I. INTRODUCTION

Since the invention of computers, scientists have wanted to devise virtual scenarios and environments with the maximum realism possible. They have also tried to create interaction with virtual elements identical to how we use them in real life. Over time, computer-generated graphics and sound have improved to the point of seeming real, now we just need to create the appropriate interaction. The purpose of this project is to create a computer program that allows interaction with many virtual elements as naturally as possible, creating a sense of absolute immersion in the three-dimensional scene.

This project is based in two funds, Virtual Reality [1] [2] and Human Computer Interaction [3]. The first involves the immersion of the user in a virtual environment generated by a computer; the second focuses on creating devices which allow the user to have a faster, more fluid and natural interaction with the computer.

Recently many devices have appeared in the virtual reality market, all of them aimed to videogame consumers, these devices have increased the sense of absolute immersion and finally have lowered their market prices.

In this project we have developed a simulator that allows you to move freely in a virtual environment and interact with the different elements that the user can find. These interactions concern: changing objects, activating some animations, moving freely in a three-dimensional space and editing the color and intensity of the stage lights. All of these whilst keeping the feeling of immersion in the virtual space.

II. VIRTUAL REALITY DEVICES & DEVELOPMENT ENGINES

The Project began investigating the vast majority of virtual reality devices that were bearable by the majority of videogame players: Oculus Rift [4], Kinect [5], Razer Hydra [6], Leap Motion [7] and Virtuix Omni [8]. Once all were analyzed, we chose a combination between them that provided the greatest sense of realism and immersion. Finally Oculus Rift and Kinect from Xbox 360 were chosen because their combination offered a greater sense of moving freely and immersing in a virtual world.

Oculus Rift (Figure 1) is a Head Motion Display that was carried out through a crowdfunding campaign in Kickstarter, needing the amount of 250.000$ in 30 days. In less than 36 hours, the campaign had over a million dollars and ended with an investment of 2.537.429$. Oculus Rift has a 7-inch screen and offers a view field of 110º. This outstanding feature is what makes it attractive and through it the user has a higher sense of immersion. The device contains gyroscopes, accelerometers and magnetometers to follow the rotations performed by the user’s head. These sensors are similar to those used in smartphones but with a greater precision.

Figure 1. Oculus Rift

Kinect (Figure 2), initially known as Project Natal is a device designed for the Xbox 360. It dispenses the remote control thanks to a sensor created by PrimeSense that detects the user's movement. Kinect works through a peripheral camera that can recognize gestures, the face, the voice, the player's movements and the static objects in their visual field. Kinect is a horizontal bar connected to a support containing a VGA camera that captures RGB image, a depth sensor and a microphone array.

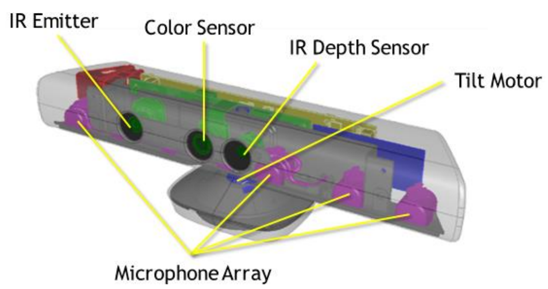

Figure 2. Kinect outline

The second step carried out was the choice of a development engine. Oculus Rift currently supports being developed using two game engines: Unity [9] and Unreal Engine [10]. Although Unreal has the best quality concerning final results, Unity, on the other hand, allows the integration of all the elements needed to make the simulator quicker and with better efficiency, giving good results in a short period of time [11]. Oculus Rift has its own SDK [12] provided by the company OculusVR, it is very easy to use due to its simplicity. Furthermore, Kinect offers a complete but also complex SDK that allows creating very different kind of projects. Due to its complexity, we need to program at low level which wasn't an aim of this project. The developer community has created several plugins for Kinect to help small groups of developers to carry out their creations. In our case, we have used a plugin called ZigFu [13] which transforms the data received from Kinect to a simple model in Unity (Figure 3).

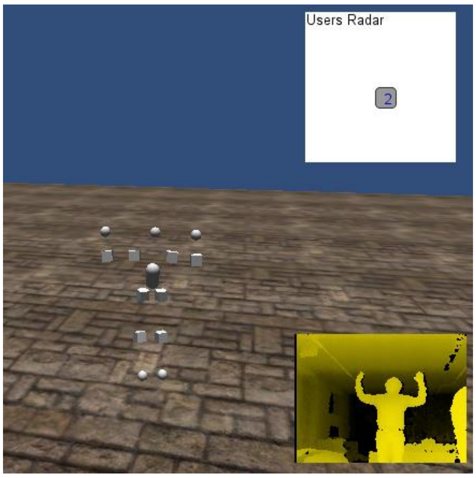

Figure 3. Example of ZigFu in Unity.

III. DEVELOPMENT

Once we chose the development engines we started with the development. The first task was the creation of the user movement in a three-dimensional environment. We had to be very careful with this aspect because the movements which had to be chosen had to be the most natural possible. The drawback was that Kinect does not detect the orientation of the users so they must always be facing the device.

At the end we followed the next methodology. First, the user is detected by Kinect. Then they are asked not to move along about fifteen seconds, during this period of time three values are calculated. The first of these values is the distance calculated between the user's feet. The other values are the angles of each of the user's shoulders. Using these three values, we can get several conclusions about the users performance.

If users separate their feet farther than the distance previously calculated, when making the gesture to walk, the virtual avatar will consequently start moving forward (Figure 4). The same will happen with the shoulders. However, concerning the case related with the shoulder rotation. If we know each shoulder rotation angle at their initial time, we can detect if the user is leaning their body towards one direction. As a result, the virtual avatar will rotate in the same direction as the user does (Figure 5). Later on, another action was set for rotations, using as benchmark the head.

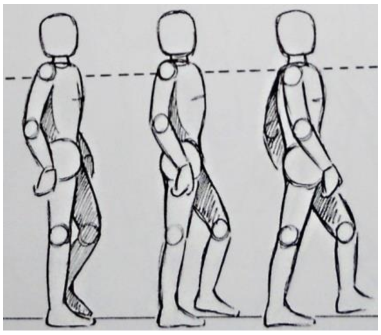

Figure 4. Doing the walk gesture the virtual avatar will move forward.

Once the code to detect the user's motion in three-dimensional space was designed, we had to develop the interaction with the scenario objects. After testing different types of interactions, it was decided that the easiest and most intuitive way was to implement a push (Figure 6) for each hand. A push is an applied force typically intended to drive or impel.

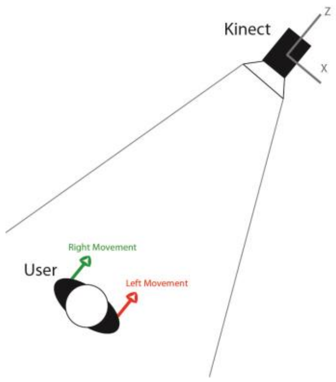

Figure 5. How Kinect detects the user's orientation.

After we implemented all the code, the results were the following: if the user gets in front of an object and makes a push with his right hand, he will have the possibility to change the object for another one; he will also interact with it, he can also change the color of the lights in the room. However, making a push with the left hand will let the object be moved around. To leave the object where the user desires he will make another push with his left hand, it will also be possible to change the intensity of the lights.

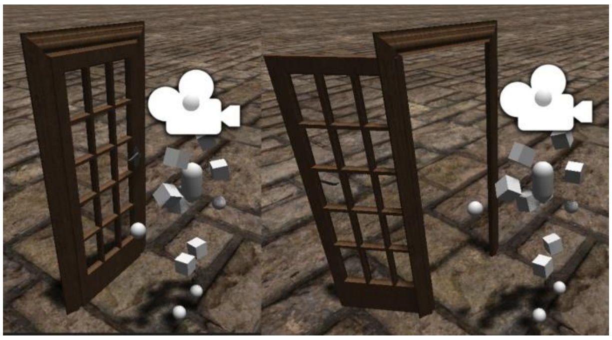

Figure 6. Interacting with the door.

Thus the actions that can be performed in the simulator are five: moving freely around the stage, changing objects, the translation of this objects, changing the lighting properties in the scene(color and intensity) and interact with animated objects.

IV. USER TESTS

Once the program was finished it was tested by thirteen users. The range of this test was twofold: check that users could move whilst interacting with the different elements and verify which of the two movements used in rotations was the one preferred by users.

The test consisted in four phases: completing the initial tutorial, changing objects for other ones, moving that new object to another place and editing de room lights.

The results were surprising as rewarding; most users achieved all the targets. When the test was over, they were asked several questions about their experience. Their responses were similar. For example, they said that they had never done anything which gave them that sense of immersion; they could perceive details from the objects in the virtual scenario as they had never done before such as the space between objects or their dimension. They also appreciated that various fixes were needed to make the environment seem more real like improving the quality of the Oculus Rift display, using sounds or implementing more animations.

Regarding the movement to activate the rotation, the users were asked which one they preferred, using their shoulders or their head. Very concise results were obtained: seven users preferred rotation using the shoulders and the other six using their head. From these results we came to the following conclusion: most users that chose the shoulder rotation had experience with first-person games. However, many of the users that chose the head rotation were not familiar with that kind of games.

V. SIMILAR PROJECTS

Although the emergence of virtual reality devices aimed at ordinary consumers is recent, some developers are already presenting their creations and even setting up their own businesses.

The largest project in this field is being conducted at NASA's Jet Propulsion Laboratory. They are using an improved version of the Oculus Rift and Kinect to control a mechanical arm [14].

In Aalto University, Finland. Tuuka Takala and Mikael Matveinen have started from a project of Oculus Rift to integrate in the scene Kinect, Razer Hydra and Playstation Move [15] to interact with the elements. Currently, this project is the most similar that exists. Although the way they interact is more natural, the only purpose that they follow is the reaction to the physical stimuli [16].

Another similar project is taking place at the University of Southern California where a team of 30 persons have created a videogame called Wild Skies, which uses Kinect, Oculus Rift, Razer Hydra and Playstation Move. Its main feature is that all the player's movements are tracked in a surface of 2 m^2 [17].

Arch Virtual is an American company that makes realistic three-dimensional environment which can be explored with the Oculus Rift. With costumers like Suzuki or the NBA team Sacramento Kings prove to be a company that has experience in this sector [18].

VI. CONCLUSIONS

This project has given us the opportunity to experience the key of virtual reality, the presence.

Presence is defined as being present. This concept can only be understood if you have previously experienced it. Presence is a fact that we ignore, as the act of breathing, we are not aware of doing it on purpose. We see, smell, feel… because we have a physical presence in the real world, without this we could not perceive any stimulus. This is the goal to be chased in the virtual reality, the user has to forget their presence in the real world and be transported to the virtual world.

To perform this process is not simple and having in mind that the main element of virtual reality is the Head Motion Displays it must have certain requirements:

- 20 ms motion-to-last-photon latency.
- 3 ms pixel persistence.
- 95 Hz refresh.
- 110º field of view.
- 1K x 1K resolution per eye.
- High-quality, well-calibrated optics.
- A millimeter-accurate tracking with a quarter-degree-accurate rotation.

However, there are other aspects that can influence the user's experience. We refer to the 3D audio, motion capture and interaction within the virtual scene.

VII. FUTURE GUIDELINES

The first purpose of this project is based on designing an architectural environment where the user can modify and interact with all the elements which they have at their disposal.

Nowadays many interior designers and architects have trouble showing their ideas to their customers. With this project, all these professionals have at their disposal a tool to visualize the result of their work and make all the changes that they want to [19] [20].

A highlight of this project is its easy scalability as it can be adapted to different disciplines. In order to change its functionality they should only need to change the different elements that are located in the virtual scene and they could even create new interaction from other gestures. One different application, for example, might be to create scenarios for videogames.

However, several industry experts say that it will take until 2015 to start seeing in the market virtual reality which can offer a competitive performance [21].

REFERENCES

1. Sherman, W., y Craig, A. (2002). *Understanding Virtual Reality: Interface, Application and Design*. USA: The Morgan Kaufmann Series in Computer Graphics, http://deca.cuc.edu.cn/Community/cfs-filesystemfile.ashx/__key/CommunityServer.Components.PostAttachments/00.00.00.07.15/1558603530-VirtualAh.pdf
2. Vrs. *What is Virtual Reality?* http://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html
3. Kshitiz, A. (2001) *Introduction to HCI*. http://es.slideshare.net/deskala/introduction-to-hci
4. Oculus VR. http://www.oculusvr.com/
5. Microsoft Developer Network. *Kinect for Windows Sensor Components and Specifications*. http://msdn.microsoft.com/en-us/library/jj131033.aspx
6. Razerzone. http://www.razerzone.com/es-es/gaming-controllers/razer-hydra-portal-2-bundle
7. Leap Motion. https://www.leapmotion.com/
8. Virtuix. http://www.virtuix.com/
9. Unity. https://unity3d.com/
10. Unreal Engine. https://www.unrealengine.com/what-is-unreal-engine-4
11. Méndez, R. (2014). *Unreal Engine 4 vs. Unity – a quick overview*. http://arges-systems.com/blog/2014/05/12/unreal-engine-4-vs-unity/
12. TechTerms. (2010). SDK. http://www.techterms.com/definition/sdk
13. ZigFu. http://zigfu.com/
14. OpsLab JPL, Youtube. (2013). *NASA Robot Arm Control with Kinect*. https://www.youtube.com/watch?v=pqNC72fgetc
15. Engadged. (2010) *Playstation Move motion controller*. http://www.engadget.com/products/sony/playstation/move/motion-controller/specs/
16. RUIS VR, Youtube. (2013). *TurboTuscany: Oculus Rift+Kinect+Razer Hydra+PS Move*. https://www.youtube.com/watch?v=-YYiTkf3sDs
17. UscGames, Youtube. (2013). *2013 Spring Demo Day Holodeck*. https://www.youtube.com/watch?v=HGNx8ZgnIPw
18. ArchVirtual. http://archvirtual.com/
19. King, T. (2005). *Architecture's Virtual Shake-Up*. http://news.bbc.co.uk/2/hi/programmes/click_online/4385006.stm
20. Brouchod, J. (2013). *11 Quick Oculus Rift Development Tips for Architectural Visualitzation*. http://archvirtual.com/2013/10/07/11-quick-oculus-rift-development-tips-for-architectural-visualization/#.VAhVU_l_tqU
21. Abrash, M. *What VR Could, Should, and Almost Certainly Will Be within Two Years*. http://media.steampowered.com/apps/steamdevdays/slides/vrshouldbe.pdf