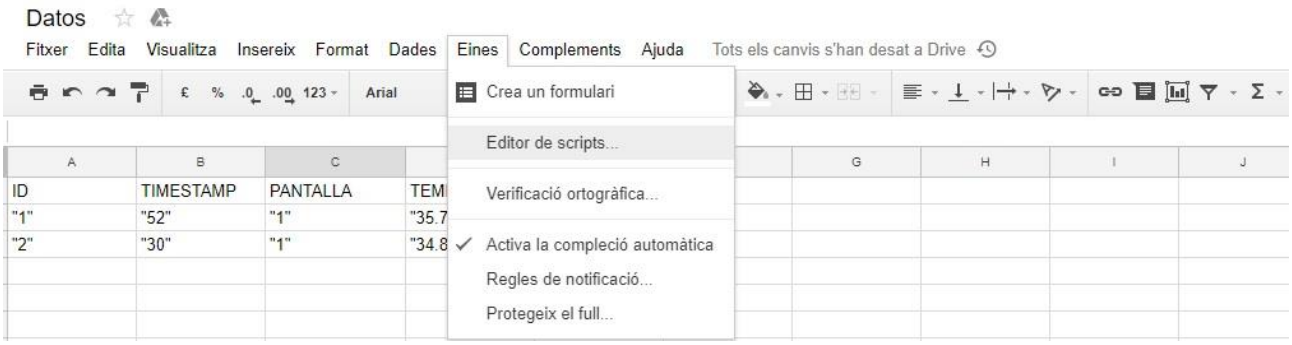

Figure 44: Google Spreadsheet with application data

The script has two parts:

```
function logProductInfo() {
  var sheet = SpreadsheetApp.getActiveSheet();
  var data = sheet.getDataRange().getValues();
  for (var i = 1; i < data.length; i++){
    Logger.log('ID :'+data[i][0]);
    Logger.log('TIMESTAMP :'+data[i][1]);
    Logger.log('PANTALLA :'+data[i][2]);
    Logger.log('TEMPERATURA :'+data[i][3]);
    Logger.log('CONDUCTIVIDAD :'+data[i][4]);
  }
}

function doGet(e) {
  var param1 = JSON.stringify(e.parameter.id);
  var param2 = JSON.stringify(e.parameter.timestamp);
  var param3 = JSON.stringify(e.parameter.pantalla);
  var param4 = JSON.stringify(e.parameter.temperatura);
  var param5 = JSON.stringify(e.parameter.conductividad);
  var sheet = SpreadsheetApp.getActiveSheet();
  sheet.appendRow([param1,param2,param3,param4,param5]);
  return HtmlService.createHtmlOutputFromFile("WebDatos");

}
```

Figure 45: Code.gs script

Code.gs

This part what it does is that when you enter this URL:

https://script.google.com/a/macros/uji.es/s/AKfycbwfH2EoqCvUHFg6dqU94Ji10siSK_9pwJRDZ--A49Eau3N02X18/exec?id=id&timestamp=tiempo&pantalla=1&temperatura=temperatura&conductividad=conductancia,

obtain the values of the parameters id, timestamp, display, temperature and conductivity, and store them in the Spreadsheet.

The doGet function is the one that obtains and collects the values, storing them in the variable sheet.

The logProductInfo function stores the values in the Spreadsheet.

```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
  </head>
  <body>
    Datos recibidos
  </body>
</html>
```

Figure 46: WebDatos.html script

WebDatos.html

This part, all that it does is that, when entering the URL, shows you the message "Data received".

From Unity the data is sent like this:

```
Application.OpenURL ("https://script.google.com/a/macros/uji.es/s/AK-fycbwfH2EoqCvUHFg6dqU94Ji10siSK_9pwJRDZ--A49Eau3N02X18/exec?id="+id+"&timestamp="+tiempo+"&pantalla=1&temperatura="+temperatura+"&conductividad="+conductancia);
```

The data is sent each time you change the scene to be able to track the player.

ID → Id/username.

Timestamp → Time taken by the user in the application.

Pantalla → Scene in which the user is located.

Temperatura → Temperature at that time of the user.

Conductividad → Conductance at that moment of the user.

Change of scene

To change the scene a coroutine is used to display a loading scene between scene and scene.

The instruction that does this is: `StartCoroutine (LoadLevelSlider (n));`, which calls to the function:

```
IEnumerator LoadLevelSlider(int n){
    asyn = Application.LoadLevelAsync (n);
    while (!asyn.isDone) {
        yield return null;
    }
}
```

where n is the number of the scene to be accessed.

What it does is load a scene asynchronously, that is, while the scene continues to run, allowing you to place a loading screen while the scene is being changed.

Menu

For the scene change it uses a menu using the canvas objects offered by Unity, but instead of using a Screen Space canvas, it uses a World Space canvas, because for Cardboard the screen has to be divided in two, and using a Screen Space, the canvas occupied the entire screen.

To select the corresponding scene, a pointer is used in the center of the screen, and when the pointer is above an option, the bluetooth button is pressed to access the scene.

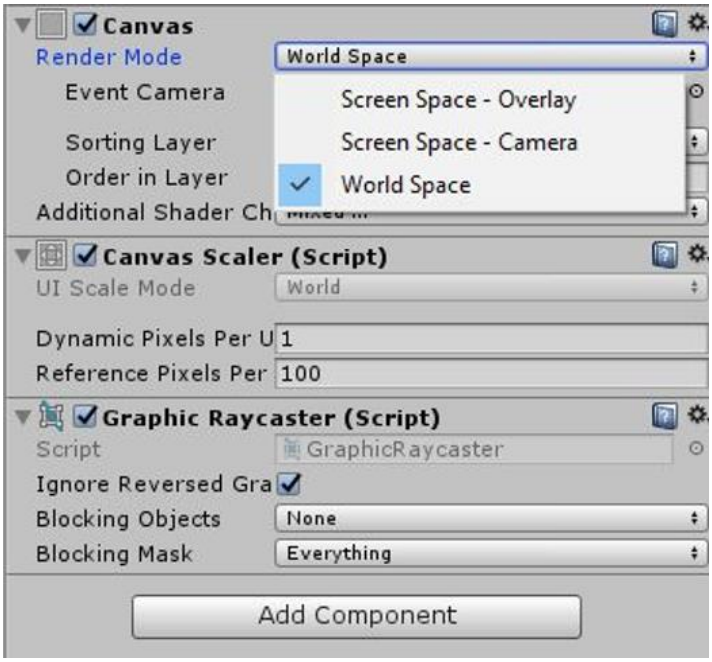

Figure 47: Canvas parameters

For the button to work correctly I use the following:

On the one hand, each button on the canvas has an Event Trigger that detects if the pointer enters and exits the button. When it enters the button it uses a function MenuControl.RecibirDatos that receives the number of the scene to which it carries that button. When the pointer leaves the button, a 0 is sent.

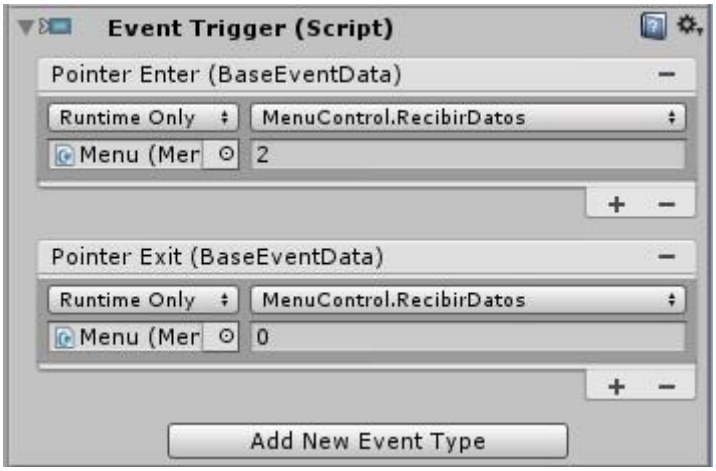

Figure 48: Button Event Trigger parameters

Within the same Script of the menu there is a function that changes of scene that uses the following instruction:

```
if (n!=0&&Input.GetKeyDown(KeyCode.JoystickButton0)){
```

If the parameter you have received (n) is different from 0 (that is, the pointer is inside a button), and the joystick button 0 has been pressed, the scene is changed.

Workplan review and testing

Desviations from the initial plan

I had many problems with the sensor, since I had never worked with Arduino, I was very lost and I had to report and document more than expected. So I had to employ extra work from the planned to this. Instead of 20 h in the end I spent about 45 hours.

I also spent more time than expected to the art of video game, because it has been the part with the difference that I have spent more time, to make sure that the final appearance was good.

Also I had problems with virtual reality. My initial idea was to use the Oculus platform, since I had access to it in my internship. But when trying to test them on my computer (I work on a laptop) I didn't get it to work because of compatibility problems with the HDMI port, and I didn't have another computer on which to make it work.

So due to the features I had, I had to make the application for Google Cardboard. The problem was that the sensor in Arduino that I used worked perfectly in computer, but when trying to connect it to the mobile did not detect it correctly.

I spent 10 more hours just trying to get the device to work with the mobile, but after getting nothing I ended up using the Unity Debug and running the application from Unity with the mobile connected via USB.

By doing so, it works perfectly, both the mobile device with the application and the sensor and sending data to the web page. But since it is not an exported application, I have also exported a mobile version that does not use the sensor data.

The implementation and use of virtual reality cost me a lot less time, since I used a Google Cardboard prefab that already had the necessary scripts for its operation.

	Febrero	*Marzo*	*Abril*	*Mayo*	*Junio*
Task 1	10 h				
Task 2	20 h	30 h			
Task 3		3 h			
Task 4		10 h			
Task 5		25 h	15 h	10 h	
Task 6			20 h		
Task 7			10 h		
Task 8			10 h		
Task 9			10 h	20 h	15 h
Task 10				10 h	
Task 11				20 h	
Task 12				5 h	
Task 13					5 h
Task 14					40 h
Task 15					60 h

Figure 49: Project's final planning aproach

Testing

During debugging on the mobile I discovered a number of problems, for example:

- The screen did not look correctly, it was placed vertically rather than horizontally.
- The bluetooth control does not function correctly when entering the menus.
- Problems with colliders.
- Certain scenes looked worse and slower.
- The gyroscope was not working correctly.
- There were scenes where the Skybox did not look right.

In terms of testing with different users, I was not able to test the full application for lack of time, but I was able to test with individual scenes.

Most people claimed that they were more relaxed after the following scenes:

- Boat trip
- Beach environment
- The attitude towars relaxation and the environment

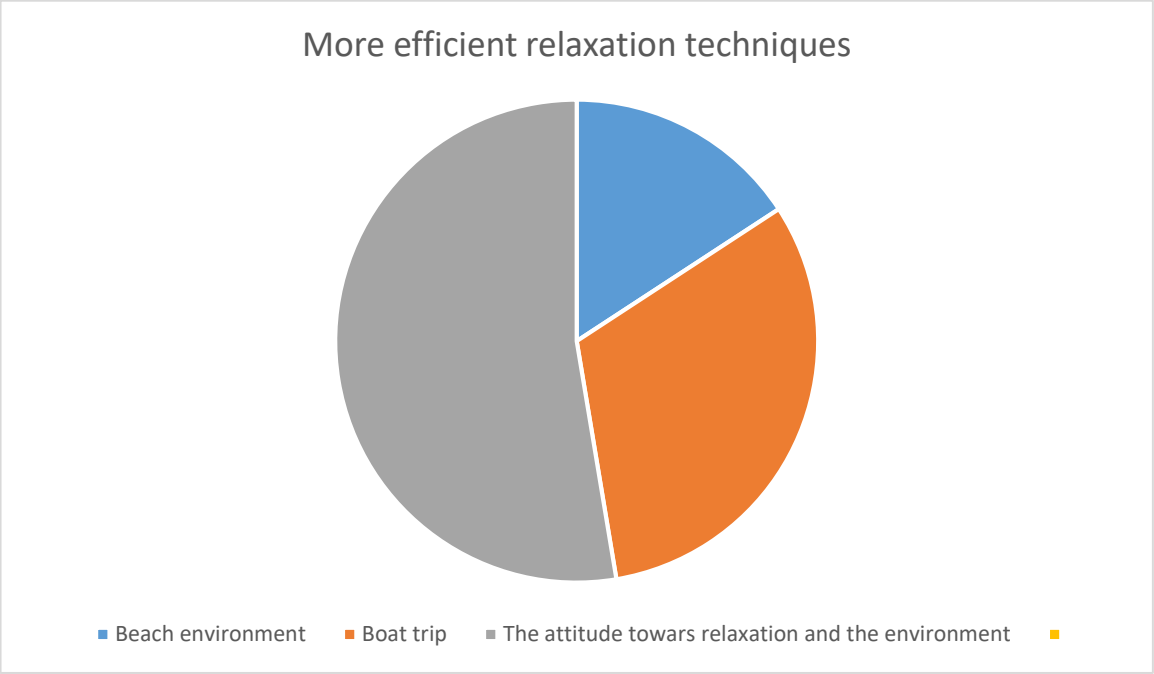

Figure 50: Graphic of relaxation techniques preferred by users

Results and Conclusions

First of all, all the goals proposed at the beginning of this project will be analyzed. We analyze individually all the objectives and the conclusión extracted of all of these. The first of the objectives was as follows:

Obj.1: Create an application that uses different relaxation techniques so that it can be adapted to the user.

This goal is fulfilled, since I have managed to unite different relaxation techniques in a single application, in which the user can choose which techniques to practice.

Obj.2: The user enjoys a complete immersive virtual experience.

This goal has also been fulfilled, because I have created an immersive virtual experience where the user can relax and disconnect from the outside world.

Obj. 3: The user feels more relaxed after using the application, and in turn learns and discovers different methods of relaxation that he can apply at the moment he needs them.

According to the results of the testing (even if they are only of individual scenes and not of the complete experience) I have found that the user is more relaxed after using relaxation techniques and more than one has wanted to repeat or use those tenacies for his account.

General conclusions

I consider that I have met the initial goal, since I have managed to create an interactive virtual reality environment using different relaxation techniques.

In addition, I have learned a lot in many different areas, since this project covers a little of everything.

I have learned to make applications using virtual reality, as well as basically how to use an Arduino board, read and use their data. I have also learned how to use a spreadsheet as a database to store my application data on the internet. I have improved my 3D design and animation skills, as well as my Unity implementation competencies.

I really think this application is very useful, and even if it is a project of only 300 hours of work, I think that with a little more work it could be very useful or reuse the idea and help a lot of people. Because it's more and more in increasing virtual reality for therapeutic reasons.

In summary, this final grade work, has helped me to train, apply my knowledge acquired in the degree, gathering different knowledge from different branches to create this common project.

Figure 51: Final images of the environment part 1

Figure 52: Final images of the environment part 2

Figure 53: Final images of the environment part 3

Bibliography

[1] Consequences of Stress Factors in labor performance in Emergency Area

[2] Técnicas de relajación para aliviar el estrés

[3] Técnicas de meditación

[4] What is Virtual Reality

[5] Paisajes de realidad virtual para aliviar el estrés y combatir la depresión

[6] Realidad virtual para tratar el estrés

[7] Curar el estrés, la ansiedad y las fobias con realidad virtual

[8] Realidad virtual, la nueva medicina contra el estrés

[9] HTC Vive

[10] Oculus Rift

[11] Samsung Gear VR

[12] Google Cardboard

[13] Las mejores gafas de realidad virtual baratas

[14] Biofeedback

[15] Qué es el biofeedback

[16] E-Health Sensor Platform V2.0 for Arduino and Raspberry Pi

[17] Sensor para la muñeca que indica el nivel de estrés

[18] Procemientos de relajación, método auditivo. J.F. Gonzalez Ramirez

[19] Técnicas de relajación y respiración. Fco. Xavier Méndez Carrillo, José Olivares Rodríguez y María José Quiles Sebastián.

[20] Google Apps Script

Appendix

Audio 1: The attitude towars relaxation and the environment

Mantengo una postura cómoda, sentado o tumbado que me permita respirar profundamente y mantener apoyado todo el cuerpo, de forma que no necesite poner ningún músculo en tensión.
Dirijo la atención a la respiración, comienzo a respirar con el abdomen, despacio, muy despacio, sin prisa, profundamente, permitiendo que el aire circule con fluidez hasta el fondo de los pulmones.
(pausa de 10 segundos)
Mantengo la atención concentrada en mi respiración, siento las sensaciones que produce el aire al penetrar y salir por mi nariz.
La noto descender hasta el fondo de los pulmones dilatando suavemente el abdomen.
(pausa de 10 segundos)
Sigo con mi atención concentrada en percibir todas las sensaciones que se desprenden del acto de respirar.
(pausa de 7 segundos)
Compruebo que cada vez que expulso despacio el aire de mis pulmones, me siento un poco más tranquilo.
(pausa de 11 segundos)
Ahora, mientras mantengo la atención concentrada en experimentar todas las sensaciones que se desprenden de la respiración, me repito mentalmente cada vez que inspiro:
Respiro tranquilo, respiro tranquilo
(pausa de 20 segundos)
Sigo repitiéndome mentalmente estas frases mientras respiro profunda y fluidamente con el abdomen, tomo consciencia de las sensaciones de tranquilidad que se desprenden de respirar con el abdomen despacio.
(pausa de 10 segundos)
Ahora con cada inspiración me digo mentalmente: ***Respiro tranquilo***
Y con cada espiración: ***Estoy tranquilo***
Inspiro y me repito: ***Respiro tranquilo.***
Espiro y me repito: ***Estoy tranquilo.***
(pausa de 20 segundos)
Ahora me centro en mis pensamientos, *(pausa)*, sensaciones físicas, *(pausa),* sentimientos, *(pausa),* cómo estoy en este momento, *(pausa),* cómo me siento, *(pausa),* en qué pienso, *(pausa),* solamente observo con atención, sin juicio, *(pausa),* observo incluso lo que no me

gusta, *(pausa),* regreso ahora a mi respiración, conecto de nuevo con ella, *(pausa*), tomo consciencia de todo mi cuerpo, *(pausa),* de cada parte de él, *(pausa),* del cuerpo en su totalidad, *(pausa),* amplío mi consciencia a todo lo que me rodea, *(pausa),* a mi entorno*, (pausa),* el lugar en el que estoy, *(pausa),* y voy abriendo los ojos poco a poco.

Audio 2: Weigth experience

Mantengo una postura cómoda, sentado o tumbado que me permita respirar profundamente y mantener apoyado todo el cuerpo, de forma que no necesite poner ningún músculo en tensión.
Llevo la atención a mi brazo derecho, tomo consciencia de todas las sensaciones que emanan de él.
(5 segundos)
Ahora dirijo mi atención a hacerme consciente del peso de mi brazo derecho, me concentro en percibir esta sensación con nitidez.
(3 segundos)
Cada vez que expulso el aire de mis pulmones, siento que el brazo derecho se vuelve pesado, cada vez más y más pesado.
(3 segundos)
Mientras mantengo la atención concentrada en sentir las sensaciones de peso del brazo derecho, me repito mentalmente cada vez que expulso el aire:
Mi brazo derecho pesa.
(5 segundos)
Dejo de repetirme estas frases y llevo mi atención al brazo izquierdo, tomo consciencia de todas las sensaciones que emanan de él.
(3 segundos)
Ahora me concentro en sentir las sensaciones de peso de mi brazo izquierdo, las percibo lo más nítidamente posible.
(5 segundos)
Continúo respirando lenta y profundamente, soy consciente de que cuando expulso despacio el aire de mis pulmones, mi brazo izquierdo se vuelve más pesado, cada vez más y más pesado.
(pausa)
Me repito interiormente cada vez que expulso lentamente el aire de los pulmones:
Mi brazo izquierdo pesa.
(5 segundos)
Dejo de repetirme estas frases y dirijo la atención a la pierna derecha, tomo consciencia de todas las sensaciones que se desprenden de ella.

(3 segundos)

Ahora me concentro en percibir las sensaciones de peso de mi pierna derecha.

(3 segundos)

Siento que cada vez que expulso el aire muy despacio, las sensaciones de peso se incrementan y la pierna derecha se encuentra más y más pesada.

(10 segundos)

Continúo con la atención concentrada en percibir el peso de mi pierna derecha, noto que a medida que la sensación de peso corporal aumenta, me encuentro más y más tranquilo.

(5 segundos)

Me repito mentalmente cada vez que expulso el aire:

Mi pierna derecha pesa.

(7 segundos)

Dejo de repetirme estas frases y me concentro en la pierna izquierda, tomo consciencia de todas sus sensaciones.

(5 segundos)

Ahora me concentro en sentir el peso de mi pierna izquierda.

(5 segundos)

Observo que cada vez que expulso el aire, la sensación de peso se incrementa y se extiende por toda la pierna, siento la pierna izquierda pesada, muy pesada.

(3 segundos)

Me repito mentalmente cada vez que expulso despacio el aire de los pulmones:

Mi pierna izquierda pesa.

(7 segundos)

Tomo consciencia de las sensaciones de peso que se extienden por ambos brazos y piernas, cada vez que expulso muy despacio el aire de los pulmones, los brazos y las piernas se vuelven más y más pesados, siento como si estuvieran pegados a la superficie en la que se apoyan.

(3 segundos)

Me concentro en sentir estas sensaciones al tiempo que me repito mentalmente cada vez que expulso el aire:

Los brazos y las piernas pesan.

(10 segundos)

Siento mi cuerpo pesado, muy pesado, como si fuera atraído por una gran fuerza hacia abajo.

(3 segundos)

Tomo consciencia del peso de todo el cuerpo mientras me repito mentalmente:

Mi cuerpo pesa, mi cuerpo pesa.

(10 segundos)

Me siento tranquilo, muy tranquilo, sensaciones de relajación y bienestar me recorren.

Ahora voy a hacer un recorrido por todas las sensaciones que progresivamente he ido haciendo aparecer en mi cuerpo, me voy a repetir mentalmente las frases siguientes y al mismo tiempo voy a dirigir la atención a percibir las sensaciones sugeridas por estas frases lo más nítidamente posible. Cuanto más me concentre, más claramente las sentiré.
Inspiro con el abdomen despacio, al tiempo que me repito mentalmente:
Respiro tranquilo
(5 segundos)
Permanezco tranquilamente durante unos momentos en la posición en la que me encuentro, disfrutando de las sensaciones de relajación que he conseguido.
(10 segundos)
Ahora voy a contar del 5 a 1. Cuando llegue a uno me recuperaré.
5 4 3 2 1
Respiro profundamente, 2 ó 3 veces, me muevo libremente poco a poco, despacio, y me recupero.

Audio 3: Warm experience

Mantengo una postura cómoda, sentado o tumbado que me permita respirar profundamente y mantener apoyado todo el cuerpo, de forma que no necesite poner ningún músculo en tensión.
Llevo la atención a mi mano derecha, me concentro en sentir el calor que se desprende de ella.
Aunque en un primer instante no lo perciba, la mano está caliente, irradia calor. Si me concentro con atención lograré sentir la temperatura de la mano con nitidez.
(3 segundos)
Respiro profundamente, despacio, muy tranquilo. Siento el calor de mi mano derecha, me hago consciente de él.
(5 segundos)
Mantengo la atención concentrada en percibir el calor que irradia de la mano derecha, cada vez que expulso el aire muy despacio, aumenta las sensaciones de calor de mi mano derecha.
(3 segundos)
Me concentro en esta sensación mientras me repito mentalmente:
Mi mano derecha está caliente.
(10 segundos)
Dejo de repetirme estas frases y llevo la atención a mi brazo derecho, siento su calor, tomo consciencia de él.
(7 segundos)

Cada vez que expulso despacio el aire de los pulmones siento que la temperatura del brazo derecho aumenta.

(7 segundos)

Tomo consciencia de esta sensación al mismo tiempo que me repito mentalmente.

Mi brazo derecho está caliente.

(10 segundos)

Dejo de repetirme estas frases y llevo la atención a mi mano izquierda, me concentro en sentir el calor que se desprende de ella.

(7 segundos)

Comprendo que al concentrar la atención soy capaz de percibir la temperatura de la mano izquierda, siento su calor.

(3 segundos)

Me repito mentalmente cada vez que expulso el aire:

Mi mano izquierda está caliente.

(10 segundos)

Dejo de repetirme estas frases y me concentro en mi brazo izquierdo, siento la temperatura del brazo, tomo consciencia de ella.

(7 segundos)

Siento el calor de mi brazo izquierdo, soy consciente de esta sensación al mismo tiempo que me repito mentalmente:

Mi brazo izquierdo está caliente.

(10 segundos)

Dejo de repetirme estas frases y me concentro en mi pierna derecha, tomo consciencia de su temperatura.

(7 segundos)

Siento el calor que irradia de mi pierna derecha, tomo consciencia de su temperatura al tiempo que me repito mentalmente cada vez que expulso el aire:

Mi pierna derecha está caliente.

(10 segundos)

Dejo de repetirme estas frases y dirijo la atención a mi pierna izquierda. Tomo consciencia de su temperatura.

(7 segundos)

Me concentro en sentir el calor que irradia de mi pierna izquierda y me repito mentalmente cada vez que expulso el aire:

Mi pierna izquierda está caliente.

(10 segundos)

Las sensaciones de calor se extienden por todo mi cuerpo, una sensación muy agradable me recorre, estoy muy tranquilo, muy relajado.

(10 segundos)

Ahora voy a hacer un recorrido por todas las sensaciones que progresivamente he ido haciendo aparecer en mi cuerpo, me voy a repetir mentalmente las frases siguientes y al mismo tiempo voy a dirigir la atención a percibir las sensaciones sugeridas por estas frases lo más nítidamente posible. Cuanto más me concentre, más claramente las sentiré.

Inspiro con el abdomen despacio, al tiempo que me repito mentalmente:

Respiro tranquilo

(5 segundos)

Permanezco tranquilamente durante unos momentos en la posición en la que me encuentro, disfrutando de las sensaciones de relajación que he conseguido.

(10 segundos)

Ahora voy a contar del 5 a 1. Cuando llegue a uno me recuperaré.

5 4 3 2 1

Respiro profundamente, 2 ó 3 veces, me muevo libremente poco a poco, despacio, y me recupero.