UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

DEGREE DISSERTATION

Mechanical engineering degree

DESIGN OF VR APP APPLIED TO COGNITIVE TRAINING

Volume I

Authors: Adrián Mora Pedregosa
Jesús María Merino

Director: Jordi Torne Ribé

Date: January 2018

Resum

L'objectiu principal d'aquest projecte és el disseny d'una aplicació de realitat virtual per millorar el tractament dels pacients amb deteriorament cognitiu lleu, així com estudiar els possibles avantatges que aquesta tecnologia pot proporcionar en aquest camp.

Es va escollir la realitat virtual perquè permet augmentar la sensació d'immersió pel que fa a les tecnologies actuals. Actualment la realitat virtual s'està utilitzant amb aquest tipus de tractament i està aconseguint gran resultats amb els pacients.

A més, mitjançant l'ús d'aquesta tècnica d'immersió visual, s'espera que ajudi a millorar la capacitat dels pacients davant nous problemes, com pot ser la iniciació a la realitat virtual, una qüestió fonamental que ajuda a la millora dels pacients que es troben en les primeres etapes de la malaltia.

L'aplicació consisteix en un entorn de supermercat virtual on el pacient pot realitzar diverses proves. En aquesta hi haurà diferents nivells amb diverses complexitats, sempre després d'haver realitzat un tutorial previ.

L'aplicació s'ha realitzat en dues fases diferents: primer es va crear el guió, amb col·laboració amb la unitat d'Alzheimer de l'Hospital Clínic. Els nivells de l'aplicació es van definir aquí. El següent va ser la realització de l'aplicació amb col·laboració amb la companyia Vysion 360.

Per a la seva utilització per la unitat d'Alzheimer de l'Hospital Clínic, l'aplicació tenia que complir diferents criteris. En primer lloc, els nivells de dificultat tenen que ser suficients per realitzar un tractament a llarg termini. En segon lloc, per crear una bona experiència de immersió, l'entorn creat té que ser el més realista possible. Finalment, s'ha creat una base de dades local per guardar la informació de totes les sessions, utilitzat posteriorment en l'anàlisi de evolució dels pacients.

Amb aquesta aplicació, s'espera que els resultats en els pacients amb deteriorament cognitiu lleu milloren respecte a les tècniques anteriors. Especialment gràcies a la gran experiència d'immersió aconseguida amb la realitat virtual, la qual ajuda a la concentració dels pacients durant el tractament.

Resumen

El objetivo principal de este proyecto es diseñar una aplicación de realidad virtual para mejorar el tratamiento de pacientes con deterioro cognitivo leve, así como estudiar las posibles ventajas que esta nueva tecnología puede proporcionar en este campo.

Se eligió la realidad virtual porque permite aumentar la sensación de inmersión con respecto a las tecnologías actuales. Hoy en día la realidad virtual se está utilizando con este tipo de tratamientos y está logrando grandes resultados en los pacientes.

Además, mediante el uso de esta técnica de inmersión visual, se espera que ayude a mejorar la capacidad de los pacientes frente a nuevos problemas, como puede ser la iniciación a la realidad virtual, una cuestión fundamental que ayuda a mejorar a los pacientes que se encuentran en las primeras etapas de la enfermedad.

La aplicación consiste en un entorno de supermercado virtual donde el paciente puede realizar varias pruebas. En esta habrá diferentes niveles con diversas complejidades, siempre tras haber realizado un tutorial previamente.

La aplicación se ha desarrollado en dos fases diferentes: primero se creó el guión, en colaboración con la unidad de Alzheimer del Hospital Clínico. Los niveles de la aplicación se encuentran definidos en éste. El paso siguiente fue la realización de la aplicación en colaboración con la compañía Vysion 360.

Para su utilización por la unidad de Alzheimer del Hospital Clínico, la aplicación tenía que cumplir diferentes criterios. En primer lugar, los niveles de dificultad deben ser suficientes para realizar un tratamiento a largo plazo. En segundo lugar, para crear una buena experiencia de inmersión, el entorno creado debe ser lo más realista posible. Finalmente, se ha creado una base de datos local para guardar la información de todas las sesiones, utilizados posteriormente en el análisis de evolución de los pacientes.

Con esta aplicación, se espera que los resultados en pacientes con deterioro cognitivo leve mejoren con respecto a las técnicas anteriores. Especialmente gracias a la gran experiencia de inmersión lograda por la realidad virtual, que ayuda a la concentración de los pacientes durante el tratamiento.

Abstract

The main aim of this project is to design a virtual reality app to improve the treatment of patients with mild cognitive impairment as well as study the possible advantages that this new technology may provide in this field.

Virtual reality has been chosen because it allows increasing the feeling of immersion with respect to the current technologies. Nowadays virtual reality is being used to proceed with this kind of treatments and it is achieving great results in patients.

Furthermore, by the use of this visual immersion technique, it is expected to help improve the patients' capacity over new topics, as can be virtual reality, a fundamental issue to overcome by patients who are in the early stages of the disease.

The application consists in a virtual supermarket environment where the patient may perform various tests. In the application there will be different levels with various complexities, always relying on a previously performed tutorial.

The application has been developed in two different phases: first the script was created, in collaboration with the Hospital Clinic's Alzeimer's unit. The levels of the application were defined in it. Then, the actual app was made with the support of the company Vysion 360.

In order to be used in Hospital Clinic's Azheimer's unit, different criteria had to be fulfilled by the application. First, difficulty levels should be enough to perform a long term treatment. Second, in order to create a good immersive experience, the created environment should be as realistic as possible. Finally, a local database is required in order to save the data of all sessions, used in the patients' evolution analysis.

Using this application, results on patients with mild cognitive impairment are expected to improve with respect to old techniques. Especially thanks to the great immersive experience accomplished by the virtual reality, helping the patients to be more focused during the treatment.

Acknowledgements

In the last weeks of the final project is the time to express our gratitude to people who helped us in these months, because of their fundamental participation on it.

Firstly, our family and friends since they have supported us on this path, they know better than nobody that it wasn't easy but they have always helped us at all times.

Thanks to Jordi Torner and Francesc Alpiste for giving us this opportunity, because this has been the first time where we have been able to work with a great company and a hospital.

A fundamental part of the project was Visyon360, and especially Iñigo Gaínza and Josep Lorente for sharing their experience and helping us with the video game engine, Unity3D.

Hospital Clínic gave us a good opportunity of seeing how rewarding is to help people.

Kuroi helped us with the Audio-visual contents, special thanks for Andoni Sarabia, who was a crucial part, sacrificing his time recording audios and recording videos on several parts of the application.

At last but not least, the pedagogue Bélen Catalán who helped us with the writing of dialogues and the provision of the app voice. But above everything, thanks to Belén for offering us her advice and support.

Index

Figures index

Tables index

1. Preface

1.1. Project background

The project was born because of the need to update cognitive training. Nowadays, this training is very old and needs some changes.

Therapies continue with the same mechanisms and this is always a problem, because the immobility it doesn't allow progress in any direction. Also there are several studies where the use of VR in the cognitive training is quoted.

For these reasons Jordi Torner, Francesc Alpiste and the Alzheimer's unit from Hospital Clínic decided to begin with this project, in order to study the inclusion of this technology in the therapy.

1.2. Motivation

1.2.1. Medical motivation

The introduction of Virtual Reality techniques has meant a great improvement on the field of cognitive training. Furthermore, there are several studies talking about using Virtual Reality in Dementia or cognitive problems, for this reason the doctors and UPC teachers have believed that VR could help to improve cognitive training.

Nowadays, cognitive training exercises are not using the technology supports and they have not been updated for some years. Virtual Reality immerses patients in new scenery, this immersive experience cannot be simulated with the old cognitive exercises, and patients will be able to interact with an artificially created 3D environment.

1.2.2. Personal motivation

Virtual reality is growing by leaps and bounds and in the near future could substitute or improve current technology. Therefore learning about it could be very beneficial for our professional careers and for it this reason we decided to create an application for cognitive training.

This app can improve patients' lives, in other words, it is not a regular Final Project because the application may even substitute the current cognitive exercises and it might be a great and important development in cognitive training.

Working with an important VR company and a top hospital is a great challenge, since we will see real problems and search for an optimal solution as quickly as possible. Another point to consider is the relationship with the hospital, because sometimes they ask for things which cannot be recreate on Virtual Reality (VR) and we have to reach a consensus to solve it.

1.3. Previous knowledges

To carry out the project is necessary to have previous knowledge on different subjects, which are going to be explained below:

1.3.1. Video game engine

Unity 3D

Knowledge of this software has been very important to do this project as Unity 3D is the video game engine.

At the beginning our team did several official Unity tutorials, which helped to learn the basic commands of Unity. To create the menu games we searched for many official and unofficial tutorials, where different canvas were explained and they we used to get better results.

On the other hand, scripting is possibly the hardest part to learn since we had never studied C# language, commands and interfaces before.

Finally, we needed a place where we could save patient information and the simplest solution was to work with the data program SQLite.

Fig. 1 Unity 3D logo

1.3.2. Assets/Objects

Realistic assets and objects are very important to improve the immersion feelings, but these are not always that easy to find, below, the different forms to do it will be explained. These different kinds of assets are:

Official assets/objects

Sometimes there are asset packs, which are extremely easy to download and they appear on Unity in minutes.

Other objects

Normally the necessary objects are not available on Unity and they must be designed. For this reason we had to use with other programs such as:

Blender

Blender was very useful in modelling objects which didn't appear in any official asset, because of this, we learned the basic commands of Blender.

Fig. 2 Blender's logo

Kinect

Sometimes the structures of the pieces were very laborious and we decided to do them with Kinect, which allows us to scan the objects and import the structure and texture into Unity.

Fig. 3 Kinect's logo

Paint

Usually finding objects was easy, but when they were imported into Unity the texture was lost and the easiest way to get a realistic texture was by using Paint.

Fig. 4 Paint's logo

2. Introduction

2.1. Project goals

2.1.1. Medical goals

- The main goal in this project is to create a virtual reality app that improves the mechanics of the cognitive training department. We must not forget that cognitive problems do not have a cure but we can slow down the disease. The aim is to produce an app in which patients can be immersed into a real supermarket thanks to a virtual reality, and it might improve the old techniques and increase the good results.

- On a VR app is always very important the realism of the object on it, for that reason searching a usual supermarket structure is fundamental, also products have to be easily and quickly recognizable.

2.1.2. Project goals

- Acquiring knowledge in VR technology.
- Learning C# language to be able to carry out a correct scripting.
- We want to be acquainted with new(s) programs such as Unity 3D or Blender.
- An elemental goal is working in group, for it we have really good communication achieve (at) the best solution.
- We have to optimize time and divide tasks to achieve the goals in time.
- Finally, we must be able to do an evaluation of the project and searching future ways to improve it.

2.2. Scope of the project

The aim of this dissertation is to create a VR application to improve the lives of patients who suffer from mild cognitive impairment. For this purpose, the VR application has to be a realistic simulation of a supermarket and not to be frustrating when it comes to difficulty, so in order to achieve this goal the app will have different game modes and in each of them different levels of difficulty. For improving the immersion and training the products are easily recognizable for the patient.

The application is the first step on a long path , when the project is carried out other students will check its efficiency, if the results are positive, the hospital and UPC members will start thinking the next steps like new environments, game modes and difficulty levels.

3. State of art

3.1. Virtual reality

3.1.1. Definition

Virtual reality could be defined as an artificial environment created by software, which thanks to the immersive capabilities of this technology, makes the user believes that they are in a real world.

Nowadays, VR is primarily experienced trough sight and sound, but due to the so called “VR second wave” a lot of developments on what comes to software and hardware are taking the immersive experience to the next level. Some of these developments could be the wearable computers which allow making more lightweight hardware or the haptic devices that let the user feel the environment.

3.1.2. History

VR’s technology evolution is shown in the **figure 5**, but we are about to explain few of the most important developments that took part on it.

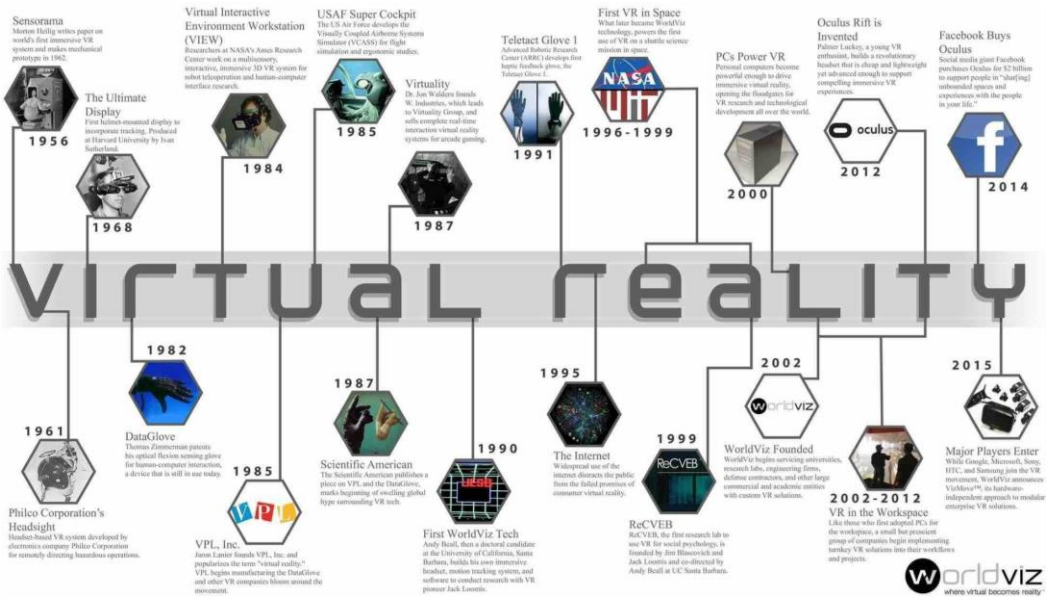


Fig. 5 Virtual Reality’s timeline (extracted from Worldviz web page)

As many of the current technologies, VR's first approach to the world was through science fiction. In the year 1930, before the existence of any technology that could make possible VR, a film created by Stanley G. Weinbaum contained the idea of a pair of glasses that let the wearer experiences a fictional world.

The next appearance that could be considered close to the VR definition was in the year 1957, when the cinematographer Morton Heilig developed the Sensorama (patented in 1962). The Sensorama was intended to fully immerse the user in a film thanks to stereo speakers, a stereoscopic 3D display, fans, smell generators and a vibrating chair. In the **figure 6** we can see an image of the named invention.

Fig. 6 Immersive experience Sensorama

In 1968 Ivan Suntherland created the first head mounted display (HMD) that was connected to a computer, called Sword of Damocles. It was a really large and heavy machine which made it impossible for any user to wear it comfortably. As we can see in the **figure 7**, besides the graphics generated by the computer were very primitive wireframe rooms and objects.

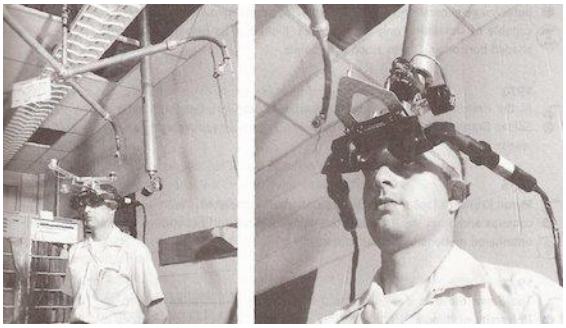


Fig. 7 First head mounted display, Sword of Damocles

In 1995 the Nintendo Virtual Boy, which we can see in the **figure 8**, was launched as the first ever portable console that could display true 3D graphics. It was first released in Japan and North America at a price of $180 but it was a commercial failure despite price drops.

The reported reasons for this failure were a lack of colour in graphics (games were in red and black), there was a lack of software support and it was difficult to use the console in a comfortable position. The following year they discontinued its production and sale.

Fig. 8 Nintendo Virtual Boy console

The first seventeen years of the 21st century have seen the second wave of the VR that we mentioned before, thanks to the recent evolution on computer technology, specially on small and powerful mobile technologies. Also because of the constant drops in prices we have reached a point where the unfulfilled promises made by VR in the 90s will finally come to reality and will be seen in the market. These promising technologies are the ones which we are going to talk about in this project. Also we will discuss the different kinds of current VR technologies and we will conduct a market analysis.

3.2. Market analysis

Down below we are going to show a small description of each of the current possibilities available in the market for VR. The order of presentation will be from the simplest technology to the most complex one.

Google Cardboard

Google Cardboard is a VR platform developed by Google for using it in combination with a head mount and a smartphone. As we can see in the **figure 9** the components of the Google Cardboard are low-cost, they consists of a piece of cardboard cut in a precise shape, two lenses 45mm focal lenses, magnets and a rubber band. It was introduced at Google I/O 2014 developer's conference, with its own platform for developing VR applications.

Thanks to the low cost of the developer's kit, it had a really good reception; over 10 million Cardboards were shipped until March 2017. Of course this tech is plenty of limitations, such as we can talk about the lack of controllers, which makes more difficult the interactions with apps, another downside is that the power of the display depends on the user phone.

Fig. 9 Google CardBoard's first design

But thanks to the good reception in market, Google is working on the next generation called Google Daydream, that we can see in the **figure 10**, which includes a wireless controller that makes possible to interact with apps.

Fig. 10 Google daydream prototype

Samsung Gear VR

Samsung Gear VR is a mobile VR headset developed by Samsung Electronics in collaboration with Oculus. In this gadget, the headset display and processor is a Samsung mobile while the Gear VR acts as the controller. A trackpad is located on the right of the device as we can see on the **figure 11**.

Fig. 11 Samsung Gear VR second generation

Oculus Home is the main facility to download content on the Samsung Gear VR. In September 2017 Samsung released a new Gear VR controller which is independent and will be include with the updated model.

Oculus Rift

Oculus Rift is a VR Headset which started as a Kickstarter campaign in 2012 in which achieved 2.5 million dollars when the initial goal was 250 thousand dollars, it was a great success.

This device needs the use of an external computer, which makes a huge difference on the graphics and processing capabilities, of course these hardware requirements make much more expensive the initial investment but the difference in the immersion experience justifies it.

Thanks to the OLED display, the integrated headphones, the controllers and the tracking system, the immersive experience is much more real, so we can say that this is a step forward than the other gadgets we have talked about.

Fig. 12 Oculus Rift head mounted display

HTC Vive

In what comes to immersive experience on VR, the HTC Vive is the best option users can found on the current market. HTC Vive is developed by HTC with the collaboration of Valve, a video game company which owns the well-known Steam platform. The device was released in 2016 and, like the Oculus Rift, also it needs an external computer for its working.

The main difference between the Oculus and HTC is the tracking system HTC uses based on scaled room which allows a better movement that is traduced in a better user experience.

Fig. 13 HTC VIVE head mounted display

At the table below we can see a synthetized comparison between the named technologies:

Model	Google Cardboard	Samsung Gear	Oculus Rift	HTC Vive
Image				
Price	5€	129,99€	499€	699€
Platform	Most smartphones	Samsung smartphones	PC	
Controller	None	Trackpad	Motions controllers	Motion controllers
Positional tracking	None	None	Camera	Laser towers
Advantages	• Low price • Phone compatibility	• Low price •	• God controllers • Commodity • Good quality in graphics • Allows user displacement • Nice immersive experience	• God controllers • Commodity • Good graphics quality • Allows user displacement in a wider range • Nice immersive experience • A lot of information in what comes to applications development
Downsides	• Very simple	• Only works with Samsung phones • Limited by mobile hardware	• High price • High PC specifications requirements	• High price • High PC specifications requirements

Table 1 Comparison VR technologies

3.3. Possible applications

VR has an enormous potential to change the future for a huge number of fields, such as medicine, business, architecture, the army, etc… Hereunder we are talking about the most significant in some fields:

Architecture

The tool allows the user to make virtual tours through spaces designed in three dimensions, thanks to the computer-generated images, which will replace hand-drawn renderings and that is traduced in reducing costs and time.

Fig. 14 Example of VR architectire app prototype extracted from affinityvr web page

Also another side use that will enhance the security will be the use of those 3D designed spaces, to see in how much time someone will be able to exit the building in case of emergency.

Business

The main use is the conferences by VR, so instead of travelling for a meeting or job interview, companies are using VR conference rooms. An example of the result that can be obtained is shown seen in the **figure 15.**

Fig. 15 Image of how could look a possible VR meeting

Gaming

It is by far the field where VR is having the fastest advances, which is increasing the interest in VR because of the multiple games showing off in market, but mostly because of the resources used on it can be extrapolated to other areas where can be very useful.

In what comes to games, there are possibilities for every customer and a wide range of prices which are dropping down each year. In the figure 16 is shown one PC game entitled Skyrim which was fully converted to VR game, the critics have flattered for the good experience achieved.

Fig. 16 Skyrim game scene

Medical

This field is also having a lot of progression in what comes to new application and interest of professionals, thanks to the wide range of possibilities than can be achieved with this tech.

For example, on the rehabilitation area, a year-long study with paraplegics patients wearing VR headsets, who had to move through a stadium were able to regain some brain functions associated with moving their legs, each of the patients tested recovered some control and the half of them upgraded from full paraplegics to partial paraplegics.

In what comes to the odontology area, there are tests where using a coastal scene to distract patients, they have less pain sensation.

In psychology there are possibilities in many treatments. There can be found due to the great capacity of immersive experience, where the patient can be exposed in different scenarios that the psychologist wants to.

Another example we could find is in treatment of burn victims, who were treated with the help of a game called SnowWorld, which allows users to throw snowballs at penguins while listening to Paul Simon, this treatment has proven more effective than morphine.

Many more examples of how VR is revolutionizing some fields in medicine could be quoted, as last example of this could be on the field we are interested about, the possible treatments in cognitive impairment or dementia patients.

“VR- based cognitive rehabilitation systems support procedures for mitigating behavioural and psychological symptoms of patients having mild cognitive impairment and early-stage Alzheimer’s disease.” (American Journal for Alzheimer’s disease and other dementias). In a report published by CNN, it was mentioned that scientist are beginning to study different leisure activities and their impact to the brain, those kind of activities could reduce the chances of developing mild cognitive impairment by 30% to 50%, according to a 2011 study published in The Journal of Neuropsychiatry.

VR is taking a unique spot in the world of Alzheimer and Dementia through a quality of life enhancement, education and attenuation of symptoms.

4. Current situation on mild cognitive impairment treatment

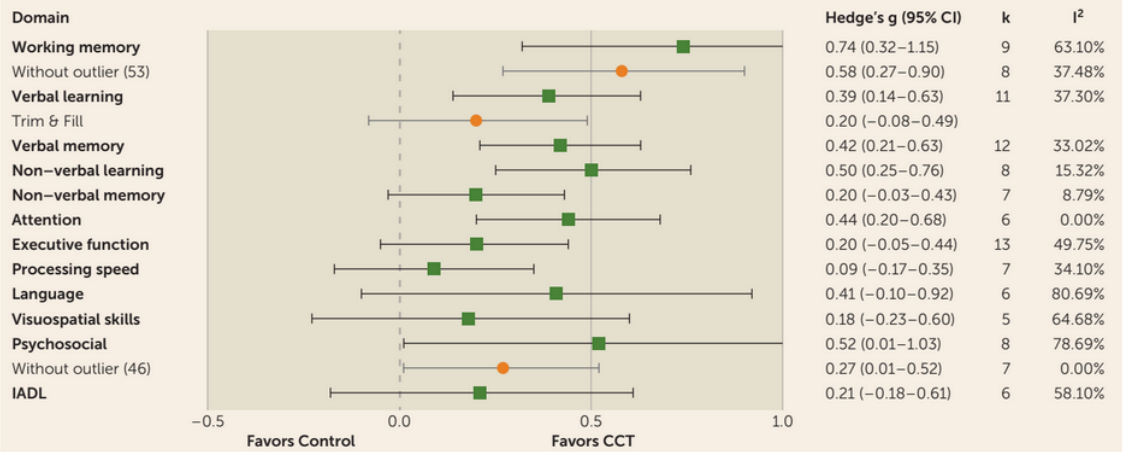
The demographic changes of our aging population lead to increase the number of patients with mild cognitive impairment (MCI), frequently a precursor to dementia. The UK National Institute for Health and Clinical Excellence (NICE) recommends follow-up to ensure dementia is diagnosed and care planned at an early stage, but no specific treatments; the truth is that is known little about how to treat it.

There are two types of intervention, pharmacological intervention and non-pharmacological intervention, since the main kind of intervention of this work is non-pharmacological treatment, we will not deepen on pharmacological procedures.

When it comes to the studies that evaluate treatment for MCI the first problem we find is the inclusion criteria, this means that in different studies the criteria used to include patients and evaluate their level of MCI is different, which implies that the obtained results are difficult to compare. For this reason, taking conclusions becomes a hard task.

In one of the most comprehensive reviews we could find, a review of 41 studies concluded with a lack of evidences on efficacy in the recovery of capabilities of the patients. Nevertheless, a significant reduction on cognitive decline was pointed out.

The two big different types of treatment are computer assisted treatment and non-computer assisted treatment. Computer assisted treatments show a more effective results in global cognition, memory, working memory, and attention and helps improve psychosocial functioning, including depressive symptoms. Effects on other domains such as executive function and processing speed are negligible.



[a] IADL=Instrumental activities of daily living.

Fig. 17 Obtained results from a MCI training trial, extracted from article "Computerized Cognitive Training in Older Adults With Mild Cognitive Impairment or Dementia: A Systematic Review and Meta-Analysis"

As we can see in the **figure 17** is hard to obtain consistent results in the studies, this factor is represented by I^2, another issue of studies is the small size of samples, represented by K. Future trials should include larger sample sizes and directly compare CCT alternatives in order to optimize outcomes. Finally, there is insufficient data to determine whether training gains can be maintained over the long-term without further training,

So Computerized Cognitive Training (CCT) is normally carried out by computers or consoles like Nintendo Wii, but thanks to the new technologies that are appearing in the scene like VR, new CCT trainings are possible where a much more immersive experience will take better results.

The future of this kind of treatments goes through more interactive tasks for the patients, which have a bigger repercussion in their daily life. A consideration of more motivational and emotional trainings that can reinforce the cognitive training and keep the results in long term. This is the reason because the app we developed consisted on a daily task as going to the super market with a list of the products that the patient needs to buy.

As we have mentioned before, using a daily work as might be go to buy some products to the supermarket can be more motivational for the patient than the current training. All of these considerations mentioned leaded us to the development of a VR app called VR SUPERMARKET EXPERIENCE which follows a script created in collaboration with the doctors that guided us into the basic needs that a CCT should have.

5. Definition of the project

5.1. Scale of the project.

This project consists on a Virtual Reality application that allows improving the current techniques used in cognitive impairment treatments thanks to increase the immersive capacity of the technology. With a view to provide an easier way to the development of the project it was divided into two main parts: the modelling of the environment and app mechanics with the user interface

The main environment modelling priorities were to make the most realistic and user friendly spaces, since the target of the app are people who are not familiar with this tech. In order to do so, at the beginning was essential to work in collaboration with the Alzheimer's unit from Hospital Clínic, who are specialists on the treatment, they helped us to define the essential needs for the user.

The app mechanics consisted, as its own name denotes, on the mechanics of the app which embraces from the user interface to the different levels of difficulty. At this part the collaboration with the Alzheimer's unit from Hospital Clínic was important too, because in different meetings that took place at their facilities we developed the main script that contained the cognitive training exercises. But the most important help at this part was given by Visyon 360, the company provided us the hardware we needed, as the HTC VIVE, as well as the technic support due to our scarce knowledge of coding area.

The following scheme shows the workflow in order to build the app:

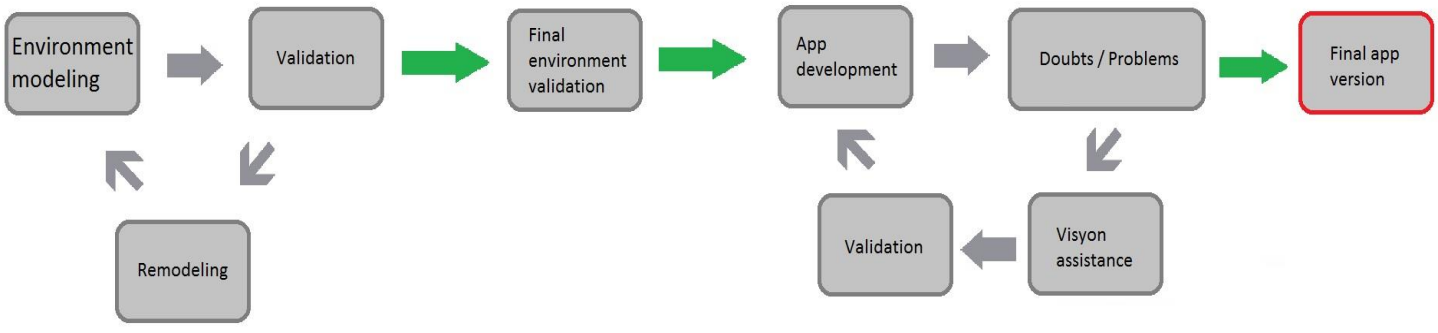


Fig. 18 Woorkflow scheme

5.2. Working methodology and planning

As we have commented previously, we took the decision of dividing this project in two parts, the environment modelling and the app mechanics development..

At the first part, as shown in the figure 18, Alzheimer's unit from Hospital Clínic helped us in the validation of the different environment components. This was more a design work where we used different programs to recreate the best training atmosphere, all of these programs used are explained with more detail at the "Design" section

In order to simplify the development of the app and difficulties we could find, we decided to appeal to methodology used by professionals; it consisted on the Getting Things Done (GTD) method.

The workflow of GTD consists on five different stages: Collect, process, organize, plan and do. This is used on the app development field because allows modulating the biggest problems into smaller ones with the purpose of simplifying and approaching goals in less time.

Each stage is represented by an inbox, each one represents a different stage of the task, thanks to develop and use the trusted system that deals with day-to-day inputs, an individual can free up mental space to begin moving up to the next level.

Because of our need to develop the app in one semester we decided to make a little variation of GTD methodology, this change consisted on simplifying the different stages a task could have. We decided to have three different stages: added to task list, in progress and done. Thanks to a free online tool named "Trello" we were allowed to carry out this task with a good synchronization in our team.

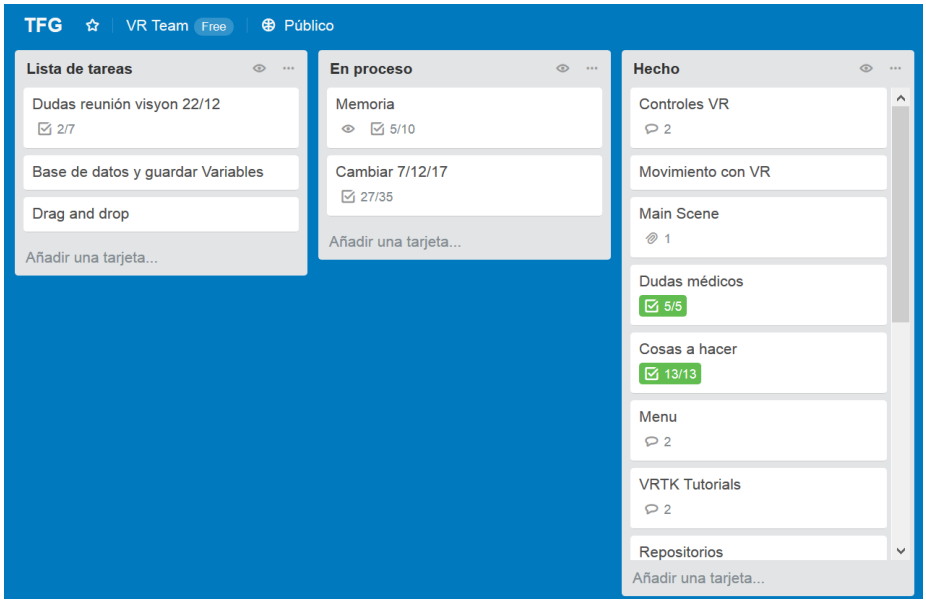


Fig. 19 Screenshot took from Trello's web page

At the app mechanics development Visyon 360 had an important participation because of their extensive experience in the area. Thanks to them we could develop the most user friendly possible app.

Due to the Getting Things Done method we were able to organise in a much more stepped way our Gantt diagram where is possible to see the progression of the whole project.

This diagram can be seen in the next page.

5.3. Gantt diagram

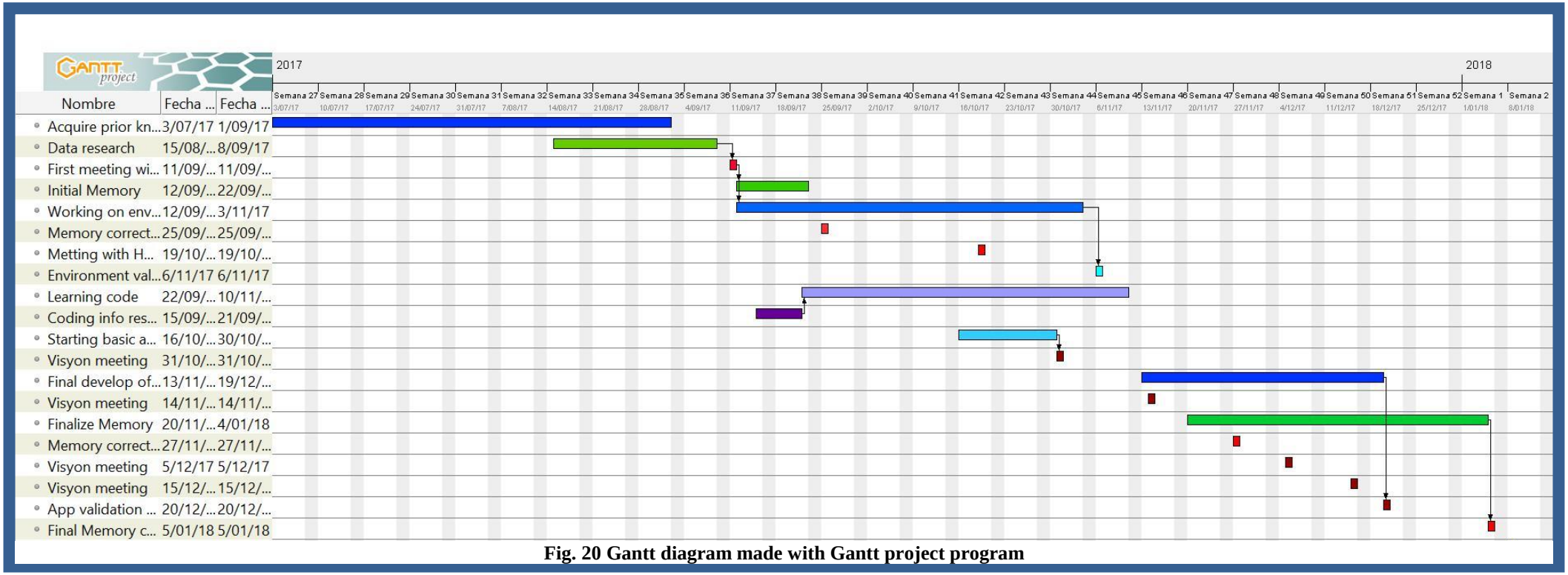


Fig. 20 Gantt diagram made with Gantt project program

5.4. App requirements

Before start developing the app a series of requirements where specified to be accomplished in the final version.

Hereunder these requirements are shown in two different kinds, Functional, which includes the software behaviour, and the Non-Functional ones, also known as quality attributes.

Functional requirements:

- Real time data collection.
- Different difficulties levels.
- Random shopping lists inside a range of pre-written ones.
- Most similar to real life products assets.
- Possibility of exit the game while is running.
- Good lighting inside the supermarket to make the most realistic immersion.
- Most user friendly interactive experience.
- Main menu only available to use for the doctor.
- Local database only accessible by the doctor and not online.
- Using numeric ID for recognising the patient.

Non-functional requirements:

- The application has to be developed for its use on VR HTC VIVE.
- The processing needs to be as lightweight as possible in order to satisfy the compatibility with the hardware loaned.
- User interface kept in the simplest way with the purpose of making the easier to use the app for the patients with a learning curve the most plane as possible.
- Stable final version free of bugs and with a robust architecture.

6. Chosen technology

6.1. VR Hardware

When it comes to the decision of which VR hardware to use is obvious the need of analysing different factors.

The possible choices of VR hardware are various, and thanks to the technical support of company Visyon, we have got to two of the top possible hardware in existence; The Oculus Rift and the HTC VIVE, both of them mentioned on the “Market analysis” below.

Both, the Oculus Rift and HTC Vive, are great VR hardware and the possibilities that they offer are immense, but they take different paths to conceive VR so we will try to explain our choice based on the differences between them.

First, we will start with the comparison between the displays, both of them have two OLED displays which are positioned just inches away your eyes, the two headsets have the same resolution, 1.080x1.200 each display, and the same 110-degree field of view so there is not difference in this part. Although they appear to be the same immersive experience considering that their displays each of them is conceived for two different kinds of immersion, the Oculus Rift was first conceived to play sat down, so in comparison with the HTC Vive, that is a far more involving experience thanks to the possibility of walking around and the bigger play area; This is a huge breakpoint for our decision of the hardware.

Fig. 21 Comparison of play set between Oculus Rift (left) and HTC VIVE (right)

Another breakpoint we found decisive were the device controllers, when the Oculus Rift first come with a XBOX controller but has the possibility of buying the new controllers from August 2017, the HTC Vive has from the beginning two ergonomic controllers, one for each hand, which besides having a great design and usability, have a much better tracking system that takes the immersion experience beyond any other seen before.

The last point to consider before taking the final decision is the possibility of suffering motion sickness. With the Oculus Rift, the possibility of suffering it is bigger since the experience happens being sited, with the HTC Vive is practically inexistent thanks to being stand up while user is immersed.

As a conclusion of the choice of hardware we can say that both, the Oculus Rift and the HTC Vive, offer different but really good immersive experiences, but for our needs the HTC Vive seems to be a better option because of the usability and the characteristics mentioned before.

6.2. Engine development

When developing a VR application, the choice of the development engine becomes one of the most important, even more when the developers are people with no experience on the field.

First of all the possible engines for this project are Unity 3D, Game Maker and Unreal Engine, a quick comparison of the three possibilities with our needs is made in the next table:

Properties	Unity 3D	Unreal Engine	Game Maker
Full 3D support	✔	✔	✘
Free version	✔	✔	✘
Available on windows	✔	✔	✔
VR support	✔	✔	✘
VRTK (virtual reality toolkit)	✔	✘	✘
3D design programs integration	✔	✘	✘
Help forums	✔	✔	✔
Visyon assistance	✔	✘	✘

Table 2 Engine development comparison

Thanks to this quick comparison we can discard Game maker as a development engine, in this moment the best option are Unity 3D and Unreal Engine.

Both of the engines offer really good development environments for users, but when it comes to an active community, Unity 3D takes advantage because of its forum, where there are a lot of repositories and documentation for solving possible problems and for beginner users like us, it is a really good point to keep in mind.

Both, the Unreal Engine and the Unity 3D, offer a free version, this version does not have all the features that can provide but for our needs is more than enough.

So far both engines show really similar results so now it's time to tilt the balance in favour of one or other.

The coding language used by Unity 3D is C# that in comparison with Unreal Engine, which uses C++, is much more beginner friendly language. Unreal also has the BluePrints system, which is a modular system to make the code implementation simpler, but due to its module based system is also a limitation.

Finally another factor to take in consideration was Visyon 360 support in the development of the application. The most part of the projects the company makes is developed by Unity 3D, so the assistance we can receive is considerably bigger.

So far we can conclude that both of the engines offer a good development environment for beginners, with Unity 3D the company support we can receive by Visyon 360 is bigger and the user forums Unity offers, these point made us finally decide on this development engine.

Unity offers a really self-adjustable environment where the user can adjust depending on the needs of this moment as we can see in the figure 22, where we were developing the "Main Scene" and our main needs were to have the inspector, scene view, hierarchy of the scene and the project assets in full view.

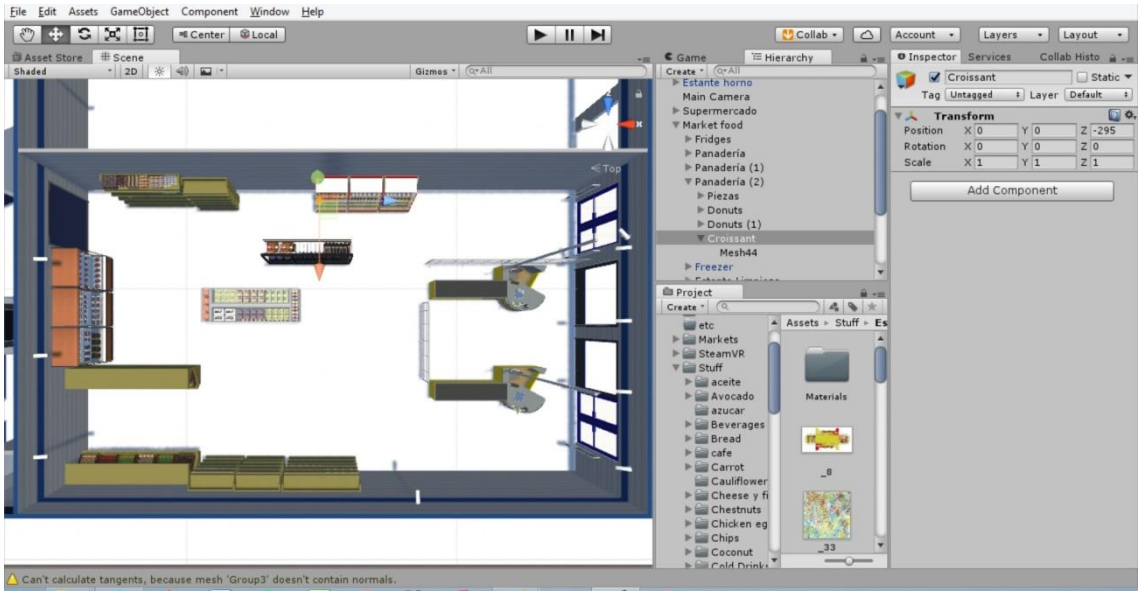


Fig. 22 Main screen Unity 3D (screenshot from Unity 3D)

Another feature of Unity 3D is the Collaboration Tool, which allows having a common project between a group of users and work in it at the same time and actualize through wireless connection.

Unity 3D also has its own animation module, as we can see in the figure 23, which allowed us to create different animations and integrate them in the scenes, since we needed to create different distractions for the patient to increase the level of difficulty in the experience.

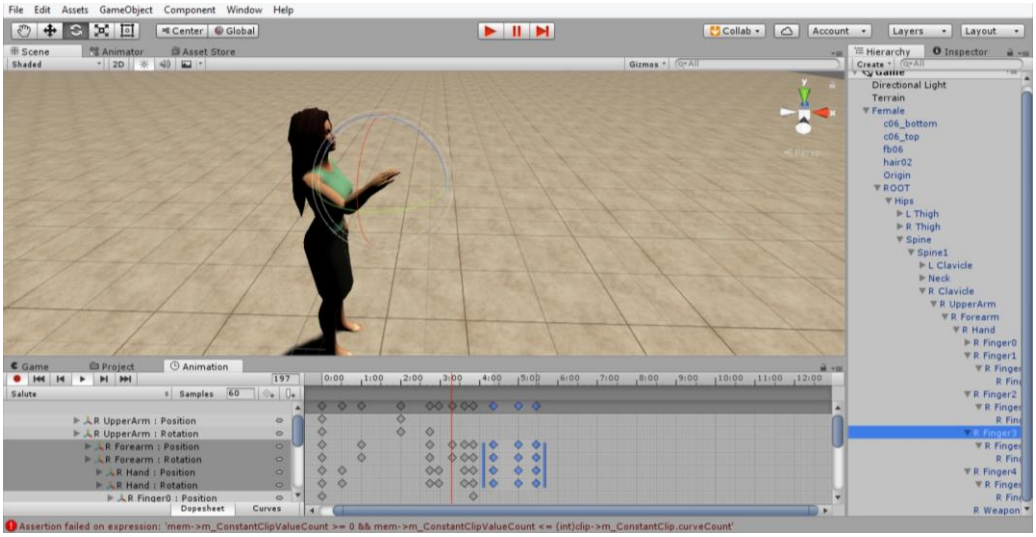


Fig. 23 Animation module (screenshot from Unity 3D)

The good compatibility of Unity 3D with the design programs and the different formats supported, also it helped with the recreation of a realistic environment.

The need of programs with a simple polygons mesh to avoid lag while running the experience made us to appeal to programs like SketchUp, to develop the supermarket environment, the possibility of exporting from SketchUp to FBX format facilitated this task.

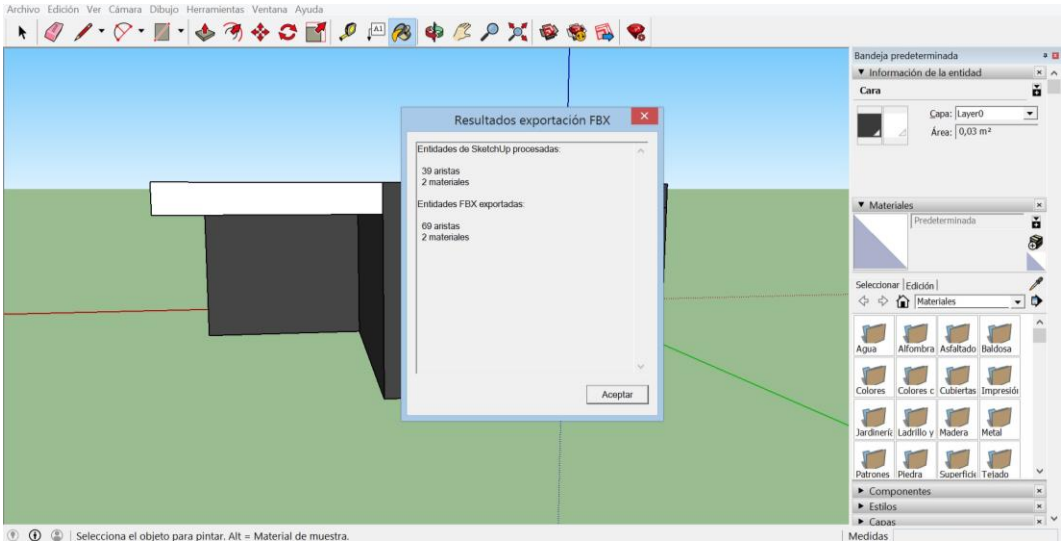


Fig. 24 Exportation message from SketchUp (screenshot from SketchUp)

6.3. Programing software

Unity has its own programing software included, Monodevelop, but in order to help us as code beginners we decided to use Visual Studio, a Microsoft program which offers a lot of features like smart auto-completed, smart syntaxes that were very useful.

Visual Studio admits many different programing languages like C, C++, C #, Python, etc...

We worked with C# because as explained before is easier for beginner and Unity also supports it.

Fig. 25 Visual Studio's logo

6.4. Extra resources

Because of the wide range of the project, we needed to use extra resources from third people to enhance the immersion experience. These resources have taught us basic ideas of technologies we are not related to.

Video creation

With the help of Andoni Sarabia and his experience on the content creation field we could create the explanation videos for the main scene levels, these videos are supposed to help patients to the better understanding of the commands and app mechanics.

Voice recording

In order to improve the understanding of the instructions given to a patient for any level, it was decided to add the audio explanations. In order to do so, and with the help of Belén Catalán and Andoni Sarabia, we recorded Belén's voices, after recording the audios a post-process was made to clear of possible impurities and to apply the presets for making the audio sounds like if it is reproduced from a speaker inside the scene.

7. Design

The supermarket and tutorial design are very important in a realistic simulation, because patients need it to improve their cognitive problems successfully. Also they should recognize the different products easily and quickly, for that reason these products have forms and textures very similar to a real supermarket.

Below, we are explaining the different ways to get the supermarket and tutorial assets.

7.1. Unity 3D tools

Downloading official unity assets

Unity has a wide range of assets which have helped us to design the atmosphere, usually they are free, but if they are not, we have used other websites or programs.

Official Unity assets are downloaded directly on Unity and do not need another modelling program.

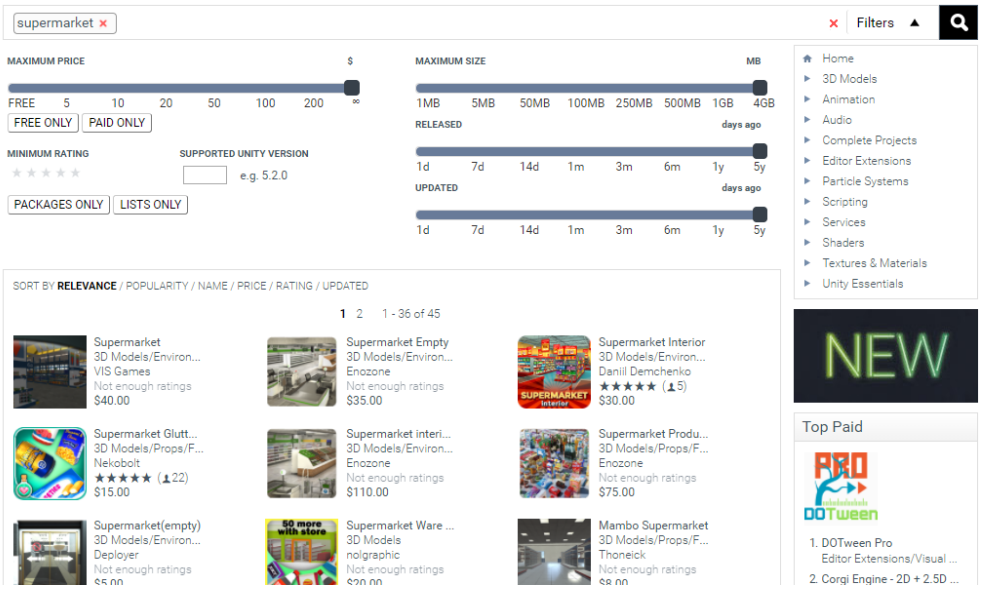


Fig. 26 Asset Store from Unity 3D (screenshot from unity 3D)

Creating objects with Unity

Building objects with different forms is easy on Unity, and they can be modified easily. However, the main problem is that you have to import the texture, sometimes we downloaded an image on internet but other times had to be designed with another program such as Paint, which is going to be explained in the following section.

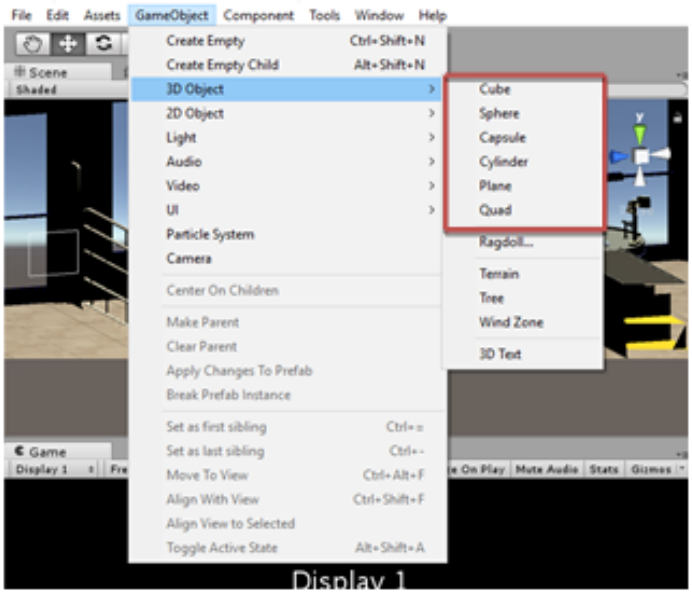


Fig. 27 Game Object creation menu (Screenshot UNity 3D)

7.2. Paint

A great problem is when you introduce an object on Unity and the texture is lost, at this moment you can find another object or do the texture with an image.

Sometimes it is necessary to do the second option because you cannot find another object on another website, and Paint is a well-known tool and almost every computer has it.

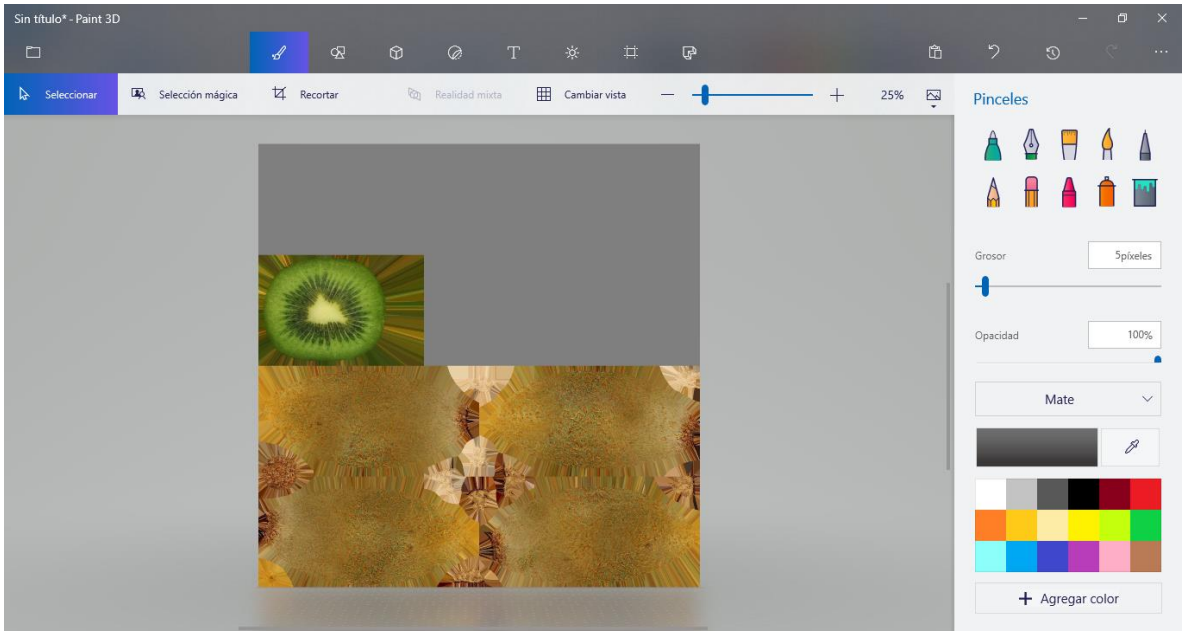


Fig. 28 Creating a texture with Paint (screenshot from Paint)

7.3. Websites

In several times, official Unity assets were not free or there are not the objects that we needed, but on internet exist websites which have a great library of 3d models and they are easy to use, such as:

Archive3d.net

Archive3d.net is an intuitive web, where if you want an object to your Unity project it can download the object (.3ds, the same type of unity assets) and import it on your asset folder. The process is very quick and easy.

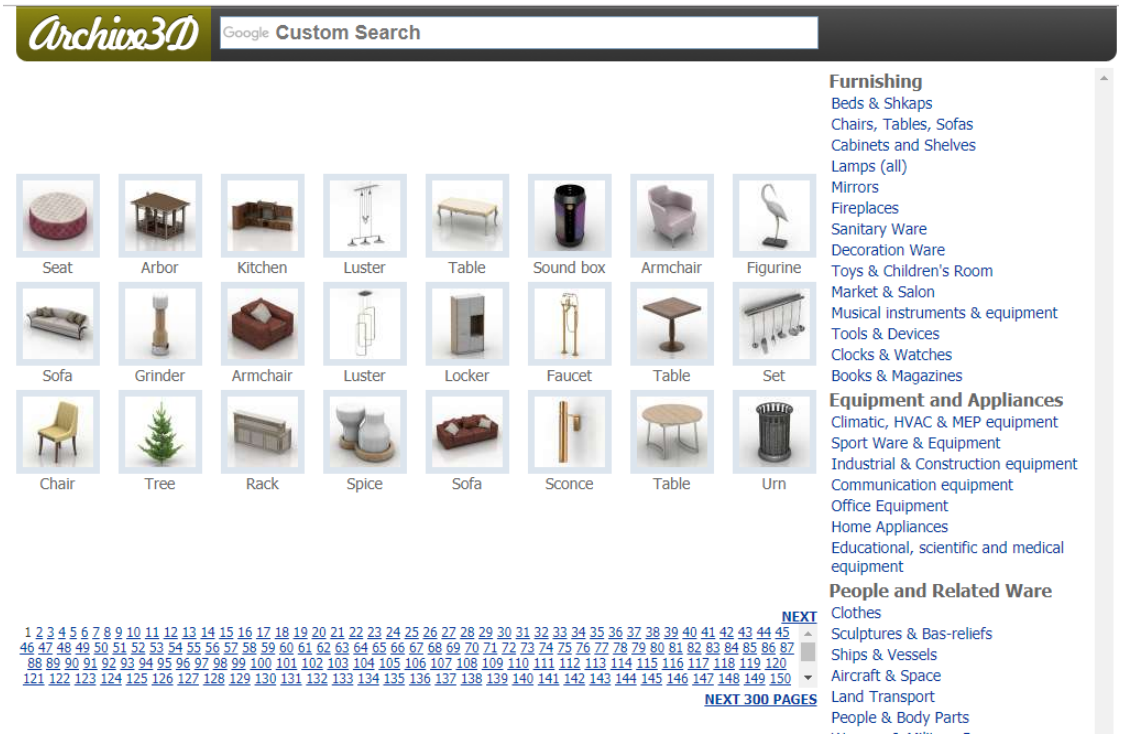


Fig. 29 Archive 3D repository (Archive 3D web page screenshot)

Sketchup and 3dwarehouse.sketchup.com

When it came to the process of importing models to Unity, after downloading them, Sketchup allowed us to open the models and clean them of possible imperfections.

Once the model was ready, Sketchup allows exporting the model with the desired format by choosing it on the export options, so it made quite easy to allow the compatibility.

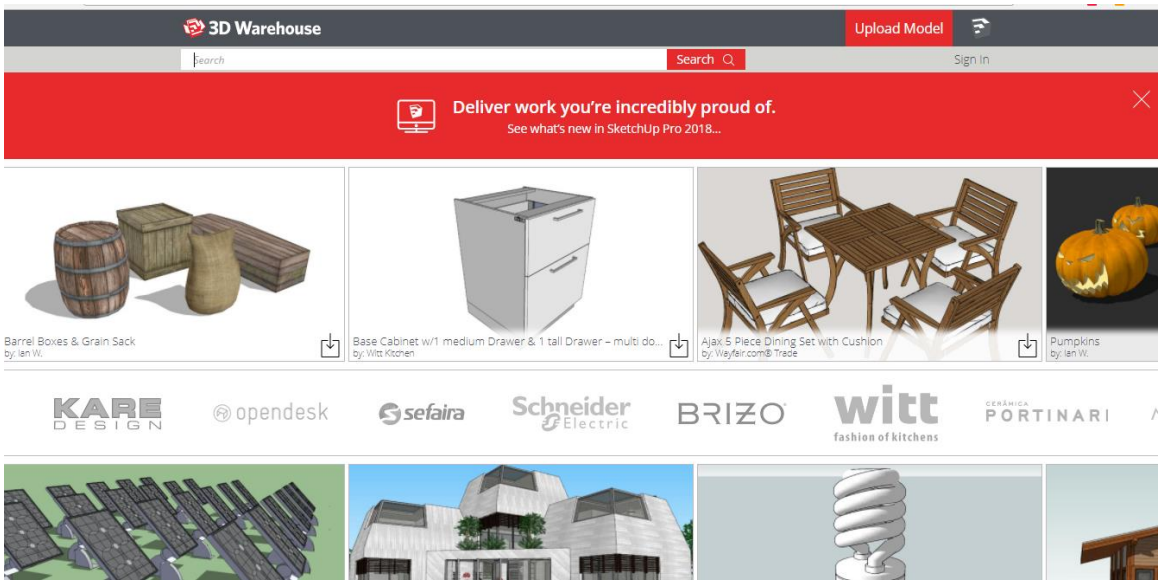


Fig. 30 3D Warehouse repository (screenshot from 3D warehouse web page)

7.4. Skanect

Skanect is a scanning program which uses Kinect to do it. Many objects are hard to design, for that reason we use Skanect with strange structures, such as in the tutorial explains how the patients can change the direction by moving their heads, this information is easier if they have an image as the next:

Fig. 31 3D scaned bust (Skanect screenshot)

Skanect is a cheap option to get great scans with a simple Kinect and the learning is very easy because you can find some tutorials on the Internet. This is the process:

Prepare

In this area you can modify settings to do the scan or load an old scan.

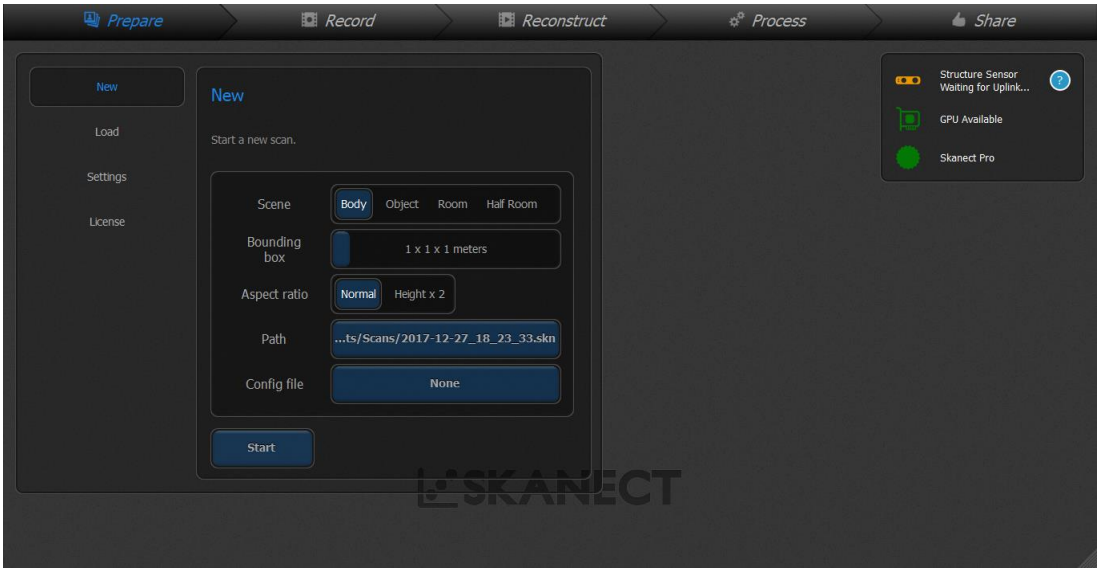


Fig. 32 Skanect Settings (Skanect screenshot)

Record

This is the most important area because you record the body/object, it is advisable that the process is slow and careful to achieve a great results.

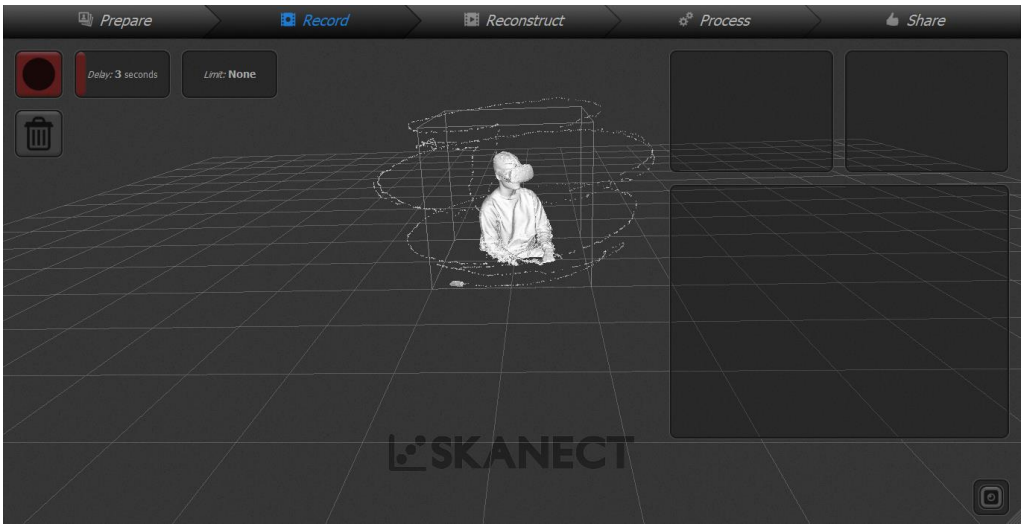


Fig. 33 Patron followed by the Kinect (Skanect screenshot)

Reconstruct

The result are shown, if it is not optimal you can reset or fusion de second point.

Process

It is possible that the result has errors, they can be modified in this area, such as holes or remove parts. A fundamental point is the body/object colours because it is a basic point to heighten the realism.

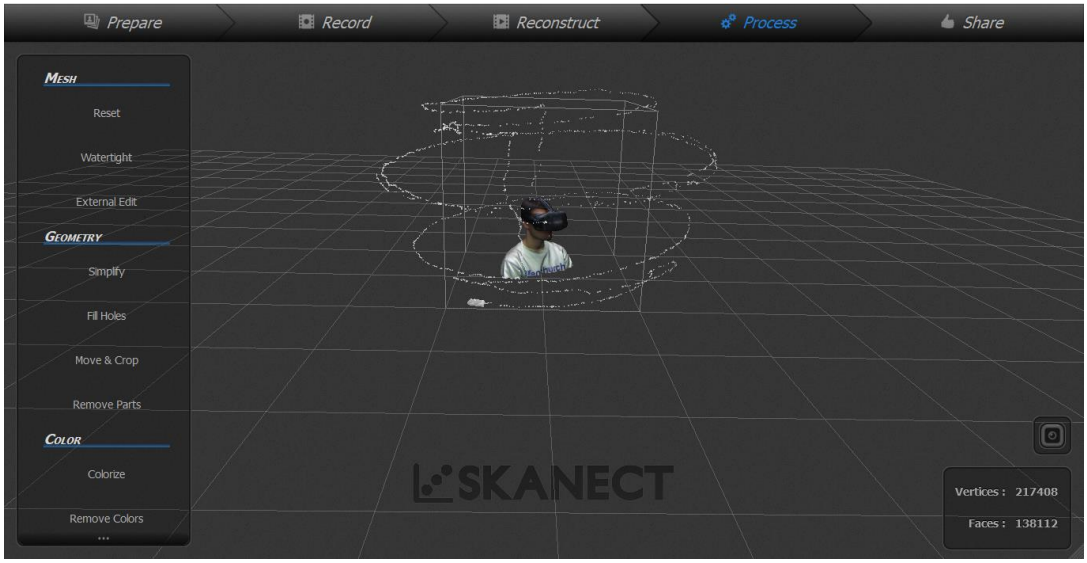


Fig. 34 Post-processing Settings (Skanect screenshot)

Share

The scan can be saved as .skn or export the model to work with it in other programs.

8. Menus and scenes description

8.1. Menus

Menus are very important in the application distribution, doctors will be able to select different mode games. These menus are in 2D so that they can choose easily.

On the other hand, when doctors are on the menu game patients will watch different sceneries thus they will be able to enjoy of Virtual Reality.

The menus are explained in more detail below:

ID menu

Doctors

The doctor of cognitive department will insert the ID of the patient, which will be a number to remain the patient's anonymity, the results will be kept in a database.

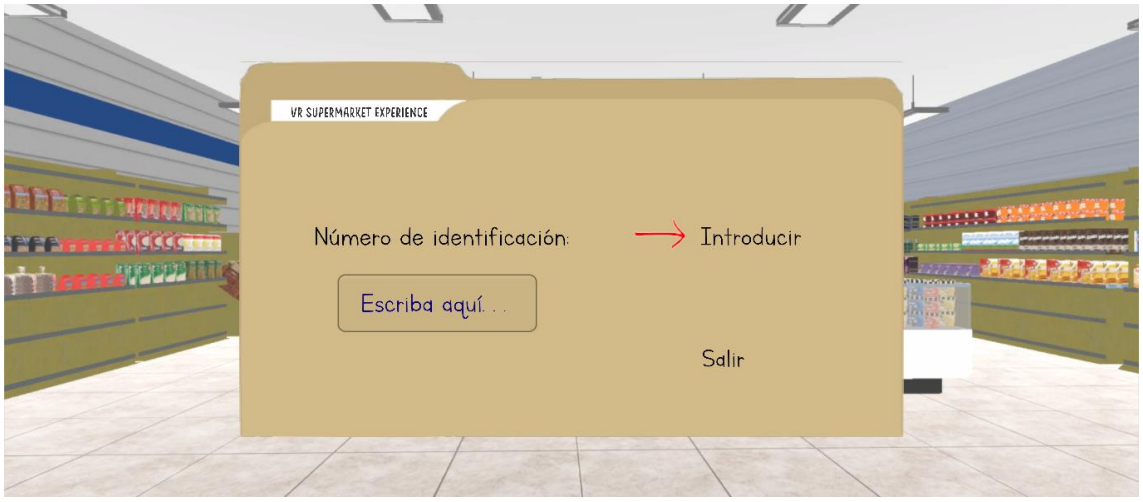


Fig. 35 ID scene (screenshot from Unity 3D)

Patients

At the same moment patients will watch other scenery, where the supermarket assistant is adding fruits to the fruit shelf.

Fig. 36 Waiting scene (screenshot from Unity 3D)

Main Menu

Doctors

The second menu gives you the possibility to choose between two game modes, also doctors can change the ID or leave the application.

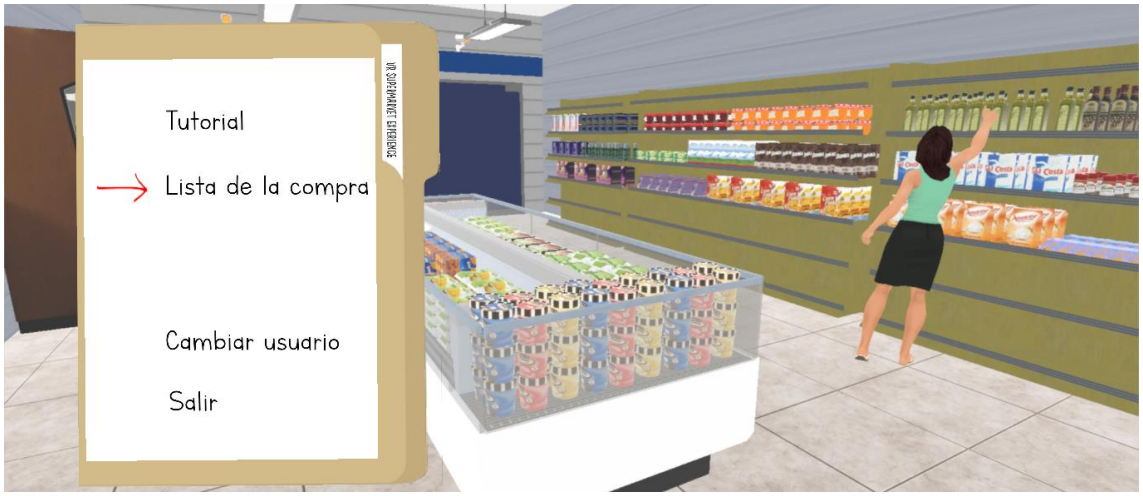


Fig. 37 Choosing modes scene (screenshot from Unity 3D)

Patients

At the same moment the patients will watch other scenery, where the supermarket assistant is adding oil bottles to the shelf.

Main scenes menu

Doctors

The main scene has different difficulty levels, which will depend on the patient's ability. Also there are videos which will help the patient to learn the new commands of the different levels.

Fig. 38 Choosing levels scene (screenshot from Unity 3D)

Patients

The scenery is the same as the previous menu.

Final menus

Doctors

After every mode or game level doctors will be able to do repeat to the patient it or leave this menu to return to the main menu.

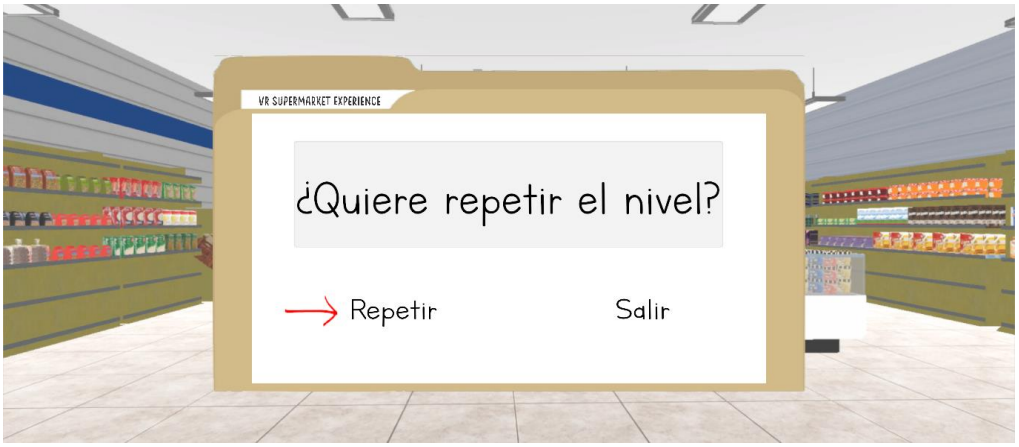
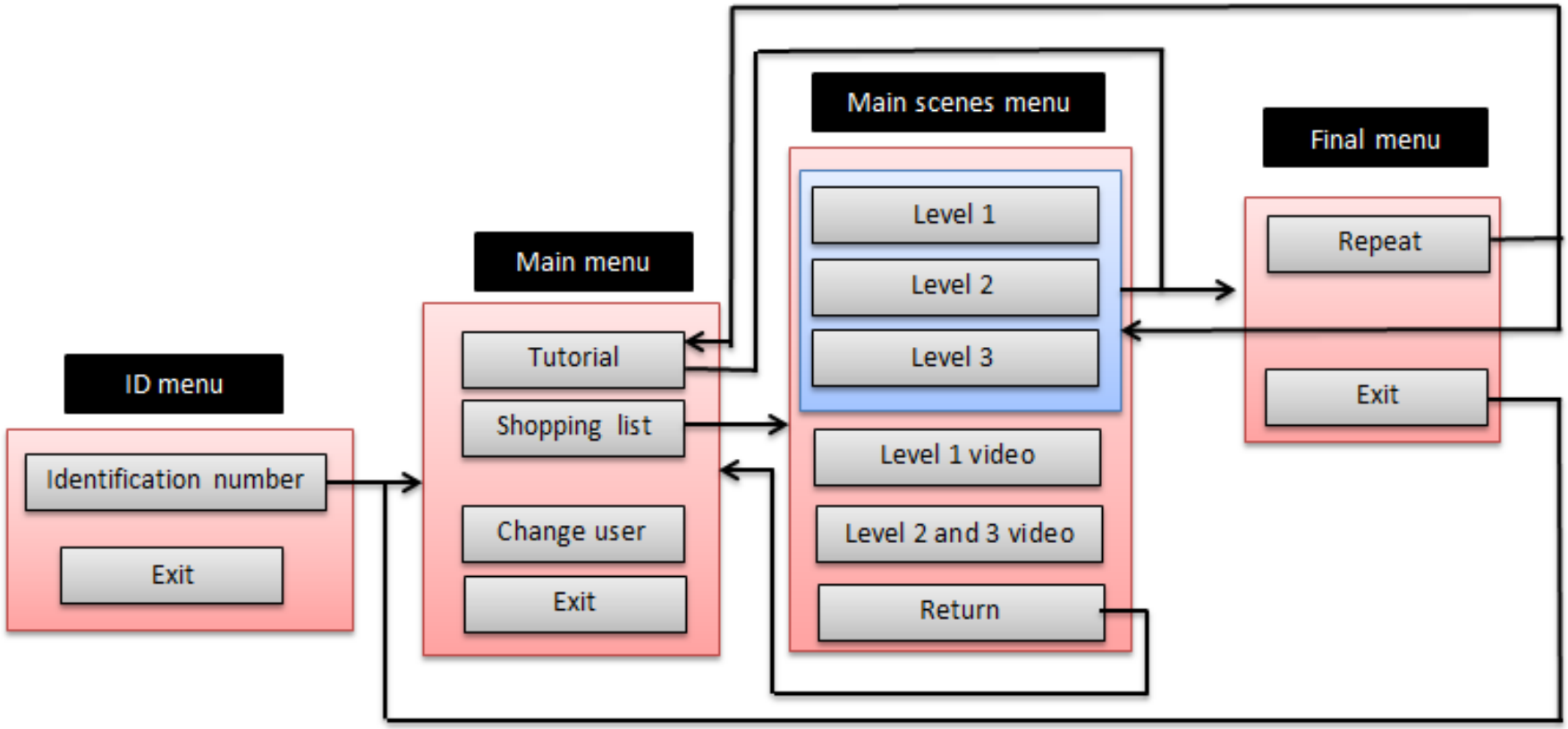


Fig. 39 Repeat level scene (screenshot from Unity 3D)

Patients

At the same moment the patients will watch other scenery, where the supermarket assistant is going around the fridge.

Menus flow diagram:



8.2. Tutorial

Virtual reality can have a difficult adaptation when you are used to this technology, for this reason we believe that a tutorial can help to speed up this process.

Basic commands are explained through videos and voice, we want that every step of it is understood quickly and easily.

In this way the voices have been designed between Belén Catalán, pedagogue, and our team, she has proposed substantial improvements in this area, as well as Belén proved the tutorial voices.

To sum up, the tutorial of this app was thought to know the basic commands as if it was other level of the game since it is easier to learn.

These commands are explained in different zones of the tutorial, below you can see step by step every part of it:

In the first part of the tutorial patients are in a room where a voice welcomes them, then it explains them how they should change direction when users plays. It is going to explain with three spheres which change the colour (green) when they look at them.

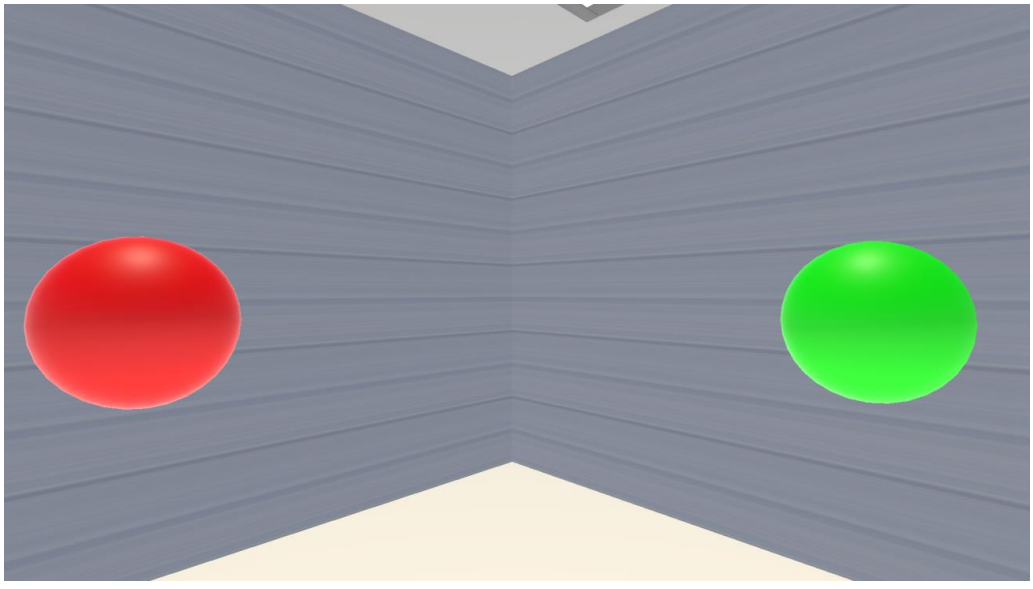


Fig. 40 First part of the tutorial (Tutorial screenshot)

Secondly, patients are going to learn how they should move with the right trackpad because the left trackpad is disabled to ease the learning, they can leave the first room at this moment, the door shown in the figure 40 will be opened when patients approach it.

Fig. 41 Moment before the door gets opened (Tutorial screenshot)

Following the last two points they should follow a little circuit to check the knowledge and they will arrive to the next part of tutorial.

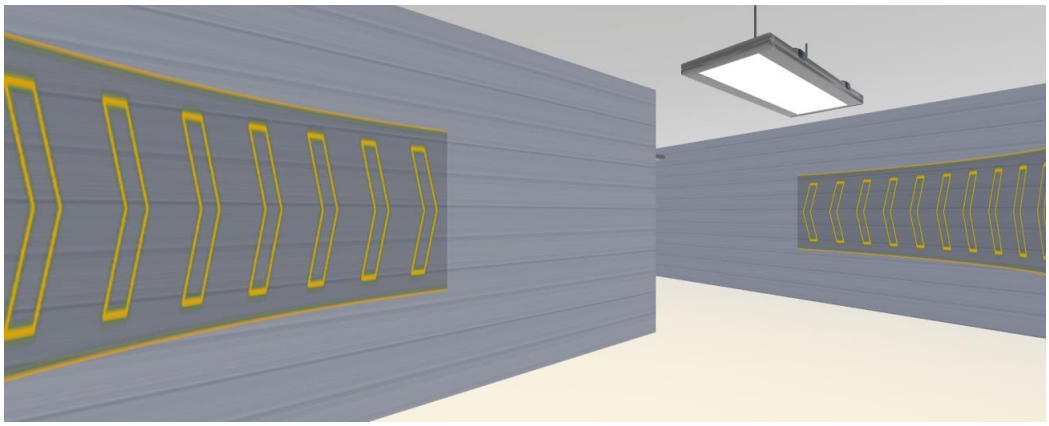


Fig. 42 Path to follow by the patient (Tutorial screenshot)

In the second room users are going to learn how they have to drag and drop the different objects, they must pull the index trigger.

To prove it they should place different objects in the shelving, there are three heights.

Fig. 43 Tutorial final scene (Tutorial screenshot)

Finally, if the doctor believes that the patient has to repeat the tutorial when the point four finishes patients can do it.

At any part, when patients carry out this part correctly, they will be congratulated because we believe that it creates a good feeling between patients and this application.

8.3. Main scenes

The three main training scenes are structurally similar with a few differences that will be explained below.

After a welcome audio clip to the level at the level one, the patient will be able to start the training.

Fig. 44 Starting moment of the level one (level on screenshot)

At this point the first difference takes part, unlike level one the shopping list will not always be visible, it will be shown for a stipulated time and will disappear after that.

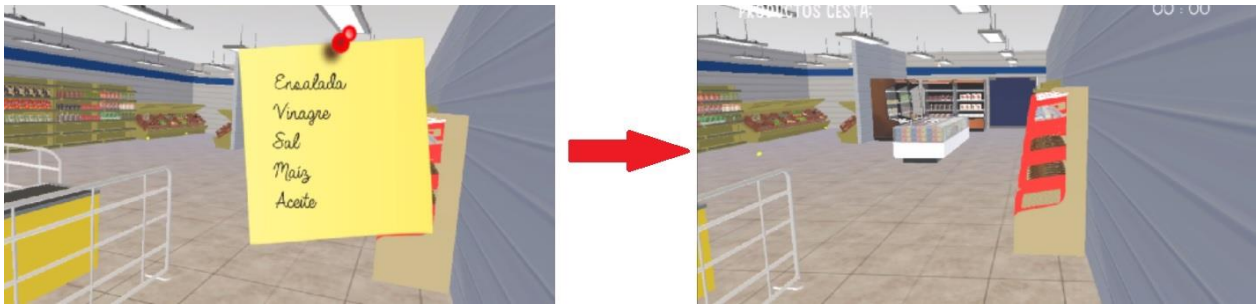


Fig. 45 The shopping list disappears after the stipulated time

The shopping basket will always be attached to the left hand of the patient for avoiding distractions of losing it.

All the products of the supermarket are able to be dragged because of the decision of adding products to the basket that are not on the list. For introducing the objects on the basket the user will have to introduce them on it and press the left trigger. Once the product is placed on the basket the product name will appear on canvas.

Fig. 46 Moment where the canvas is fully appreciated

Level two and three were decided to include distractions in order to increase the difficulty. At level two the distraction is a woman who salutes the user at the entrance, at level three a second distraction is included, where the woman says a promotional offer.

Fig. 47 promotions distractions humanoid

On the levels two and three, as explained before, the shopping list will not always be visible. The patient will consult it by pressing the Menu button, for the consulting time the movement will not be allowed to ease the concentration on the list.

Fig. 48 Path to follow for making appear the shopping list

In any time during the training the user will be able to exit the scene by two ways. The first one, if the user needs to stop the training in the middle of it, is by pressing the Menu button of the right controller and selecting Yes by pointing and pressing the trigger of the right controller.

The other way to finish the training is to exit by the exit door and will appear the same canvas that when pressing the Menu button.

Fig. 49 Finishing level canvas

9. Development on Unity 3D

9.1. Menus

9.1.1. Common elements

In this section we are explaining the common parts of the development of menus in Unity 3D.

In every menu the first step is to create a canvas, which we are going to work in the next steps.

(GameObject -> UI -> Canvas)

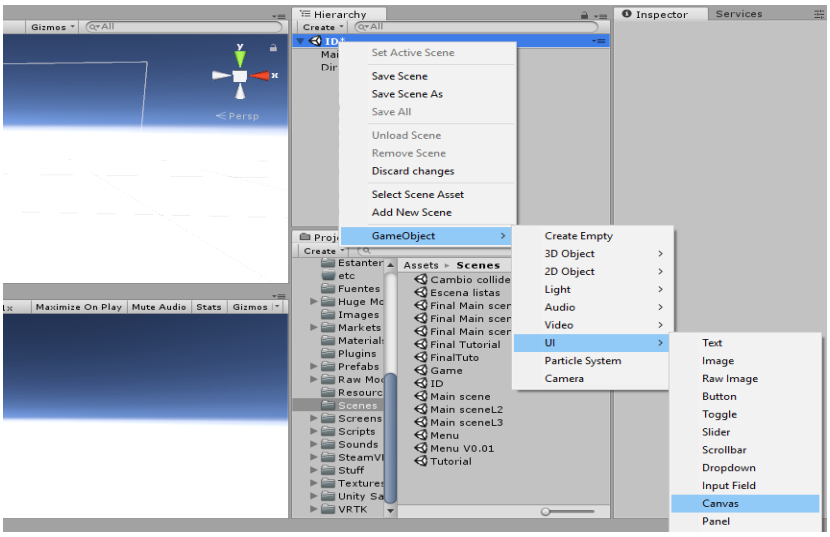


Fig. 50 Path to follow for create a canvas (Unity 3D screenshot)

This canvas is going to be shown in 2D, for this reason the render mode is Screen Space – Overlay, this option allows that the canvas can be watched by the doctors on the computer screen only.

Other important point is de UI Scale Mode, which determines how UI elements are scaled on Canvas, in this application it is “Scale with Screen Size”. Reference resolution is relevant to show correct dimension on different screens, in VR Supermarket Experience always work with a reference resolution of 1920 x 1080.

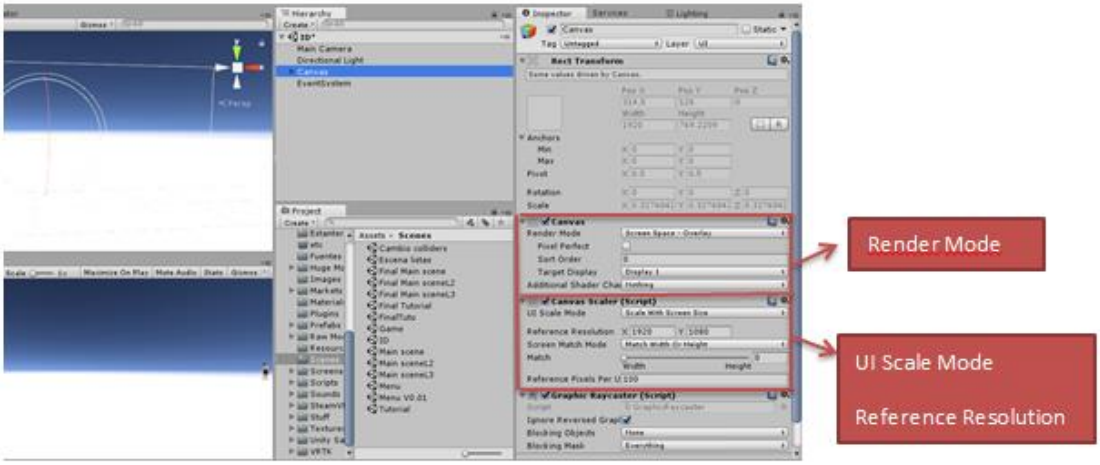


Fig. 51 Settings for the canvas (Unity 3D screenshot)

Inside every canvas there are several UI elements, but we can always watch RawImage, a panel and several buttons, which are going to be explained in the next section.

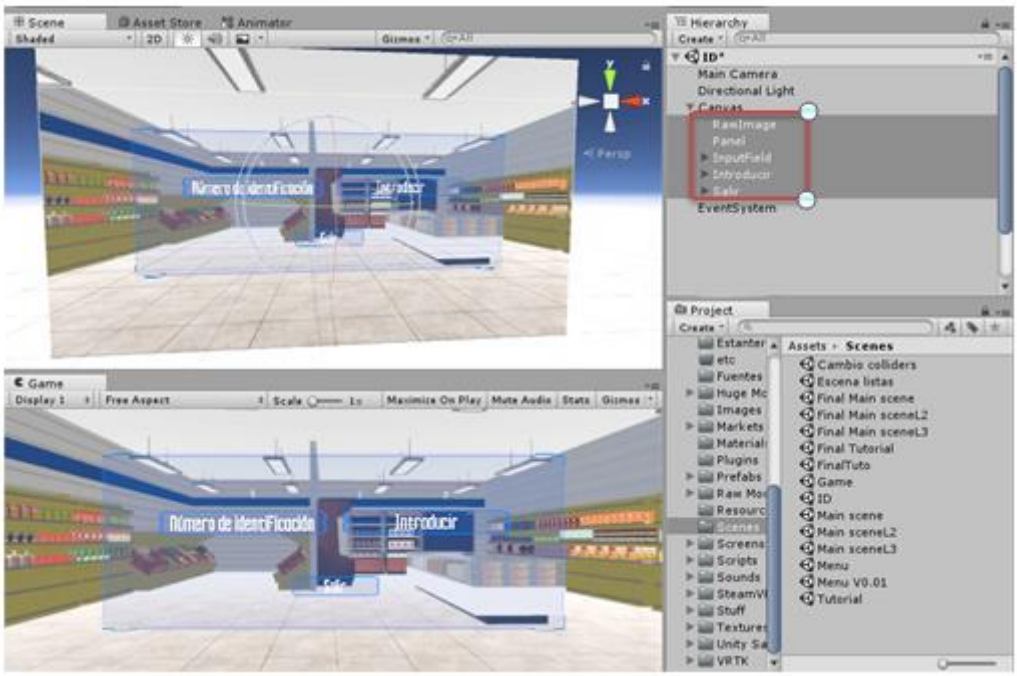


Fig. 52 Canvas components (Unity 3D screenshot)

As we have explained previously, patients won't watch the canvas, they will be in a new scenery where there is an animation and they will be able to experiment Virtual Reality at any moment. For this reason, we have added the same supermarket scene in every menu but we have changed the animation depending on the app menu.

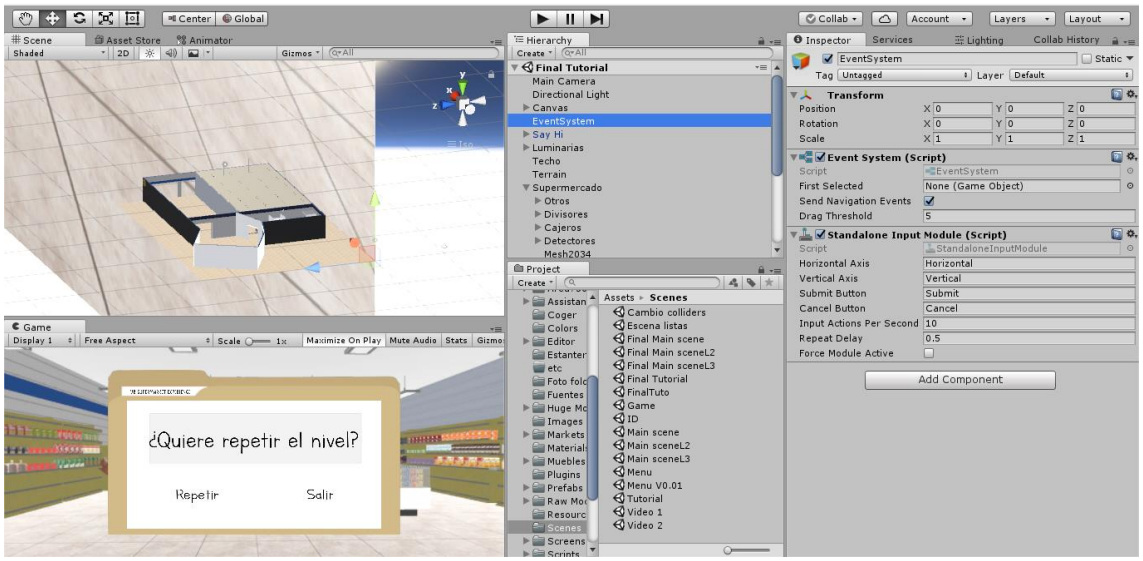


Fig. 53 Canvas working space (Unity 3D screenshot)

9.1.2. Different elements

In this section we are explaining the different parts of the development of menus in Unity 3D.

ID menu

The main differences are the buttons, the "Identification number" button is used to avoid introduce the name of patients, because this information is confidential, and the "Introduce" button keeps it in a data base.

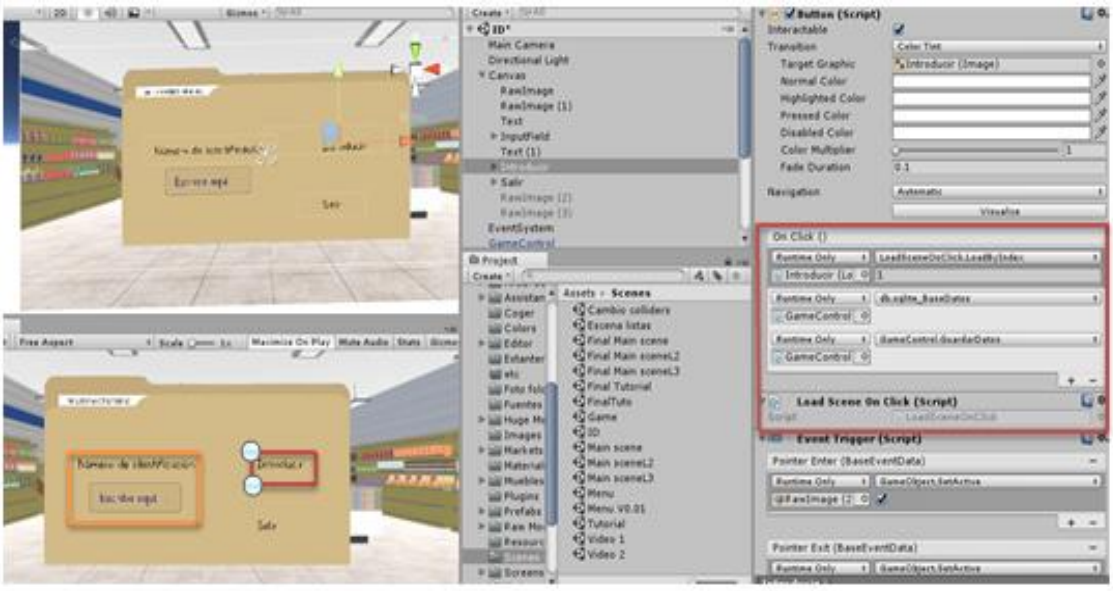


Fig. 54 Actions when the button is clicked (Unity 3D screenshot)

In the "Introduce" button the "Load Scene On Click" script has been added, which loads another scene when it is clicked.

```
public class LoadSceneOnClick : MonoBehaviour {

    public void LoadByIndex(int sceneIndex)
    {
        SceneManager.LoadScene(sceneIndex);
    }
}
```

But so that this button works correctly we have to drag the script to “On click ()” section and write the scene number which we want to load. To enumerate the scenes we have to follow the next path:

File -> Build Settings

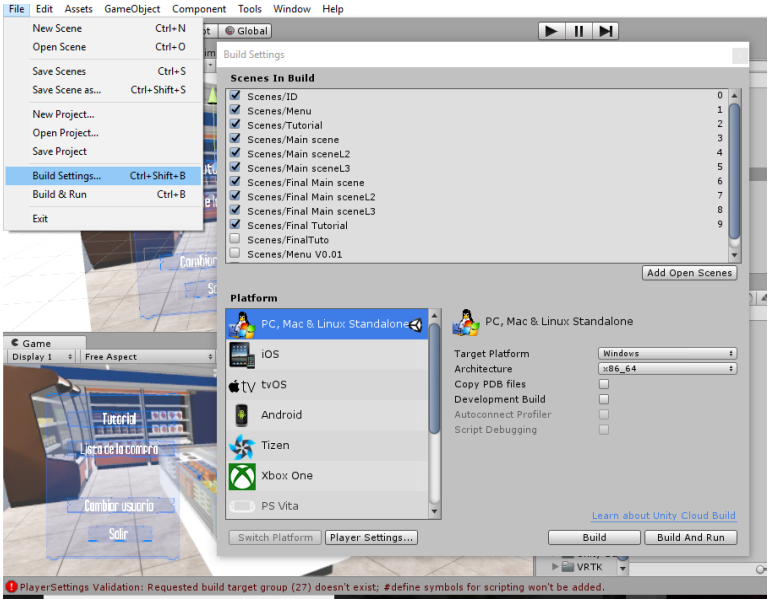


Fig. 55 Building Settings for the app (Unity 3D screenshot)

When you click the “Exit” button the application is turned off, it happens by the following code:

```
 public class QuitScene : MonoBehaviour {

    public void Quit()
    {
#if UNITY_EDITOR
        UnityEditor.EditorApplication.isPlaying = false;
#else
        Application.Quit ();
#endif
    }
}
```

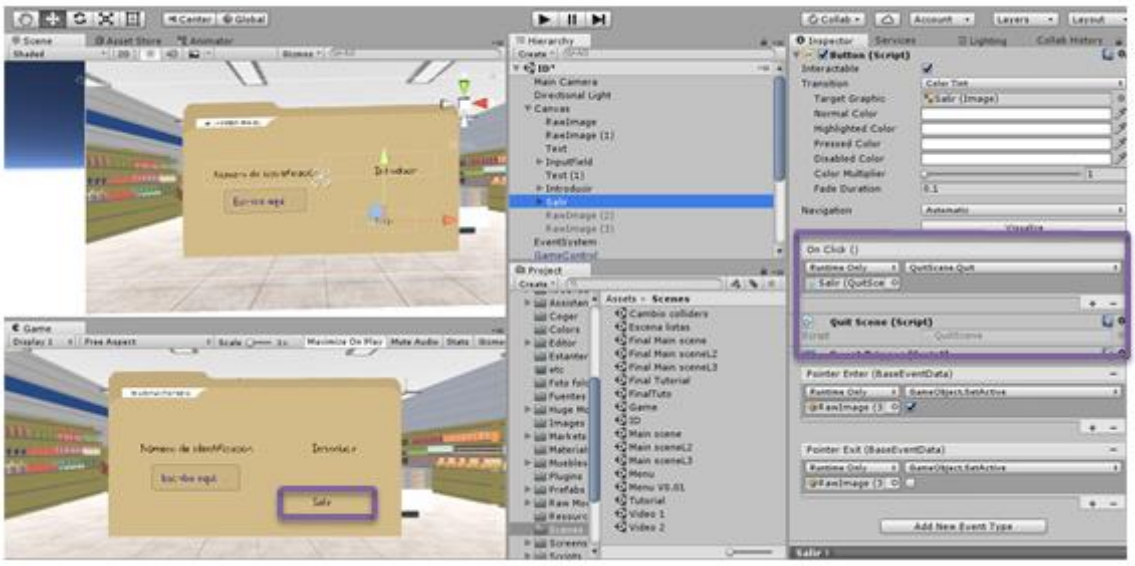


Fig. 56 Actions when the button is clicked (Unity 3D screenshot)

Main Menu

The "Shopping list" button is different than the others because it does not load a new scene, it hides this menu and the "Main scenes menu" appears, it loads the second menu quicker. To do it the path to follow is the next:

Drag "Menu" or "Main Scene Menu" to "On click()"section -> No function -> Game Object -> Set Active (bool)

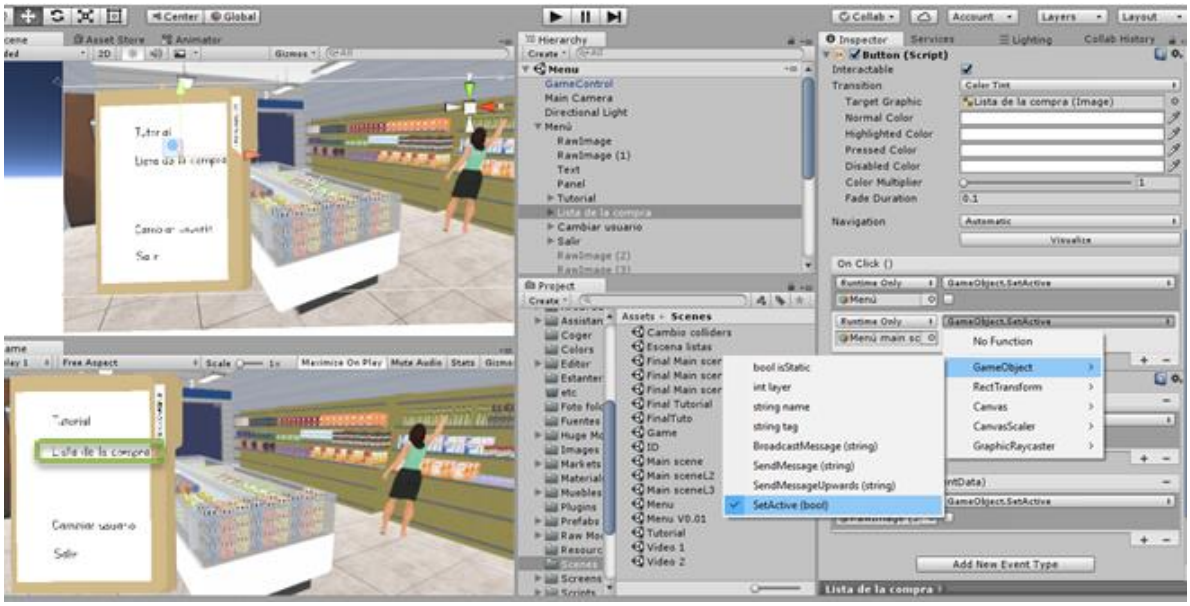


Fig. 57 Path to follow for interacting between canvas (Unity 3D screenshot)

Main scene menu and final menu

These menus work the same way as the previous ones, since their buttons only load other scenes or menus.

9.2. Tutorial

With the main purpose of an arranged learning curve the use of scripting was crucial so it took us to use C# basic code, it will be explained below.
The command *void start* is called when the scene is loaded and in our case was used in order to have the main features we needed at the beginning of the scene. At the example shown, the controllers are not activated to not allow the movement of the user and the clips are defined for when we need to play them.

```
void Start () {

    LeftController.SetActive(false);
    RightController.SetActive(false);
    audiosourceinicio = GetComponent<AudioSource> ();
    audiosourceinicio.clip = audioinicio;
    audiosourceinicio.Play ();
```

In order to know when the user is looking at the sphere we used the vector raycast. It combined with raycastHit hit let us confirm it.

Due to the need of a regular periodicity check if the user is looking at the sphere it was used the command void update, which is called on each frame and this gave us the accuracy needed.

```
void Update () {
    RaycastHit hit;


    if (Physics.Raycast (this.transform.position, this.transform.forward, out hit, distanc
eToSee)) {
        if (hit.collider.gameObject == totem1) {
            totem1.GetComponent<Renderer> ().material.color = Color.green;
```

Once is confirmed that the user is looking at the sphere, the boolean becomes true and it changes the colour to green and starts the next audio clip with the following instructions, at the same time a Coroutine is initialized, which does not allow the next sphere to appear until the clip is finished.

```
tot1 = true;
            counter = counter + 1;
            if (counter == 1) {
               StartCoroutine(tiempototem1());
               audiosourceizquierda.Play ();
            }
IEnumerator tiempototem1()
  {
     yield return new WaitForSeconds (audioizquierda.length);
     totem2.SetActive(true);
   }
```

When this first part of the tutorial is finished, the controllers get active allowing the user to start moving around the scenario.

The next part is to cross through the door, to avoid the user goes back and has distractions, the doors will get closed once the empty game object shown in the **figure 57** is trespassed.

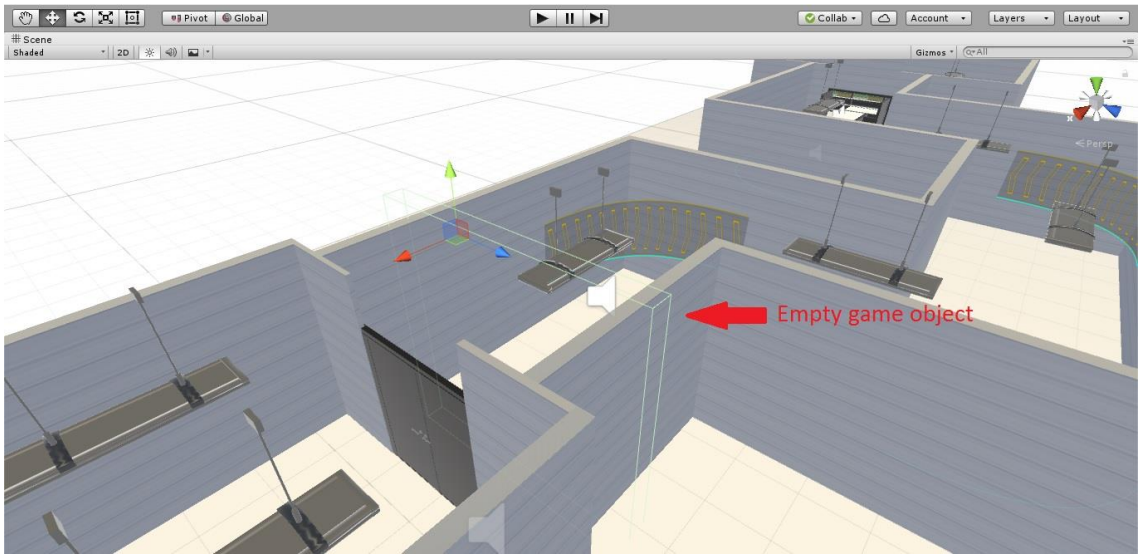


Fig. 58 Trigger collider (Unity 3D screenshot)

```
void OnTriggerEnter (Collider collider)
   {
      animation1.SetBool ("close",true);
      animation2.SetBool ("close",true);
      contador = contador+1;
      if (contador == 1) {

         audiosourcepuerta.Play ();
```

```
    audiosourcedireccion.Play ();
}
```

Once users reach the final room, where they will be taught to drag the products and drop them in the desired place. For this, the Snap Drop Zone plugin from VRTK is used, this plugin allows to have a zone where the product is standing and once it is dragged a new one appears to fill the space left for the previous one.

Another feature of this plugin is to have an empty zone with the drawn silhouette of the object which needs to be placed on it. The two explained possibilities are shown in the **figure 58**.

Fig. 59 Snap drop zones (Unity 3D screenshot)

Once the 3 objects are dropped in the right zone the tutorial is finished.

9.3. Main scenes

Again, at this scene, the use of C# had an important part on the development of it.

For starting, at the beginning of the scene, with the help of the commands void start and Coroutine which were introduced at the previous section, all the variables, clips and animations are defined and this allows a good set up for the scene development.

At the level one there is only one audio-clip before the training starts and the user can begin to move, otherwise at the levels two and three after the welcome audio-clip the shopping list will appear in the middle of the display for the memorization of it. The code shown below belongs

to the levels two and three since is more complicated in comparison with the code of the level one.

```
void Start () {
      audiosourceinicio = GetComponent<AudioSource>();
      audiosourceinicio.clip = AudioInicio;
      audiosourceinicio.Play();
      LeftController.SetActive(false);
      RightController.SetActive(false);
      StartCoroutine(tiempoaudioinicial());

   }
   IEnumerator tiempoaudioinicial()
   {
      yield return new WaitForSeconds(AudioInicio.length);
      Nota.SetActive(true);
      StartCoroutine(tiempovernota());

   }
   IEnumerator tiempovernota()
   {
      yield return new WaitForSeconds(TiempoVerNota);
      Nota.SetActive(false);
      LeftController.SetActive(true);
      RightController.SetActive(true);
      Timer.SetActive(true);
      ProductosCesta.SetActive(true);
      Objetos.SetActive(true);

   }

}
```

The different lists that will appear on each level are randomly picked from an array which contains the different lists that the doctors provided to us.

```
pet =Random.Range(0,listas.Length);
                    print (listas[pet]);

                    Lista.text = listas [pet];
```

At the time of introducing the products in the basket the code carries out the next:

First of all, with the void *OnTriggerEnter* checks when there is an object introduced in the basket, after that, the code reads the Tag and check if is on the list, depending on whether is right or wrong adds (+1) to the valid counter or wrong counter.

```
void OnTriggerEnter (Collider other) {

    if (other.tag != "Untagged") {

        other.GetComponent<Rigidbody> ();
        //other.attachedRigidbody.isKinematic = true;
        other.gameObject.transform.parent = transform;

        if (pet2 == 0) {

            if (other.tag == "Garbanzos") {
                V1 = 1;
                print (V1 + ("V1"));
                print (Valid+("Valid"));
            }
            if (other.tag == "Aceite") {
                V2 = 1;
                print (V2 + ("V2"));
                print (Valid+("Valid"));
            }
            if (other.tag == "Salt") {
                V3 = 1;
                print (V3 + ("V3"));
                print (Valid+("Valid"));
            }
            if (other.tag !="Garbanzos" && other.tag !="Aceite" && other.tag !="Salt")
            {
                NoVal = NoVal + 1;
                print (NoVal+("NoVal"));
            }
```

The next step is to disable the GameObject for avoid any interference during the training and writing the Tag to the products added on the basket list shown on the display.

```
other.gameObject.SetActive (false);

        ObjetosCesta.text= string.Format("\n"+other.tag+ObjetosPrevios2);
```

```
ObjetosPrevios= ObjetosCesta.text;
ObjetosPrevios2= ObjetosPrevios;
```

Fig. 60 Products on the Basket list canvas (Unity 3D screenshot)

The referenced code is activated when the trigger of the left controller is activated thanks to the Controller events plugin provided by VRTK.

Fig. 61 Controller vents plugin (Unity 3D screenshot)

At levels two and three for consulting the shopping list the user has to press the Menu button of the left controller, this will make the list appear in the middle of the display for the stipulated time. During this time the movement will disable to avoid distractions.

For each time the list is consulted the script adds to the counter (+1).

```
public void Contador () {
    LeftController.SetActive (false);
    RightController.SetActive (false);
    Nota.SetActive (true);
    ContadorConsulta = ContadorConsulta + 1;
```

```
        ListaCesta.SetActive (false);
        NombreProductosCesta.SetActive(false);
        StartCoroutine(TiempoMandos());
    }
    IEnumerator TiempoMandos()
    {
        yield return new WaitForSeconds(TiempoVista);
        LeftController.SetActive (true);
        RightController.SetActive (true);
        Nota.SetActive (false);
        ListaCesta.SetActive (true);
        NombreProductosCesta.SetActive(true);
    }
}
```

For the Exit menu the VRTK plugin named before, Controller events, is used. When the Menu button of the right controller is pressed or the user leaves for the exit door a different canvas that the one which is normally visible and the pointer will get activated, the normally visible one and the movement will get deactivated.

10. Commands

In this section, the application commands will be explained. It should be noted that tutorial and every level of the main scene have different commands but they follow a standard.

The basic commands will be taught in the tutorial through audios and several tests, which will help to improve quickly and entertaining for users.

However the different levels of the main scene have some different commands, but in this case the explanation will be through two videos, which are available in the main scene menu.

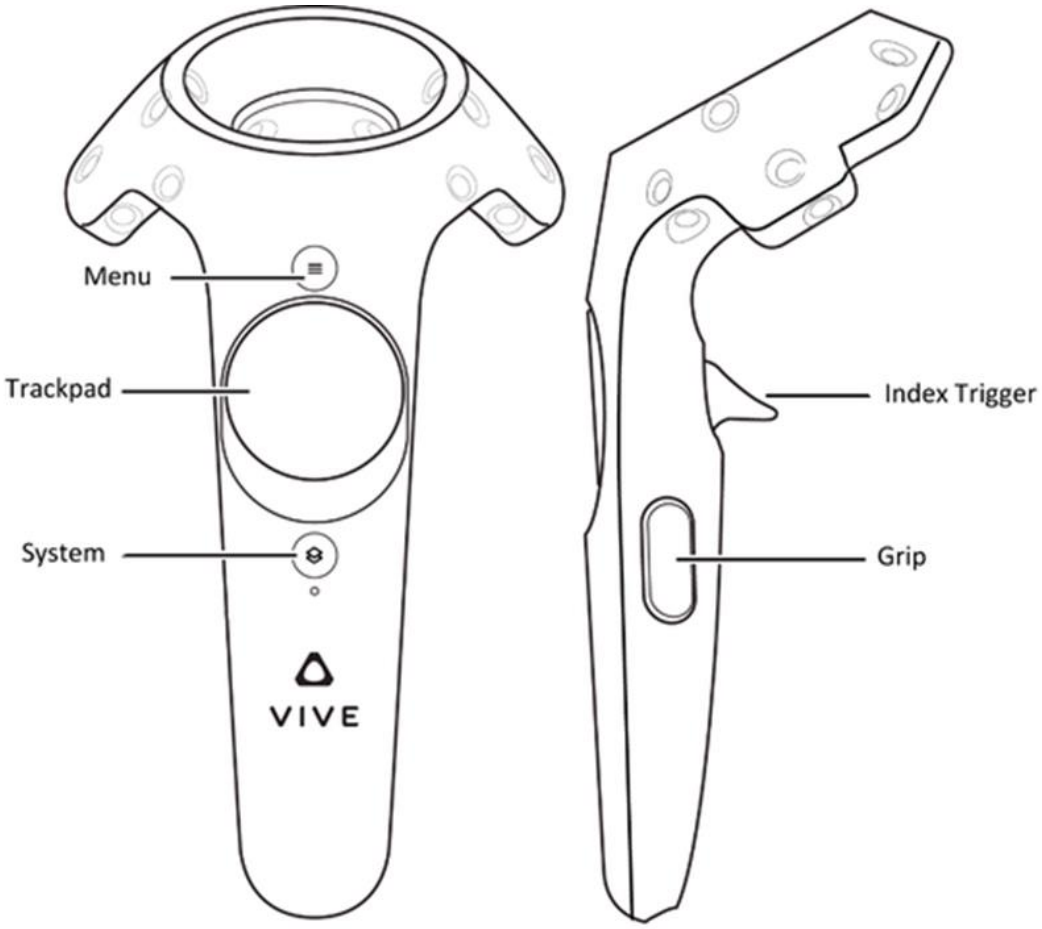


Fig. 62 Commands actions

Tutorial

	Left controller	Right Controller
Menu	Official Steam menu	Official Steam menu
Trackpad	Disabled command	Modify the direction
System	Disabled command	Exit menu
Index trigger	Disabled command	Pick up objects
Grip	Disabled command	Disabled command

Table. 3 Tutorial commands

Main scene, level 1

	Left controller	Right Controller
Menu	Official Steam menu	Official Steam menu
Trackpad	Disabled command	Modify the direction
System	Disabled command	Exit menu
Index trigger	Active basket	Pick up objects
Grip	Disabled command	Disabled command

Table. 4 Level 1 commands

Main scene, level 2

	Left controller	Right Controller
Menu	Official Steam menu	Official Steam menu
Trackpad	Disabled command	Modify the direction
System	Show list of 5 objects for 10 seconds	Exit menu
Index trigger	Active basket	Pick up objects
Grip	Disabled command	Disabled command

Table. 5 Level 2 commands

Main scene, level 3

	Left controller	Right Controller
Menu	Official Steam menu	Official Steam menu
Trackpad	Disabled command	Modify the direction
System	Show list of 7 objects for 7 seconds	Exit menu
Index trigger	Active basket	Pick up objects
Grip	Disabled command	Disabled command

Table. 6 Level 3 commands

11. Animations

The animation module quoted on the development engine section was crucial at the time of creating the distractions of the levels two and three. A more detailed explanation of the module will be explained below.

Unity has many features which make the program a good choice at the time of creating VR experiences, one of them is the animation module, this module allows inexpert users like us create animations in an easy way.

As we can see in the **figure 61**, one of the most important thinks when an animation is being created are the key frames, these key frames are the represent the position where the creator wants to move and animate the object, a humanoid in our case. Between one key frame and the next one is supposed to be a transition, Unity helps a lot by automatizing this transition.

Normally animating inert objects is quite easy, but in or case was a bit more difficult due to the need of animating a humanoid, because of the big number of articulations and parts is hard to make clean transitions and smooth movements. So it took us a long time to carry out good transitions and realistic movements to perform a good animation.

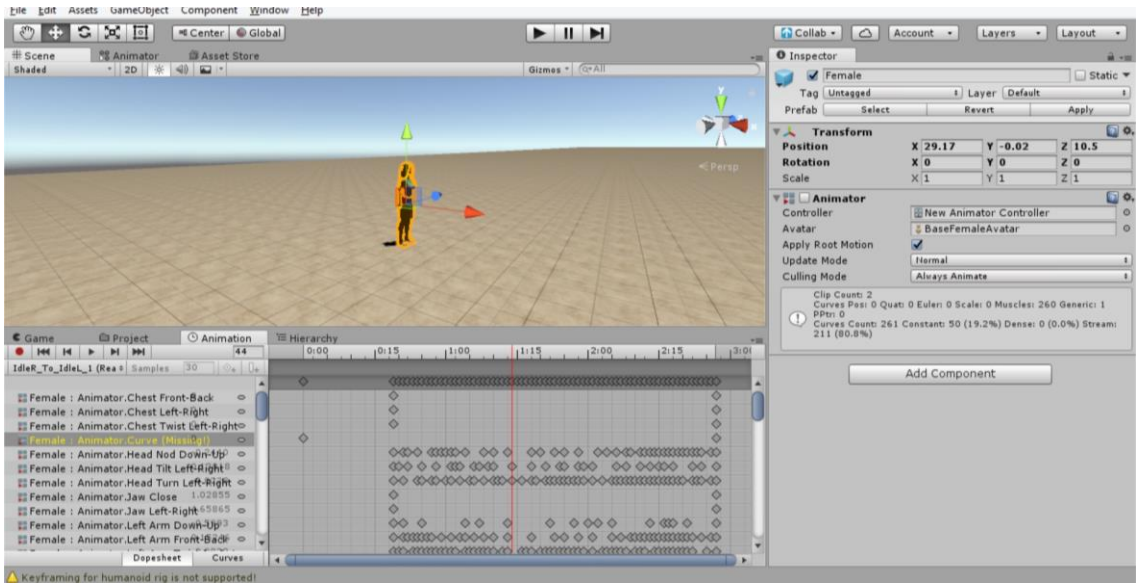


Fig. 63 Key frames example (Unity 3D screenshot)

The part explained before is just for creating single animations, after creating the necessary animations starts the use of another module of Unity, the Animator, shown next to this paragraph.

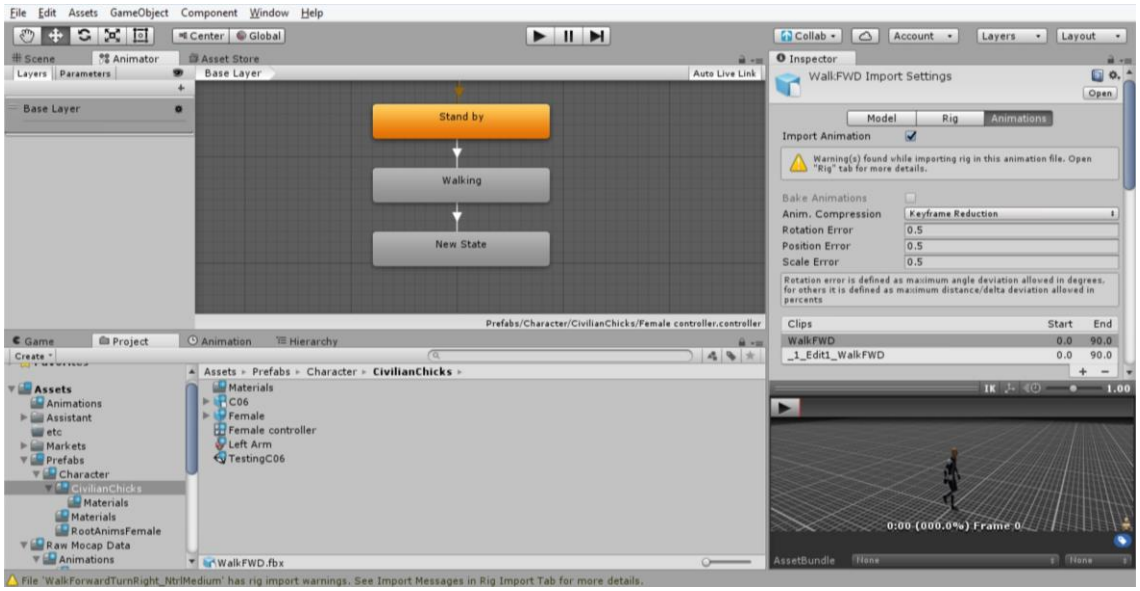


Fig. 64 Animator interface (Unity 3D screenshot)

This module was essential for putting together the animations into a bigger and more complex one.

Animator allows connecting the smaller animations like if it were a puzzle, these animations are connected by what are called transitions; Transitions are in charge of making smoother the change of animation and because of that making them the most possible realistic.

As we have commented previously animating inert objects is easier so we decided to also animate the doors of the tutorial for the closing and opening of them.

Thanks to the tools named the part of animating the models was not as complicated as could have been and saved a good amount of time and along the creating way we learned how to use them with a certain level of difficulty.

12. Database

One of the functional requirements of the app was to have a local database not accessible from external computers and for identifying the patients by an ID and not for their name in order to keep the privacy.

For this need we decide to appeal to the so known database management program MySQL, developed mostly by Oracle and Microsoft SQL Server. Thanks to the popularity of the program a big amount of tutorials exist and allowed us to solve all the doubts we had.

For creating the database we had to have very clear the variables we wanted to store, these variables are the next ones:

- ID
- Valid objects
- Non-valid objects
- Time
- Valid objects 5
- Non-valid objects 5
- Time 5
- Consults 5
- Valid objects 7
- Non-valid objects 7
- Time 7
- Consults 7

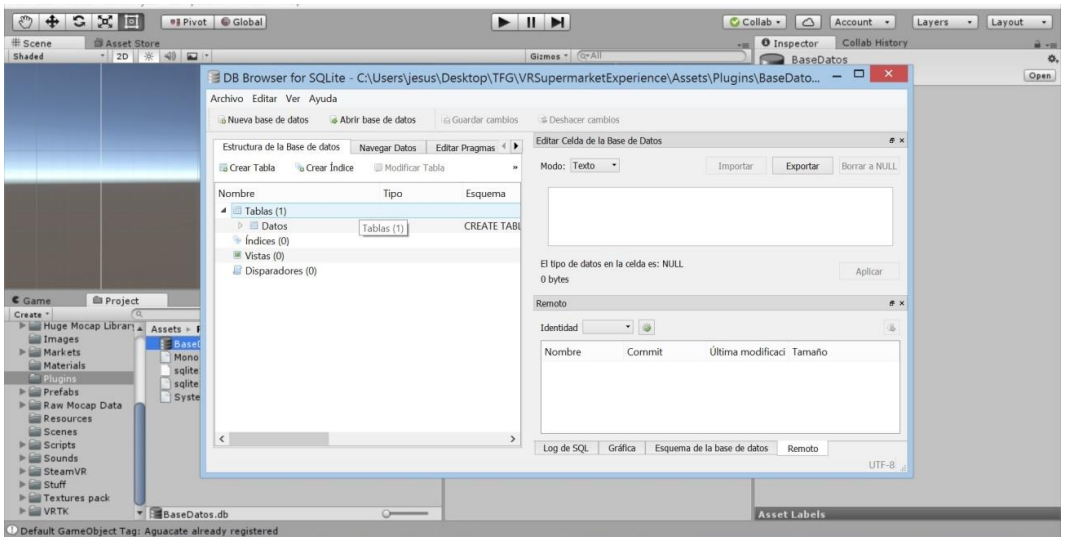


Fig. 65 DB browser with MySQL (Unity 3D screenshot)

Once the database was created the next step was to learn to connect with it, for this task we used C# code. A simple code to connect to our data through the path to follow could be like this one:

```
string conn = "URI=file:" + Application.dataPath + "/plugins/BaseDatos.db"; //Path to database.
        IDbConnection dbconn;
        dbconn = (IDbConnection) new SqliteConnection(conn);
        dbconn.Open();
```

Another big issue needed to be solved was when to write at de database, for avoiding unwanted overwriting with wrong data from failed trainings we decided to create a game manager that is storing the variables through the different scenes and once the training is finished with no incidents can be stored in the right way in the database with a code like the next one:

```
IDbCommand dbcmd = dbconn.CreateCommand();
        string sqlQuery = "UPDATE Datos SET ObjetosVal3='"+OV3+"',ObjetosNoVal3='"+ONV3+"',T3='"+T3+"' WHERE ID="+ID;
        dbcmd.CommandText = sqlQuery;
        IDataReader reader = dbcmd.ExecuteReader();
```

Another important part of connecting to the database is to close the connection and by this way only be connected when is needed.

```
reader.Close();
        reader = null;

        dbcmd.Dispose();
        dbcmd = null;

        dbconn.Close();
        dbconn = null
```

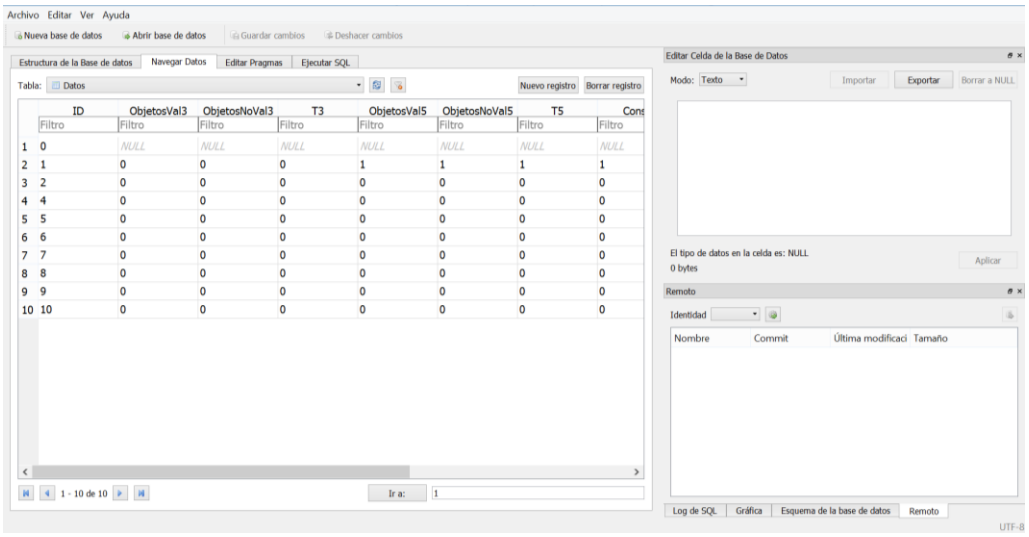


Fig. 66 MySQL data base (DB browser screenshot)

Conclusions

Once the project has finished and after the corresponding validations by the doctors and the company Vysion 360 it can be concluded that the initial goals have been achieved.

The project has consisted in the development of a Virtual Reality application aiming to improve the results obtained by the current techniques used in mild cognitive impairment treatments, thanks to a greater immersive capacity of this technology.

All the functional and non-functional requirements have been carried out.

An essential part of the project is the code. At the end of it, a solid and well-structured code has been developed, being the final version free of bugs.

Modelling a realistic environment was a fundamental part, because the supermarket had to be as real as possible to increase the user immersion. In order to do so, as it is explained in the "Design" section, all the possible technologies have been used, from 3D scanning to 3D repositories.

Another key requirement was making the VR app as user friendly as possible. To properly perform this, the tutorial and some explicative videos were created where the basic information and commands are explained.

Apart from the technical results, the project goals have been satisfactorily achieved, from learning basic code skills or enhancing the design repertory to use a cross-platform game engine like Unity.

It can be concluded that the development of a medical VR app for cognitive training has been fulfilled, allowing more immersive experience treatment and potentially improving the actual techniques.

However, the application has to be tested and it could be enhanced with some possible improvements, which would be implemented in the next months. Please find this exposed on the next section, "Future lines".

Future lines

Despite having finished the development of the app there is still more work to do, although we have made a lot of progress on the development of the VR application, this project will be continued by other students in the next months.

The next stage of the project will consist in testing the app and analysing the obtained results from a group of patients, at the same time these results will be compared with the obtained results from a non-computer assisted treatment. All of this with the purpose of evidencing the improvement on the old treatments. This new stage will be carried out by collaboration with the Alzheimer's unit from Hospital Clínic.

During our app development stage a couple of ideas with the aim of improving the app emerged, but because of the lack of time to implement them these ideas remained as these, just ideas that could be applied in the future.

One of the ideas is to add the possibility of paying the cashier the amount of money that will be calculated with the cost of the products. This includes adding the price of all the products of the supermarket and creating a code to sum the total price of the objects in the basket. Apart from these considerations, an interacting cashier should be added an interacting cashier on the checkout where the patient will have to pay the amount by deciding the number of notes and coins from a fictitious wallet. This improvement could take the app to a new level.

Another upgrade would be the addition of more shopping lists to the bank of random lists. Due to this, the time of training could be lengthened.

One possible enhancement on the immersive experience would be to add the new HTC VIVE module that converts in wireless the headset display, this could avoid possible tumbles.

Finally the addition of another animation as a distraction could open a new way of adding new levels where the number of interactions augments.

Budget

In this section, the direct and indirect costs of the application are going to be explained. The project began on 08/09/2017 and it ended on 11/01/2018, so we have worked 80 days, 640 hours.

Direct cost

Salary

In this project there have been two workers, who have been working 8 hours for 80 days.

Salary				
	Number of workers	Time (h)	Cost per-hour (€/h)	Cost (€)
Worker	2	640	30	38.400
TOTAL				38.400

Table. 7 Salary cost

Social Security

In Spain the social security cost in this type of projects is 23,60 % of the contribution base.

Social Security				
	Number of workers	Salary (€)	Percentage (%)	Cost (€)
Worker	2	19.200	23.60	9062.4
TOTAL				9062.4

Table. 8 Social security cost

Equipment

The hardware and software costs have been counted in this section.

Equipment			
	Quantity	Cost/unit (€/un.)	Cost (€)
Computer	2	1250	2500
HTC Vive	1	700	700
Unity 3D	2	Free	0
Paint	2	Free	0
Skanect	2	Free	0
TOTAL			3200

Table. 9 Equipment cost

Audio-visual costs

In the different game modes there are several parts where audio-visual contents are necessary to facility the compression of them, the costs of it are the following:

Audio-visual	
	Cost (€)
Video editions	250
Recording and audio editions	150
Game speaker	50
Tutorial edition	150
TOTAL	600

Table. 10 Audio-visual cost

Indirect cost

Installations and the associated expenses

The project has been carried out in an office, which has to be rented, and it has associated expenses.

Installations and the associated expenses			
	Months	Cost/month	Cost
Rent	5	300	1500
Internet	5	40	200
Water	5	10	50
Light	5	35	175
Gas	5	20	100
TOTAL			2025

Table. 11 Installations and associated expenses cost

Total cost.

Total cost	
	Cost (€)
Direct cost	
Salary	38.400
Social Security	9062.4
Equipment	3200
Audiovisual	600
TOTAL DIRECT COST	
Indirect cost	
Installations and the associated expenses	2025
TOTAL INDIRECT COST	2025
TOTAL	53287,4

Table.12 Total cost

Bibliography

[1] Man, D.W.K., Chung, J. and Lee, G., (2012). Evaluation of a virtual reality-based memory training programme for Hong Kong Chinese older adults with questionable dementia: a pilot study. *International Journal of Geriatric Psychiatry* 27(5):513-520

[2] Merians, A.S., Jack, D., Boian, R., Tremaine, M., Burdea, G.C. and Adamovich, S.V., (2002). Virtual reality- augmented rehabilitation for patients following stroke. Physical therapy 82(9):898-915.

[3] Munro, A., Breaux, R. and Patrey, J., (2002). Cognitive aspects of virtual environments design. Handbook of virtual environments: design, implementation, and applications, 415-434

[4] Optale, G., Urgesi, C., Busato, V., Marin, S., Piron, L., Priftis, K., Gamberini, L., Salvatore Capodieci, S. and Bordin, A., (2010). Controlling Memory Impairment in Elderly Adults Using Virtual Reality Memory Training: A Randomized Controlled Pilot Study. Neurorehabilitation and Neural Repair 24 (4) 348–357

[5] Petersen, R.C., Doody, R., Kurz, A., Mohs, R.C., Morris, J.C. and Rabins, P.V., (2001). Current concepts in mild cognitive impairment. Archives of neurology 58(12), 1985-1992.

[6] Petersen, R.C. and Morris, J.C., (2003). Mild Cognitive Impairment: Aging to Alzheimer's disease. New York: Oxford University Press

[7] Priore, C., Castelnuovo, G., and Liccione, D. (2003). Experience with V-STORE: considerations on presence in virtual environments for effective neuropsychological rehabilitation of executive functions. CyberPsychology & Behavior 6(3): 281-287

[8] Snowden, J. S. (1989). Semantic dementia: A form of circumscribe cerebral atrophy. Netherlands: IOS Press. Behavioural Neurology 2(3):167-182

[9] Talassi, E., Guerreschi, M., Feriani, M., Fedi, V., Bianchetti, A. and Trabucchi, M., (2007). Effectiveness of a cognitive rehabilitation program in mild dementia (MD) and mild cognitive impairment (MCI): a case control study. Archives of gerontology and geriatrics 44:391-399.

[10] Traykov, L., Raoux, N., Latour, F., Gallo, L., Hanon, O. and Baudic, S., (2004). Executive functions deficit in mild cognitive impairment. Cognitive and Behavioral Neurology 20(4):219-224

[11] Werner, P., (2004). Beliefs about memory problems and help seeking in elderly persons: a qualitative analysis. Clinical Gerontologist 27(4):19-30

[12]Winblad, B., Palmer, K., Kivipelto, M., Jelic, V., Fratiglioni, L. and Wahlund, L.O., (2004). Mild cognitive impairment - beyond controversies, towards a consensus: report of the International Working Group on Mild Cognitive Impairment. Journal of internal medicine 256(3):240-246

[13]Weiss, P.L., Naveh, Y. and Katz, N., (2003). Design and testing of a virtual environment to train stroke patients with unilateral spatial neglect to cross a street safely. Occupational therapy international 10(1):39-55

[14]Yeh, S. C., Chen, Y. C., Tsai, C. F. and Rizzo, A., (2012). An innovative virtual reality system for mild cognitive impairment: Diagnosis and evaluation. In Biomedical Engineering and Sciences (IECBES), IEEE EMBS Conference on (pp. 23-27)

[15]Zalla, T., Plassiart, C., Pillon, B., Grafman, J. and Sirigu, A., (2001). Action planning in a virtual context after prefrontal cortex damage. Neuropsychologia 39(8):759-770

[16]Zhang, Y., Han, B., Verhaeghen, P. and Nilsson, L.G., (2007). Executive functioning in older adults with mild cognitive impairment. Aging, Neuropsychology, and Cognition 14(6): 557-570

[17]Agrigoroaei, S. and Lachman, M.E., (2011). Cognitive Functioning in Midlife and Old Age: Combined Effects of Psychosocial and Behavioural Factors. The Journals of Gerontology 66B (1):30-40

[18]Aguinis, H., Henle, C.A. and Beaty, J.C., (2001). Virtual Reality Technology: A New Tool for Personnel Selection, International Journal of Selection and Assessment, 9, 70-83

[19]Anderson, S.W., Tranel, D., (2002). Neuropsychological consequences of dysfunction in human dorsolateral prefrontal cortex - Handbook of neuropsychology. Amsterdam: Elsevier

[20]Argüelles, S., Loewenstein, D.A., Eisdorfer, C. and Arguelles, T., (2001). Caregivers' judgments of the functional abilities of the Alzheimer's disease patient: impact of caregivers' depression and perceived burden. Journal of Gerontology: PSYCHOLOGICAL SCIENCES, 56(2):78-84

[21]Bird, M. and Kinsella, G., (1996). Long-term cued recall of tasks in senile dementia. Psychology and Aging, 11(1):45-56

[22]Burgess, P.W., Veitch, E., Lacy Costello, A. and Shallice, T., (2000). The cognitive and neuroanatomical correlates of multitasking. Neuropsychologia 38(6):848-863

[23]Fernández-Calvo, B., Rodríguez-Pérez, R., Contador, I., Rubio-Santorum, A. and Ramos, F., (2011). Eficacia del entrenamiento cognitivo basado en nuevas tecnologías en pacientes con demencia tipo Alzheimer. Psicothema. 23(1): 44-50

[24]Fillit, H.M., Simon, E.S., Doniger, G.M. and Cummings, J.L., (2008). Practicality of a computerized system for cognitive assessment in the elderly. Alzheimer's & Dementia 4(1):14-21

[25]Flynn, D., Van Schaik, P., Blackman, T., Femcott, C., Hobbs, B. and Calderon, C., (2004). Developing a virtual reality-based methodology for people with dementia: a feasibility study. CyberPsychology & Behavior 6(6):591-611

[26]Garcia, L., Kartolo, A., & Méthot-Curtis, E. (2012). A discussion of the use of virtual reality in dementia. INTECH Open Access Publisher

[27]Godefroy, O., (2003). Frontal syndrome and disorders of executive functions. Journal of Neurology 250(1):1-6

[28]Hill, N., Mowszowski, L., Naismith, S., Verity, L., Chadwick, V., Valenzuela, M. and Lampit, A., (2017). Computerized Cognitive Training in Older Adults with Mild Cognitive Impairment or Dementia: A Systematic Review and Meta-Analysis. American Journal of Psychiatry April 2017, Vol. 174, No. 4, pp. 329 – 340

[29]Johnson and Hyde, (1997). Transfer of learning between 2D and 3D sources during infancy: Informing theory and practice. Developmental Review 30(2):128-154

[30]Katz, N., Rand, D., Shahar, M., Kizony, R. and Weiss, P.L., (2005). The virtual mall: a functional virtual environment for stroke rehabilitation. Cyberpsychology & Behavior 8(4):369-370

[31]Castelnuovo, G., Lo Priore, Liccione, D., C. and Cioffi, G., (2003). Virtual reality based tools for the rehabilitation of cognitive and executive functions: the V-Store. PsychNology Journal 1(3):310-325

[32]Hill, N. T. M., Mowszowski, L., Naismith, S. L., Chadwick, V. L., Valenzuela, M., & Lampit, A. (2017). Computerized cognitive training in older adults with mild cognitive impairment or dementia: A systematic review and meta-analysis. American Journal of Psychiatry, 174(4), 329–340

[33]Fernández-Calvo, B., Rodríguez-Pérez, R., Contador, I., Rubio-Santorum, A., & Ramos, F. (2011). [Efficacy of cognitive training programs based on new software technologies in patients with Alzheimer-type dementia]. Psicothema, 23(1), 44–50. Retrieved from http://www.redi-bw.de/db/ebsco.php/search.ebscohost.com/login.aspx?direct=true&db=cmedm&AN=21266141&site=ehost-live

[34]Optale, G., Urgesi, C., Busato, V., Marin, S., Piron, L., Priftis, K., … Bordin, A. (2010). Controlling memory impairment in elderly adults using virtual reality memory training: A randomized controlled pilot study. Neurorehabilitation and Neural Repair, 24(4), 348–357. https://doi.org/10.1177/1545968309353328

[35]Man, D. W. K., Chung, J. C. C., & Lee, G. Y. Y. (2012). Evaluation of a virtual reality-based memory training programme for Hong Kong Chinese older adults with questionable dementia: A pilot study. International Journal of Geriatric Psychiatry, 27(5), 513–520. https://doi.org/10.1002/gps.2746

[36]Lampit, A., Hallock, H., & Valenzuela, M. (2014). Computerized Cognitive Training in Cognitively Healthy Older Adults: A Systematic Review and Meta-Analysis of Effect Modifiers. PLoS Medicine, 11(11). https://doi.org/10.1371/journal.pmed.1001756

[37]Blender. (2017). Es.wikipedia.org. Retrieved 19 May 2017, from https://es.wikipedia.org/wiki/Blender

[38]Fernández Rodríguez, A., & Valdaracete Peinado, J. (2012). Introducción a Blender (Licenciatura). Escuela Universitaria de Ingeniería Técnica de Telecomunicación

[39]Google +. HTC Vive: ¡ya las hemos probado!. HobbyConsolas. Retrieved 19 May 2017, from http://www.hobbyconsolas.com/reportajes/htc-vive-ya-hemos-probado-120498

[40]Losada Suárez, O. (2015). Motor de juego como plataforma para la aplicación de algoritmos de Inteligencia Artificial (Licenciatura). Universidad Autónoma de Madrid

[41]Manresa Yee, C., Abásolo, M., Más Sansó, R., & Vénere, M. (2011). Realidad virtual y realidad aumentada. Interfaces avanzadas. (1st ed., pp. 59-74). Buenos Aires, Argentina: Editorial de la Universidad Nacional de La Plata (Edulp). ISBN Nº 978-950-34-0765-3

[42]Marqués Rubio, R. Estudio Comparativo de Software de Diseño de Gráficos en Tres Dimensiones (Licenciatura). Universidad de Valladolid

[43]MODELADO 3D COMPARATIVOS BLENDER SKETCH 3DS MAX. (2017). prezi.com. Retrieved 19 May 2017, from https://prezi.com/lwr6kml1kyhl/modelado-3d-comparativos-blender-sketch-3ds-max

[44]OCULUS RIFT - Realidad-Virtual.net. Realidad-Virtual.net. Retrieved 19 May 2017, from http://www.realidad-virtual.net/oculus-rift

[45]Ypenaar, D. Wat is de Beste VR Bril van 2017 - Top 5 Beste Virtual Reality Brillen. DeTopvijf.nl. Retrieved 19 May 2017, from http://www.detopvijf.nl/beste-vr-bril

[46]Unity (motor de juego). (2017). Es.wikipedia.org. Retrieved 19 May 2017, from https://es.wikipedia.org/wiki/Unity_(motor_de_juego)

[47]Unity - Game Engine. (2017). Unity. Retrieved 19 May 2017, from https://unity3d.com

[48]Sensorama | IDIS. Proyectoidis.org. Retrieved 19 May 2017, from http://proyectoidis.org/sensorama/

Annex A. Program codes

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class StopGravity7 : MonoBehaviour {

    public Text ObjetosCesta;
    private string ObjetosPrevios;
    private string ObjetosPrevios2;
    public GameObject referencia;
    private int V1=0;
    private int V2= 0;
    private int V3= 0;
    private int V4= 0;
    private int V5= 0;
    private int V6= 0;
    private int V7= 0;
    private int pet2;
    public int Valid=0;
    public int NoVal=0;
    public GameObject time;
    public int timeint;

    // Use this for initialization
    void Start () {

        pet2 = referencia.GetComponent<ListaCinco> ().pet;

    }

    void OnTriggerEnter (Collider other) {

        if (other.tag != "Untagged") {

            other.GetComponent<Rigidbody> ();
            //other.attachedRigidbody.isKinematic = true;
            other.gameObject.transform.parent = transform;

            if (pet2 == 0) {

                if (other.tag == "Maíz") {
                    V1 = 1;
```

```
                    print (V1);
                }
                if (other.tag == "Ensalada") {
                    V2 = 1;
                    print (V2);
                }
                if (other.tag == "Aguacate") {
                    V3 = 1;
                    print (V3);
                }
                if (other.tag == "Tomate") {
                    V4 = 1;
                    print (V4);
                }
                if (other.tag == "Zanahoria") {
                    V5 = 1;
                    print (V5);
                }
                if (other.tag == "Salt") {
                    V6 = 1;
                    print (V5);
                }
                if (other.tag == "Aceite") {
                    V7 = 1;
                    print (V5);
                }
                if (other.tag !="Maíz" && other.tag !="Ensalada"
 && other.tag !="Aguacate" && other.tag !="Tomate" && other.tag
!="Zanahoria" && other.tag !="Salt" && other.tag !="Aceite")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }
            }

            if (pet2 == 1) {

                if (other.tag == "Helado") {
                    V1 = 1;
                    print (V1);
                }
                if (other.tag == "Bistec") {
                    V2 = 1;
                    print (V2);
                }
                if (other.tag == "Garbanzos") {
                    V3 = 1;
                    print (V3);
```

```
                }
                if (other.tag == "Salt") {
                    V4 = 1;
                    print (V4);
                }
                if (other.tag == "Aceite") {
                    V5 = 1;
                    print (V5);
                }
                if (other.tag == "Patatas de bolsa") {
                    V6 = 1;
                    print (V5);
                }
                if (other.tag == "Pan") {
                    V7 = 1;
                    print (V5);
                }
                if (other.tag !="Helado" && other.tag !="Bistec"
 && other.tag !="Garbanzos" && other.tag !="Salt" && other.tag !
="Aceite" && other.tag !="Patatas de bolsa" && other.tag !="Pan"
)
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }
            }
            if (pet2 == 2) {

                if (other.tag == "Pan bimbo") {
                    V1 = 1;
                    print (V1);
                }
                if (other.tag == "Mantequilla") {
                    V2 = 1;
                    print (V2);
                }
                if (other.tag == "Mermelada") {
                    V3 = 1;
                    print (V3);
                }
                if (other.tag == "Tomate") {
                    V4 = 1;
                    print (V4);
                }
                if (other.tag == "Leche") {
                    V5 = 1;
                    print (V5);
                }
```

```
                if (other.tag == "Aceite") {
                    V6 = 1;
                    print (V5);
                }
                if (other.tag == "Manzana") {
                    V7 = 1;
                    print (V5);
                }
                if (other.tag !="Pan bimbo" && other.tag !="Mant
equilla" && other.tag !="Mermelada" && other.tag !="Tomate" && o
ther.tag !="Leche" && other.tag !="Aceite" && other.tag !="Manza
na")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }
            }
            other.gameObject.SetActive (false);

            ObjetosCesta.text= string.Format("\n"+other.tag+Obje
tosPrevios2);
            ObjetosPrevios= ObjetosCesta.text;
            ObjetosPrevios2= ObjetosPrevios;
        }
    }

    // Update is called once per frame
    void Update () {
        Valid = V1 + V2 + V3 + V4 + V5 + V6 + V7;
        timeint= (int)time.GetComponent<TimeCounter>().time;
    }
}
```

```
using UnityEngine.UI;
using System.Collections;
using UnityEngine;
using System.Collections.Generic;

public class TimeCounter : MonoBehaviour {
    public Text Timer;
    public float time= 0.0f;
    // Use this for initialization

    void Start () {

    }

    // Update is called once per frame
    void Update () {
        time += Time.deltaTime;

        var minutes = (int)time / 60;
        var seconds = time % 60;
        var fraction = (time * 100) % 100;

        Timer.text = string.Format("{0:00} : {1:00} ", minutes,
seconds);

        //Timer.text = string.Format("{0:00} : {1:00} : {2:000}"
, minutes, seconds, fraction);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class StopGravity5 : MonoBehaviour {

    public Text ObjetosCesta;
    private string ObjetosPrevios;
    private string ObjetosPrevios2;
    public GameObject referencia;
    private int V1=0;
    private int V2= 0;
    private int V3= 0;
    private int V4= 0;
    private int V5= 0;
    private int pet2;
    public int Valid=0;
    public int NoVal=0;

    //public GameObject time;
    //public int timeint;
    // Use this for initialization
    void Start () {

        pet2 = referencia.GetComponent<ListaCinco> ().pet;

    }

    void OnTriggerEnter (Collider other) {

        if (other.tag != "Untagged") {

            other.GetComponent<Rigidbody> ();
            //other.attachedRigidbody.isKinematic = true;
            other.gameObject.transform.parent = transform;

            if (pet2 == 0) {

                if (other.tag == "Ensalada") {
                    V1 = 1;
                    print (V1);
                }
                if (other.tag == "Vinagre") {
                    V2 = 1;
                    print (V2);
```

```
                }
                if (other.tag == "Salt") {
                    V3 = 1;
                    print (V3);
                }
                if (other.tag == "Maíz") {
                    V4 = 1;
                    print (V4);
                }
                if (other.tag == "Aceite") {
                    V5 = 1;
                    print (V5);
                }
                if (other.tag !="Ensalada" && other.tag !="Vinag
re" && other.tag !="Salt" && other.tag !="Maíz" && other.tag !="
Aceite")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }
            }

            if (pet2 == 1) {

                if (other.tag == "Arroz") {
                    V1 = 1;
                    print (V1);
                }
                if (other.tag == "Pollo") {
                    V2 = 1;
                    print (V2);
                }
                if (other.tag == "Guisantes") {
                    V3 = 1;
                    print (V3);
                }
                if (other.tag == "Salt") {
                    V4 = 1;
                    print (V4);
                }
                if (other.tag == "Cebolla") {
                    V5 = 1;
                    print (V5);
                }
                if (other.tag !="Arroz" && other.tag !="Pollo" &
& other.tag !="Guisantes" && other.tag !="Salt" && other.tag !="
Cebolla")
                {
```

```
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }
            }

            if (pet2 == 2) {

                if (other.tag == "Ajo") {
                    V1 = 1;
                    print (V1);
                }
                if (other.tag == "Bistec") {
                    V2 = 1;
                    print (V2);
                }
                if (other.tag == "Aceite") {
                    V3 = 1;
                    print (V3);
                }
                if (other.tag == "Salt") {
                    V4 = 1;
                    print (V4);
                }
                if (other.tag == "Garbanzos") {
                    V5 = 1;
                    print (V5);
                }
                if (other.tag !="Ajo" && other.tag !="Bistec" &&
 other.tag !="Aceite" && other.tag !="Salt" && other.tag !="Garb
anzos")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }
            }
            other.gameObject.SetActive (false);

            ObjetosCesta.text= string.Format("\n"+other.tag+Obje
tosPrevios2);
            ObjetosPrevios= ObjetosCesta.text;
            ObjetosPrevios2= ObjetosPrevios;
        }
    }

    // Update is called once per frame
    void Update () {
        Valid = V1 + V2 + V3 + V4 + V5;
        //timeint= (int)time.GetComponent<TimeCounter>().time;
```

```
        }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class StopGravity : MonoBehaviour {

    public Text ObjetosCesta;
    private string ObjetosPrevios;
    private string ObjetosPrevios2;
    public GameObject referencia;
    private int V1=0;
    private int V2=0;
    private int V3=0;
    public int Valid=0;
    public int NoVal=0;
    private int pet2;
    public GameObject time;
    public int timeint;
    // Use this for initialization
    void Start () {

        pet2=referencia.GetComponent<ListaTres>().pet;

    }

    // Update is called once per frame
    void OnTriggerEnter (Collider other) {

        if (other.tag != "Untagged") {

            other.GetComponent<Rigidbody> ();
            //other.attachedRigidbody.isKinematic = true;
            other.gameObject.transform.parent = transform;

            if (pet2 == 0) {

                if (other.tag == "Garbanzos") {
                    V1 = 1;
                    print (V1 + ("V1"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Aceite") {
                    V2 = 1;
                    print (V2 + ("V2"));
                    print (Valid+("Valid"));
```

```
                }
                if (other.tag == "Salt") {
                    V3 = 1;
                    print (V3 + ("V3"));
                    print (Valid+("Valid"));
                }
                if (other.tag !="Garbanzos" && other.tag !="Acei
te" && other.tag !="Salt")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }

            }
            if (pet2 == 1) {

                if (other.tag == "Leche") {
                    V1 = 1;
                    print (V1 + ("V1"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Pan") {
                    V2 = 1;
                    print (V2 + ("V2"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Plátanos") {
                    V3 = 1;
                    print (V3 + ("V3"));
                    print (Valid+("Valid"));
                }
                if (other.tag !="Leche" && other.tag !="Pan" &&
other.tag !="Plátanos")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }

            }
            if (pet2 == 2) {

                if (other.tag == "Pollo") {
                    V1 = 1;
                    print (V1 + ("V1"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Alubias") {
                    V2 = 1;
```

```
                    print (V2 + ("V2"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Manzana") {
                    V3 = 1;
                    print (V3 + ("V3"));
                    print (Valid+("Valid"));
                }
                if (other.tag !="Pollo" && other.tag !="Alubias"
 && other.tag !="Manzana")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }

            }

            if (pet2 == 3) {

                if (other.tag == "Bistec") {
                    V1 = 1;
                    print (V1 + ("V1"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Salt") {
                    V2 = 1;
                    print (V2 + ("V2"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Aceite") {
                    V3 = 1;
                    print (V3 + ("V3"));
                    print (Valid+("Valid"));
                }
                if (other.tag !="Bistec" && other.tag !="Salt" &
& other.tag !="Aceite")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }

            }
            if (pet2 == 4) {

                if (other.tag == "Pan") {
                    V1 = 1;
                    print (V1 + ("V1"));
                    print (Valid+("Valid"));
```

```
                }
                if (other.tag == "Aceite") {
                    V2 = 1;
                    print (V2 + ("V2"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Tomate") {
                    V3 = 1;
                    print (V3 + ("V3"));
                    print (Valid+("Valid"));
                }
                if (other.tag !="Pan" && other.tag !="Aceite" &&
 other.tag !="Tomate")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }

            }
            if (pet2 == 5) {

                if (other.tag == "Endivia") {
                    V1 = 1;
                    print (V1 + ("V1"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Vinagre") {
                    V2 = 1;
                    print (V2 + ("V2"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Maíz") {
                    V3 = 1;
                    print (V3 + ("V3"));
                    print (Valid+("Valid"));
                }
                if (other.tag !="Endivia" && other.tag !="Vinagr
e" && other.tag !="Maíz")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }

            }
            if (pet2 == 6) {

                if (other.tag == "Pan") {
                    V1 = 1;
```

```
                    print (V1 + ("V1"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Queso") {
                    V2 = 1;
                    print (V2 + ("V2"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Chips") {
                    V3 = 1;
                    print (V3 + ("V3"));
                    print (Valid+("Valid"));
                }
                if (other.tag !="Pan" && other.tag !="Queso" &&
other.tag !="Chips")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }

            }
            if (pet2 == 7) {

                if (other.tag == "Arroz") {
                    V1 = 1;
                    print (V1 + ("V1"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Pollo") {
                    V2 = 1;
                    print (V2 + ("V2"));
                    print (Valid+("Valid"));
                }
                if (other.tag == "Guisantes") {
                    V3 = 1;
                    print (V3 + ("V3"));
                    print (Valid+("Valid"));
                }
                if (other.tag !="Arroz" && other.tag !="Pollo" &
& other.tag !="Guisantes")
                {
                    NoVal = NoVal + 1;
                    print (NoVal+("NoVal"));
                }
            }
            other.gameObject.SetActive (false);

            ObjetosCesta.text= string.Format("\n"+other.tag+Obje
```

```
tosPrevios2);
            ObjetosPrevios= ObjetosCesta.text;
            ObjetosPrevios2= ObjetosPrevios;
        }
    }

    void Update (){
        Valid = V1 + V2 + V3;
        timeint= (int)time.GetComponent<TimeCounter>().time;

        }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class StartWalking : MonoBehaviour {

    public GameObject Mujer;
    public GameObject LeftController;
    public GameObject RightController;
    public int TiempoEspera;//Normalmente el tiempo será de 10 s
eg pero por si se varía la animación
    Animator animation1;
    AudioSource AudioSourceMujer;
    public int TiempoSaludar;
    private int contadorsaludo;

    // Use this for initialization
    void Start () {

        animation1 = Mujer.GetComponent<Animator> ();
        AudioSourceMujer= Mujer.GetComponent<AudioSource>();
    }

    // Update is called once per frame
    void OnTriggerEnter (Collider collider) {

        animation1.enabled = true;
        LeftController.SetActive (false);
        RightController.SetActive (false);
        StartCoroutine (TiempoEsperar ());
        StartCoroutine (TiempoSaludo ());
    }
    IEnumerator TiempoEsperar()
    {
        yield return new WaitForSeconds(TiempoEspera);
        LeftController.SetActive (true);
        RightController.SetActive (true);
        Mujer.SetActive (false);
        gameObject.SetActive (false);
    }
    IEnumerator TiempoSaludo()
    {
        yield return new WaitForSeconds(TiempoSaludar);
        contadorsaludo = contadorsaludo + 1;
        if (contadorsaludo == 1) {
            AudioSourceMujer.Play ();
```

```
        }
    }
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class StartTalking : MonoBehaviour {

    public GameObject Mujer;
    public GameObject LeftController;
    public GameObject RightController;
    AudioSource AudioSourceMujer;
    public AudioClip audiodistrac;
    int contador=0;
    // Use this for initialization
    void Start () {
        AudioSourceMujer = Mujer.GetComponent<AudioSource> ();
    }

    void OnTriggerEnter (Collider collider) {

        LeftController.SetActive (false);
        RightController.SetActive (false);
        StartCoroutine (TiempoEsperar ());
        contador = contador + 1;

        if (contador == 1) {
            AudioSourceMujer.Play ();
        }
    }
    IEnumerator TiempoEsperar()
    {
        yield return new WaitForSeconds (audiodistrac.length);
        LeftController.SetActive (true);
        RightController.SetActive (true);
        gameObject.SetActive (false);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Mono.Data.Sqlite;
using System.Data;
using System;
using UnityEngine.UI;

public class SobreEscribirDatos7 : MonoBehaviour {

    private GameObject GameControlDatos;
    private int ID;
    private int OV7 ;
    private int ONV7 ;
    private int T7 ;

    // Use this for initialization
    void Start () {

        GameControlDatos = GameObject.Find ("GameControl");
        ID= GameControlDatos.GetComponent<GameControl>().ID;
        OV7 = GameControlDatos.GetComponent<GameControl> ().OV7;
        ONV7 = GameControlDatos.GetComponent<GameControl> ().ONV
7;
        T7 = GameControlDatos.GetComponent<GameControl> ().T7;
    }

    public void Escena3(){

        string conn = "URI=file:" + Application.dataPath + "/plu
gins/BaseDatos.db"; //Path to database.
        IDbConnection dbconn;
        dbconn = (IDbConnection) new SqliteConnection(conn);
        dbconn.Open(); //Open connection to the database.

        IDbCommand dbcmd = dbconn.CreateCommand();
        string sqlQuery = "UPDATE Datos SET ObjetosVal7='"+OV7+"
',ObjetosNoVal7='"+ONV7+"',T7='"+T7+"' WHERE ID="+ID;
        dbcmd.CommandText = sqlQuery;
        IDataReader reader = dbcmd.ExecuteReader();


        reader.Close();
        reader = null;

        dbcmd.Dispose();
```

```
            dbcmd = null;

            dbconn.Close();
            dbconn = null;


        }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Mono.Data.Sqlite;
using System.Data;
using System;
using UnityEngine.UI;

public class SobreEscribirDatos : MonoBehaviour {

    private GameObject GameControlDatos;
    private int ID;
    private int OV3 ;
    private int ONV3 ;
    private int T3 ;

    // Use this for initialization
    void Start () {

        GameControlDatos = GameObject.Find ("GameControl");
        ID= GameControlDatos.GetComponent<GameControl>().ID;
        OV3 = GameControlDatos.GetComponent<GameControl> ().OV3;
        ONV3 = GameControlDatos.GetComponent<GameControl> ().ONV
3;
        T3 = GameControlDatos.GetComponent<GameControl> ().T3;
    }

    public void Escena1(){

        string conn = "URI=file:" + Application.dataPath + "/plu
gins/BaseDatos.db"; //Path to database.
        IDbConnection dbconn;
        dbconn = (IDbConnection) new SqliteConnection(conn);
        dbconn.Open(); //Open connection to the database.

        IDbCommand dbcmd = dbconn.CreateCommand();
        string sqlQuery = "UPDATE Datos SET ObjetosVal3='"+OV3+"
',ObjetosNoVal3='"+ONV3+"',T3='"+T3+"' WHERE ID="+ID;
        dbcmd.CommandText = sqlQuery;
        IDataReader reader = dbcmd.ExecuteReader();


        reader.Close();
        reader = null;

        dbcmd.Dispose();
```

```
            dbcmd = null;

            dbconn.Close();
            dbconn = null;


        }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;


public class ReproduceSound : MonoBehaviour {

    AudioSource audioSource;
    public AudioClip sonido;
    // Use this for initialization
    void Start () {

        audioSource = GetComponent<AudioSource> ();
        GetComponent<AudioSource> ().clip = sonido;
        audioSource.Play();
    }


}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class RaycastTuto : MonoBehaviour {
    //public GameObject door1;
    //public GameObject door2;
    public GameObject totem1;
    public GameObject totem2;
    public GameObject totem3;
    public GameObject LeftController;
    public GameObject RightController;
    private float distanceToSee= 0.0f;


    AudioSource audiosourceinicio;
    AudioSource audiosourceizquierda;
    AudioSource audiosourcederecha;
    AudioSource audiosourcegiro;
    public AudioClip audioinicio;
    public AudioClip audioizquierda;
    public AudioClip audioderecha;
    public AudioClip audiogiro;

    private int counter;
    private int counter2;
    private int counter3;
    private bool tot1= false;
    private bool tot2= false;
    private bool tot3= false;
    //Animator  animation1;
    //Animator animation2;

    // Use this for initialization
    void Start () {
        //animation1 = door1.GetComponent<Animator>();
        //animation2 = door2.GetComponent<Animator>();
        LeftController.SetActive(false);
        RightController.SetActive(false);
        audiosourceinicio = GetComponent<AudioSource> ();
        audiosourceinicio.clip = audioinicio;
        audiosourceinicio.Play ();
        StartCoroutine (tiempo ());
        audiosourceizquierda= totem1.GetComponent<AudioSource>()
;
        audiosourceizquierda.clip = audioizquierda;
```

```
        audiosourcederecha = totem2.GetComponent<AudioSource> ()
;
        audiosourcederecha.clip = audioderecha;
        audiosourcegiro = totem3.GetComponent<AudioSource> ();
        audiosourcegiro.clip = audiogiro;

    }

    IEnumerator tiempo()
    {
        yield return new WaitForSeconds (audioinicio.length);
        distanceToSee = 200.0f;
    }

    IEnumerator tiempototem1()
   {
        yield return new WaitForSeconds (audioizquierda.length);
        totem2.SetActive(true);
    }

    IEnumerator tiempototem2()
    {
        yield return new WaitForSeconds(audioderecha.length);
        totem3.SetActive(true);
    }

    IEnumerator tiempototem3()
    {
        yield return new WaitForSeconds(audiogiro.length);
        LeftController.SetActive(true);
        RightController.SetActive(true);
    }
    // Update is called once per frame
    void Update () {
        RaycastHit hit;


        if (Physics.Raycast (this.transform.position, this.trans
form.forward, out hit, distanceToSee)) {
            if (hit.collider.gameObject == totem1) {
                totem1.GetComponent<Renderer> ().material.color
= Color.green;
                tot1 = true;
                counter = counter + 1;
                if (counter == 1) {
                    StartCoroutine(tiempototem1());
                    audiosourceizquierda.Play ();
                }
```

```
            }
            if (hit.collider.gameObject == totem2) {

                totem2.GetComponent<Renderer> ().material.color
= Color.green;
                tot2 = true;
                counter2 = counter2 + 1;
                if (counter2 == 1) {
                    StartCoroutine(tiempototem2());
                    audiosourcederecha.Play ();
                }
            }
                if (hit.collider.gameObject == totem3) {

                    totem3.GetComponent<Renderer> ().material.co
lor = Color.green;
                    tot3 = true;
                    counter3 = counter3 + 1;
                    if (counter3 == 1) {
                    StartCoroutine(tiempototem3());
                    audiosourcegiro.Play ();
                    }
                }

                    }
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class QuitScene : MonoBehaviour {

    public void Quit()
    {
#if UNITY_EDITOR
        UnityEditor.EditorApplication.isPlaying = false;
#else
        Application.Quit ();
#endif
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PreviousScene : MonoBehaviour {

    private static string lastLevel;

    public static void setLastLevel(string level)
    {
        lastLevel = level;
    }

    public static string getLastLevel()
    {
        return lastLevel;
    }

    public static void changeToPreviousLvl()
    {
        Application.LoadLevel(lastLevel);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class LoadSceneOnClick : MonoBehaviour {

    public void LoadByIndex(int sceneIndex)
    {
        SceneManager.LoadScene(sceneIndex);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ListaTres : MonoBehaviour {

    public Text Lista;
    private string[] listas = {
        "Garbanzos\nAceite\nSal",
        "Leche\nPan\nPlátano",
        "Pollo\nAlubias\nManzana",
       "Bistec\nSal\nAceite",
       "Pan\nAceite\nTomate",
       "Endivia\nVinagre\nMaíz",
       "Pan\nQueso\nChips",
       "Arroz\nPollo\nGuisantes"
    };
    public int pet;

    // Use this for initialization
    void Start () {

        pet =Random.Range(0,listas.Length);
        print (listas[pet]);

        Lista.text = listas [pet];

    }

    // Update is called once per frame
    void Update () {

    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ListaSiete : MonoBehaviour {

    public Text Lista;
    private string[] listas = {
        "Maíz\nEnsalada\nAguacate\nTomate\nZanahoria\nSal\nAceit
e",
        "Helado\nBistec\nGarbanzos\nSal\nAceite\nPatatas de bols
a\nPan",
        "Pan bimbo\nMantequilla\nMermelada\nTomates\nLeche\nAcei
te\nMazanas"
    };
    public int pet;

    // Use this for initialization
    void Start () {

        pet =Random.Range(0,listas.Length);
        print (listas[pet]);

        Lista.text = listas [pet];

    }

    // Update is called once per frame
    void Update () {

    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ListaCinco : MonoBehaviour {

    public Text Lista;
    private string[] listas = {
        "Ensalada\nVinagre\nSal\nMaíz\nAceite",
        "Arroz\nPollo\nGuisantes\nSal\nCebolla",
        "Ajo\nBistec\nAceite\nSal\nGarbanzos"
    };

    public int pet;

    // Use this for initialization
    void Start () {

        pet =Random.Range(0,listas.Length);
        print (listas[pet]);

        Lista.text = listas [pet];

    }

    // Update is called once per frame
    void Update () {

    }
}
```

```
namespace VRTK
{
    using System.Collections;
    using System.Collections.Generic;
    using UnityEngine;


    public class IrMenuRepeticion : MonoBehaviour
    {
        public GameObject MenuNota;
        public GameObject MenuSalir;
        public GameObject TouchpadLeft;
        public GameObject TouchpadRight;
        public GameObject LeftControllerMenu;
        public GameObject RightControllerMenu;


        // Use this for initialization
        void Start()
        {

        }
        void OnTriggerEnter(Collider collider)
        {
            MenuNota.SetActive(false);
            MenuSalir.SetActive(true);
            TouchpadLeft.SetActive(false);
            TouchpadRight.SetActive(false);
            LeftControllerMenu.GetComponent<VRTK_UIPointer>().en
abled = true;
            LeftControllerMenu.GetComponent<VRTK_Pointer>().enab
led = true;
            LeftControllerMenu.GetComponent<VRTK_StraightPointer
Renderer>().enabled = true;
            RightControllerMenu.GetComponent<VRTK_UIPointer>().e
nabled = true;
            RightControllerMenu.GetComponent<VRTK_Pointer>().ena
bled = true;
            RightControllerMenu.GetComponent<VRTK_StraightPointe
rRenderer>().enabled = true;
        }


    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class InicioSaludo : MonoBehaviour {

    public GameObject Mujer;
    AudioSource AudioSourceMujer;
    public int contador = 0;
    //public AudioClip AudioSaludo;
    // Use this for initialization
    void Start () {
        AudioSourceMujer= Mujer.GetComponent<AudioSource>();
    //    AudioSourceMujer.clip = AudioSaludo;
    }

    // Update is called once per frame
    void OnTriggerEnter (Collider other) {

        contador = contador + 1;

        if (contador == 1) {
            AudioSourceMujer.Play ();
            Debug.Log ("hola man");
        }
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class GoToFinalTuto : MonoBehaviour {

    public AudioClip finalsound;

    // Use this for initialization
    void Start () {

        StartCoroutine (FinalTuto ());

    }
    IEnumerator FinalTuto()
    {
        yield return new WaitForSeconds (finalsound.length);
        SceneManager.LoadScene ("Final Tutorial");
    }

}
```

```
    using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class GlobalControl : MonoBehaviour {


    public float time;
    public  static GlobalControl Instance;

    void Awake ()
    {
        if (Instance == null) {
            DontDestroyOnLoad (gameObject);
            Instance = this;
        } else if (Instance != this) {
            Destroy (gameObject);
        }
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using Mono.Data.Sqlite;
using System.Data;
using System;
using UnityEngine.UI;

public class GameControl : MonoBehaviour {

    public static GameControl control;
    //public GameObject BaseDatos;
    private GameObject dentrocesta;
    private GameObject LeftController;
    private GameObject tiempo;
    public int ID;
    public int OV3 ;
    public int ONV3 ;
    public int T3 ;
    public int OV5 ;
    public int ONV5 ;
    public int T5 ;
    public int C5 ;
    public int OV7 ;
    public int ONV7 ;
    public int T7 ;
    public int C7 ;


    void Awake () {
        if (control == null) {
            DontDestroyOnLoad (gameObject);
            control = this;
        }
        else if(control !=this)
        {
            Destroy(gameObject);
        }

    }



    public void GuardarDatos (){
```

```
        //ID = BaseDatos.GetComponent<db>().ID;
        //OV3 = BaseDatos.GetComponent<db> ().OV3;
        //ONV3 = BaseDatos.GetComponent<db> ().ONV3;
        //T3 = BaseDatos.GetComponent<db> ().T3;
        Debug.Log ("ID="+ ID + "OV3="+ OV3 + "ONV3="+ ONV3 + "T3
="+T3);

    }

    public void Escena1(){

        string conn = "URI=file:" + Application.dataPath + "/plu
gins/BaseDatos.db"; //Path to database.
        IDbConnection dbconn;
        dbconn = (IDbConnection) new SqliteConnection(conn);
        dbconn.Open(); //Open connection to the database.

        IDbCommand dbcmd = dbconn.CreateCommand();
        string sqlQuery = "UPDATE Datos SET ObjetosVal3='"+OV3+"
' WHERE ID="+ID;
        dbcmd.CommandText = sqlQuery;
        IDataReader reader = dbcmd.ExecuteReader();


        reader.Close();
        reader = null;

        dbcmd.Dispose();
        dbcmd = null;

        dbconn.Close();
        dbconn = null;

        //dentrocesta = GameObject.Find ("DentroCesta");
        //OV3 = dentrocesta.GetComponent<StopGravity> ().Valid;
        //ONV3 = dentrocesta.GetComponent<StopGravity> ().NoVal;
        //T3 = dentrocesta.GetComponent<StopGravity> ().timeint;
        //Debug.Log ("ID="+ ID + "OV3="+ OV3 + "ONV3="+ ONV3 + "
T3="+T3);
    }

    void Update(){


        if (SceneManager.GetActiveScene()==
            SceneManager.GetSceneByName("Main scene")) {
```

```
            dentrocesta = GameObject.Find ("DentroCesta");
            OV3 = dentrocesta.GetComponent<StopGravity> ().Valid
;
            ONV3 = dentrocesta.GetComponent<StopGravity> ().NoVa
l;
            T3 = dentrocesta.GetComponent<StopGravity> ().timein
t;

        }
        if (SceneManager.GetActiveScene()==
            SceneManager.GetSceneByName("Main sceneL2")) {


            tiempo = GameObject.Find ("Script");
            T5 = tiempo.GetComponent<TimerRecogida> ().timeint;
            LeftController = GameObject.Find ("LeftController");
            C5 = LeftController.GetComponent<ContadorListaConsul
ta> ().ContadorConsulta;
            dentrocesta = GameObject.Find ("DentroCesta");
            OV5 = dentrocesta.GetComponent<StopGravity5> ().Vali
d;
            ONV5 = dentrocesta.GetComponent<StopGravity5> ().NoV
al;



        }
        if (SceneManager.GetActiveScene()==
            SceneManager.GetSceneByName("Main sceneL3")) {

            tiempo = GameObject.Find ("Script");
            T7 = tiempo.GetComponent<TimerRecogida> ().timeint;
            LeftController = GameObject.Find ("LeftController");
            C7 = LeftController.GetComponent<ContadorListaConsul
ta> ().ContadorConsulta;
            dentrocesta = GameObject.Find ("DentroCesta");
            OV7 = dentrocesta.GetComponent<StopGravity7> ().Vali
d;
            ONV7 = dentrocesta.GetComponent<StopGravity7> ().NoV
al;

        }



    }
```

```
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DesactivarNota : MonoBehaviour {


    // Use this for initialization
    void Start () {
        StartCoroutine (DesactivarNotas ());
    }

    IEnumerator DesactivarNotas()
    {
        yield return new WaitForSeconds (15);
        gameObject.SetActive (false);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Mono.Data.Sqlite;
using System.Data;
using System;
using UnityEngine.UI;

public class db : MonoBehaviour {
    public GameObject GuardarDatos;
    public GameObject Input;

    private int ID ;
    private int OV3 ;
    private int ONV3 ;
    private int T3 ;
    private int OV5 ;
    private int ONV5 ;
    private int T5 ;
    private int C5 ;
    private int OV7 ;
    private int ONV7 ;
    private int T7 ;
    private int C7 ;


    void Start () {
        //sqlite_BaseDatos ();
    }


    public void sqlite_BaseDatos ()

    {
        string conn = "URI=file:" + Application.dataPath + "/plu
gins/BaseDatos.db"; //Path to database.
        IDbConnection dbconn;
        dbconn = (IDbConnection) new SqliteConnection(conn);
        dbconn.Open(); //Open connection to the database.

        IDbCommand dbcmd = dbconn.CreateCommand();
        string sqlQuery = "SELECT * FROM Datos WHERE ID= "+Input
.GetComponent<Text>().text;
        dbcmd.CommandText = sqlQuery;
        IDataReader reader = dbcmd.ExecuteReader();
```

```
        while (reader.Read())
        {
            int ID = reader.GetInt32(0);
            int OV3 = reader.GetInt32(1);
            int ONV3 = reader.GetInt32(2);
            int T3 = reader.GetInt32(3);
            int OV5 = reader.GetInt32(4);
            int ONV5 = reader.GetInt32(5);
            int T5 = reader.GetInt32(6);
            int C5 = reader.GetInt32(7);
            int OV7 = reader.GetInt32(8);
            int ONV7 = reader.GetInt32(9);
            int T7 = reader.GetInt32(10);
            int C7 = reader.GetInt32(11);

            Debug.Log( "ID= "+ID+"  Objetos válidos ="+OV3+" Obj
etos no válidos ="+ ONV3+"Tiempo ="+
                T3+"Objetos válidos 5="+OV5+"+ Objetos no válido
s5="+ONV5+" Tiempo="+T5+"Consutas 5="+C5+
                "Objetos válidos 7="+OV7+"Objetos no válidos 7="
+ONV7+"Tiempo 7="+T7+"Consultas 7="+C7);

            GuardarDatos.GetComponent<GameControl> ().ID = ID;
            //GuardarDatos.GetComponent<GameControl> ().OV3 = OV
3;
            GuardarDatos.GetComponent<GameControl> ().ONV3 = ONV
3;
            GuardarDatos.GetComponent<GameControl> ().T3 = T3;
            GuardarDatos.GetComponent<GameControl> ().OV5 = OV5;
            GuardarDatos.GetComponent<GameControl> ().ONV5 = ONV
5;
            GuardarDatos.GetComponent<GameControl> ().T5 = T5;
            GuardarDatos.GetComponent<GameControl> ().C5 = C5;
            GuardarDatos.GetComponent<GameControl> ().OV7 = OV7;
            GuardarDatos.GetComponent<GameControl> ().ONV7 = ONV
7;
            GuardarDatos.GetComponent<GameControl> ().T7 = T7;
            GuardarDatos.GetComponent<GameControl> ().C7 = C7;


        }
        reader.Close();
        reader = null;

        dbcmd.Dispose();
        dbcmd = null;

        dbconn.Close();
```

```
        dbconn = null;

          }
}
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ContadorListaConsulta : MonoBehaviour {

    public GameObject LeftController;
    public GameObject RightController;
    public GameObject Nota;
    public GameObject ListaCesta;
    public GameObject NombreProductosCesta;
    public int TiempoVista;

    public int ContadorConsulta=0;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    public void Contador () {
        LeftController.SetActive (false);
        RightController.SetActive (false);
        Nota.SetActive (true);
        ContadorConsulta = ContadorConsulta + 1;
        ListaCesta.SetActive (false);
        NombreProductosCesta.SetActive(false);
        StartCoroutine(TiempoMandos());
    }
    IEnumerator TiempoMandos()
    {
        yield return new WaitForSeconds(TiempoVista);
        LeftController.SetActive (true);
        RightController.SetActive (true);
        Nota.SetActive (false);
        ListaCesta.SetActive (true);
        NombreProductosCesta.SetActive(true);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AudioInicioMainScene23 : MonoBehaviour {
    public GameObject LeftController;
    public GameObject RightController;
    public GameObject Nota;
    public GameObject Timer;
    public GameObject ProductosCesta;
    public GameObject Objetos;

    public int TiempoVerNota;

    AudioSource audiosourceinicio;

    public AudioClip AudioInicio;


    // Use this for initialization
    void Start () {
        audiosourceinicio = GetComponent<AudioSource>();
        audiosourceinicio.clip = AudioInicio;
        audiosourceinicio.Play();
        LeftController.SetActive(false);
        RightController.SetActive(false);
        StartCoroutine(tiempoaudioinicial());

    }
    IEnumerator tiempoaudioinicial()
    {
        yield return new WaitForSeconds(AudioInicio.length);
        Nota.SetActive(true);
        StartCoroutine(tiempovernota());

    }
    IEnumerator tiempovernota()
    {
        yield return new WaitForSeconds(TiempoVerNota);
        Nota.SetActive(false);
        LeftController.SetActive(true);
        RightController.SetActive(true);
        Timer.SetActive(true);
        ProductosCesta.SetActive(true);
        Objetos.SetActive(true);
```

```
    }

}
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AudioInicioMainScene : MonoBehaviour {
    public GameObject LeftController;
    public GameObject RightController;
    public GameObject Nota;
    public GameObject Timer;
    public GameObject ProductosCesta;
    public GameObject Objetos;


    AudioSource audiosourceinicio;

    public AudioClip AudioInicio;


    // Use this for initialization
    void Start()
    {
        audiosourceinicio = GetComponent<AudioSource>();
        audiosourceinicio.clip = AudioInicio;
        audiosourceinicio.Play();
        LeftController.SetActive(false);
        RightController.SetActive(false);
        StartCoroutine(tiempoaudioinicial());

    }
    IEnumerator tiempoaudioinicial()
    {
        yield return new WaitForSeconds(AudioInicio.length);
        Nota.SetActive(true);
        LeftController.SetActive(true);
        RightController.SetActive(true);
        Timer.SetActive(true);
        ProductosCesta.SetActive(true);
        Objetos.SetActive(true);

    }
}
```

```
    using System.Collections;
using System.Collections.Generic;
using UnityEngine;


public class WhenCollision2 : MonoBehaviour {

    AudioSource audioSource;
    public AudioClip impact;
    // Use this for initialization
    void Start () {

        audioSource = GetComponent<AudioSource> ();
        GetComponent<AudioSource> ().clip = impact;

    }

    // Update is called once per frame
    void Update () {

    }

    IEnumerator OnTriggerEnter (Collider collider)
    {
        audioSource.Play ();
        yield return new WaitForSeconds (impact.length);

        gameObject.SetActive (false);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class TimerRecogida : MonoBehaviour {



        public GameObject time;
        public int timeint;


    // Update is called once per frame
    void Update () {

        timeint = (int)time.GetComponent<TimeCounter> ().time;

    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CloseFinalDoors : MonoBehaviour {
    public GameObject door1;
    public GameObject door2;
    public GameObject snap1;
    public GameObject controllerLeft;
    public GameObject controllerRight;
    Animator animation1;
    Animator animation2;
    AudioSource audiosourcepuerta;
    AudioSource audiosourceexplicacion;
    public AudioClip puerta;
    public AudioClip explicacion;
    private int contador;

    // Use this for initialization
    void Start () {
        animation1 = door1.GetComponent<Animator>();
        animation2 = door2.GetComponent<Animator>();
        audiosourcepuerta = door1.GetComponent<AudioSource> ();
        audiosourcepuerta.clip = puerta;
        audiosourceexplicacion = snap1.GetComponent<AudioSource>
 ();
        audiosourceexplicacion.clip = explicacion;
    }


    void OnTriggerEnter (Collider collider)
    {
        animation1.enabled= true;
        animation2.enabled= true;
        contador = contador + 1;
        if (contador == 1) {

            audiosourcepuerta.Play();
            audiosourceexplicacion.Play ();
            controllerLeft.SetActive (false);
            controllerRight.SetActive (false);
            StartCoroutine (tiempo ());
        }

    }
    IEnumerator tiempo()
```

```
    {
        yield return new WaitForSeconds (explicacion.length);
        controllerLeft.SetActive (true);
        controllerRight.SetActive (true);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CloseInitialDoors : MonoBehaviour {
    public GameObject door1;
    public GameObject door2;
    public GameObject arrow1;
    Animator animation1;
    Animator animation2;
    private int contador;
    AudioSource audiosourcepuerta;
    AudioSource audiosourcedireccion;
    public AudioClip puerta;
    public AudioClip direccion;
    // Use this for initialization
    void Start () {
        animation1 = door1.GetComponent<Animator>();
        animation2 = door2.GetComponent<Animator>();
        audiosourcepuerta = door1.GetComponent<AudioSource> ();
        audiosourcedireccion = arrow1.GetComponent<AudioSource>
();
        audiosourcepuerta.clip = puerta;
        audiosourcedireccion.clip = direccion;
    }


    void OnTriggerEnter (Collider collider)
    {
        animation1.SetBool ("close",true);
        animation2.SetBool ("close",true);
        contador = contador+1;
        if (contador == 1) {

            audiosourcepuerta.Play ();
            audiosourcedireccion.Play ();
        }

    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class OpenInitialDoors : MonoBehaviour {
    public GameObject door1;
    public GameObject door2;
    Animator animation1;
    Animator animation2;
    AudioSource audiosourcepuerta;
    public AudioClip puerta;
    private int contador;
    // Use this for initialization
    void Start () {
        animation1 = door1.GetComponent<Animator>();
        animation2 = door2.GetComponent<Animator>();
        audiosourcepuerta = door1.GetComponent<AudioSource>();
        audiosourcepuerta.clip= puerta;
    }


    void OnTriggerEnter (Collider collider)
    {
        animation1.SetBool ("open",true);
        animation2.SetBool ("open",true);
        contador = contador + 1;
        if (contador == 1) {
            audiosourcepuerta.Play ();
        }

    }
}
```

Annex B. Exercises

VR COGNITIVE TRAINING EXERCISES PROPOSAL.
27_10_2017 MEETING
INTERACTION. DIFFICULTY LEVELS
CHOSEEN LANGUAGE: SPANISH,
Always ends the level going through the checkout.

Level can be chosen at the beginning. It applies to every training level.
Level 1 (low): Text list, time for memorize the list unlimited, no distractions, list always visible on the screen, product in close shelves.
Level 2 (medium): Text list, long reading time, list will disappear after a stipulated time. Products are separated on the supermarket. one distraction.
Level 3 (high): Text list, short reading time, list will disappear after a stipulated time. Products are separated on the supermarket. Two distractions.

Possible interferences:

- Supermarket worker that makes offers of the day.
- Person who exits of the supermarket and salutes the patient.

DATA COLLECTION
The system will have user's data management and will collect in real time the time of the activity.
Collected data for therapist:
Kind of exercise: time/exercise, number of errors (valid and non-valid), number of times the user consults the list on the levels two and three.
EPISODIC MEMORY EXCERSISE (VR FORMAT)
Usual shopping lists products (maximum 7).

1. Therapist offers a list prepared by himself, a basic products list without any relation between them or with any connection (menu, recipe, product category).

The items will be chosen based on the frequency of use and must be easily identifiable (milk, bread, oil, sugar, chips, apple, eggs, potato bag, walnuts, cheese, sardines, chicken, etc...).
2. After this the subjects must try to memorize the list of products that they have to find in the supermarket. That is why they can use different strategies. You can use the multisensory processing strategy (remember smells, flavors, textures or associations with colors) or more

elaborated verbal strategies (formation of acronyms, semantic association, among others). The therapist depending on the complexity of the task or the difficulty in remembering different material can use the strategies of learning without error, fading of tracks, or of spaced recovery. (These strategies are prepared before the VR activity and written together with the statement of VR activity).

3. An interference exercise that does not maintain any relation with the material worked is performed.

Supermarket staff that interacts and makes offers, or asks for help to place something in shelves, to count products or to find them damaged.

4. Participants are asked to try to evoke the list of the previously worked purchase. In case of difficulty, the semantic, phonetic or recognition facilitations (in this order) can be used. These activities are requested by the therapist after VR training.