Virtual Reality Training Simulation for Controlled Land Unmanned Vehicle: A Feasibility Model

Lozano González Jorge, Neira-Tovar Leticia, Cruz Juan, and Sanchez Eduardo

Abstract Today, the use of technology is indispensable in daily life, being this an important part to facilitate to the greatest extent possible the activities carried out by human beings as an example of the scope that technology has and the impact it generates in our lives. Having its origins in the last century, virtual reality has become one of the most sophisticated systems with different applications in different environments. This chapter presents an overview of a virtual reality (VR)-based training tool for users to get qualified as pilots capable of handling real-life unmanned vehicles. This work shows some characteristics of VR systems, and the scope is to present the advantages of using virtual reality software as a training tool, promoting that virtual reality training for use of an unmanned vehicle could be as effective as real-life training.

Keywords Virtual reality · Augmented reality · Simulation · Unmanned vehicle · Training software

1 Introduction

Pioneering organizations such as GE and Boeing are using virtual reality (VR) to improve productivity, quality, and training [1]. By combining the power of humans and machines, VR will significantly increase value creation for organizations [2]. VR is a very high-end computer interface that evolves real-time simulation and interface through numerous sensorial channels. These sensorial modalities are visual, aural, tangible, smell, taste, and other senses [3]. On its part, augmented reality (AR) is a new technology that involves the overlay of computer graphics on the real world [4]. They both aim to extend the sensorial environment of an individual by mediating reality through technology. The former relies on an

L. G. Jorge (✉) · N.-T. Leticia · C. Juan · S. Eduardo
Universidad Autónoma de Nuevo León, San Nicolás de los Garza, Nuevo León, México
e-mail: jorge.lozanogz@uanl.edu.mx

F. Torres-Guerrero et al. (eds.), *2nd EAI International Conference on Smart Technology*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-031-07670-1_6

alternative setting to experience, while the latter improves existent elements with additional layers of meaning [5]. As we can see, these have many similarities, but differences as well.

In this work, we are going to focus on VR systems; VR is known as a technology that makes possible for people an experience in a world beyond reality. Such technology comes in different ways and sizes, with the aim of giving information to our senses: sight, hearing, and touch [6]. This technology qualifies better for this project thanks to its features. VR systems immerse the user in a virtual world that can be created according to the characteristics we want it to have.

Although the use of virtual reality is, in general, highly related to the world of video games and entertainment, we cannot ignore the applications that this may have, such as the formation of new employees in different industry sectors; within medicine, the most representative applications are simulators for medical training, surgery operations, treatment of phobias and psychological trauma through exposure therapies and pain management through distraction techniques, and others. In addition to providing technology with the possibility of disseminating knowledge and interacting with it, once each training program is invested, the expenditure on employee training will also be reduced. As we mentioned, all of these provide employees with a more attractive and effective environment, giving them more motivation, interactivity, or connectivity without any risk for the user or monetary losses for the company.

Based on the above, the project seeks to implement a video game-type simulator that tests the skills of users who test the simulator, thus improving their ability to control an unmanned land vehicle to perform different activities with real prototypes without additional costs or material damage that could cause delays in the processes in where they are used. To achieve this, it is recommended that the simulator has an intuitive interface that any user can control to make your experience more satisfactory. In addition, some levels have been developed so that according to the standards of user experience, the user can choose the level of difficulty he needs to improve his skills or, on the contrary, gradually increase the level and get a better training.

Virtual reality devices are expected to grow rapidly over the next 5 years, with an average CAGR of 106% and combined shipments of 43 million in 2020. Augmented reality smart glasses will experience similar growth and ship 21 million units in 2020 with a CAGR of 78%. “Sales of VR Head Mounted Displays, including mobile-based devices like the Samsung Gear VR and Google Cardboard, as well as tethered devices like the Oculus Rift and Sony’s Morpheus, will be driven by the release of high-profile devices and a growing awareness and interest in the technology from consumers” [7]. Due to the above, it is expected that the project will be a reliable and effective tool for companies in different sectors in which a prototype of both the simulator and the unmanned vehicle can be useful.

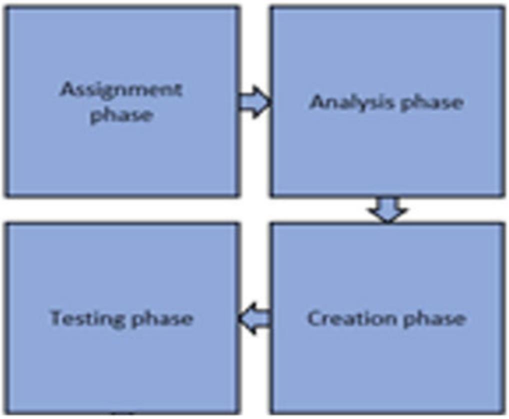

Fig. 1 Phases of methodology

2 Method

Virtual reality (VR) uses simulation and displays to trick humans and other organisms into believing they are having a perceptual experience that is different from reality. This experience is usually interactive and carefully crafted, such as games exploring exotic worlds [8]. The effectiveness of the VR-based training process depends directly on the quality of the prepared training material or on the quality of the prepared virtual training environment. A detailed overview of the product documentation and the process of product assembly (installation maps, design drawings, 3D models) enable the definition of all the elements necessary for defining the training process [9].

The method employed is based on a "Methodology for Designing Virtual Reality Applications," after rigorous evaluating sessions with people who had been involved in previous virtual reality projects and analysis of other methodologies mentioned inside the document as well. After deployment on the first projects, they made some corrections, and this methodology was further enhanced [10]. This work is based on four main steps (Fig. 1).

2.1 *Assignment Phase*

The fundamental elements that determine the impact that any education technology tool can make are informed and determined not only by its cost or speed or accessibility but also by its usability for a diverse and inclusive society [10]; therefore, in this phase, we check the main ideas of the overall concept of the virtual reality application we are going to develop; we must be able to describe the requirements and purpose of our project. Basically, feedback was given on what we wanted to achieve with the release of the project.

First of all, a brainstorm was done to have different options on how to implement a project that meets the expectations but that mainly fits within the time that we had to be fulfilled. There were many complications about how the project would be carried out, as there were many ways to achieve it. However, not all of them fit the characteristics that were mentioned; therefore, taking ideas from UAV's flight simulators, it was decided to develop the simulator as a video game, in which the user's task is to manipulate the vehicle by making it go through an obstacle course, thus picking up the objectives, being this one the best option for development process due its difficulty and best training effectiveness.

For convenience and better training, the simulator was designed to contain ten levels, each one more difficult than the previous ones. Each level was designed with the same purpose, but the easier ones were designed to learn most basic movements of the vehicle, such as directing on a plain with simple obstacles, while in the harder ones, we can see an increase in the difficulty like having alterations as slopes and holes, making it more difficult for the driver, because it no longer has to face these changes, but requires precise control of the vehicle to avoid falling over the edges and other obstacles.

2.1.1 2D Unity Requirements for the Simulator

1. Creation of a simple main interface with the name of the simulator and with noncomplex navigation. The important thing is to capture the user's attention by not being so invasive.
2. Creation of the submenus corresponding to each button. The design of all of them remains similar to the main menu but with their respective title and different options in each one.

2.1.2 CAD Requirements for the Simulator

1. Build a virtual model of a prototype unmanned land vehicle that will be used as our object to control throughout the tests of the simulator.
2. To design levels, which will gradually increase in difficulty, requiring the user to have different skills and making him go through different situations through which he will have to go through to complete the different levels and conclude if the user is able to control a prototype in real life.
3. In each level, there will be objectives in the form of medals, which will have to be collected to complete it. The design of these medals will also be carried out for their subsequent placement in the simulator. Since the levels will be surpassed once the user manages to collect all the medals, it is important to consider a design that is striking and easy to locate for users.
4. The construction of a pad, in which the user must be placed to later move to the next level.

2.1.3 Unity Editor Requirements

1. Import the models in .FBX format and add them to the “Models” folder.
2. Development of scripts for the physics of the simulator.
3. Placing the objects and respective textures to each one.
4. Edition of the cameras and lighting.
5. Programming the actions of the objects, as well as the physics ones of each one.

2.2 Analysis Phase

In this section, we analyzed what would be the necessary requirements for the creation of the simulator: the selection of actions and objects and designs that will be placed in the final program. Processes establish how methods are applied and software projects are managed. Methods supply technical descriptions on how to perform different tasks of software engineering, such as designing and building software. Tools include computer programs used for planning and implementing software, as well as otherwise supporting the process [11]. In this phase of software development, the goals of Software Safety Program are to eliminate, reduce, or control the possible hazards associated with potential software failures. Software safety requirements may include national/international standards, customer requirements, or corporate needs [12].

The first thing that had to be thought about was the menu design. The menu shows the logo and the name of the simulator at the top; the name LUV-SIM originated from the combination of the initials of land unmanned vehicle with the word simulation. Since this is the presentation of the simulator and it must achieve a positive impact on the user, using a simple and friendly interface in which he can make the adjustments he thinks is necessary, in addition to having the option of selecting the level of his preference into the play menu.

Regarding the structure of the simulator, the main thing was to choose how the levels would be structured and which skills would be required in each of the proposed levels and how they would be distributed to achieve adequate training. The objective at each level is simple, to collect the floating coins found along the map and then place the vehicle on a podium that will shine exclusively when the user has the coins found at each level. Levels were designed so that the user can first learn the basics, such as the basic movements of the vehicle, as well as get the user to adapt to the controls so that with the passing of the levels and already with some adaptation, he can successfully conclude each of the challenges presented.

As for the structure of the vehicle, it is an original model that is under development but serves, for the moment, as a representation of the prototype used exclusively for this simulator. The most significant details of the model are the adaptation of cameras and LEDs that the pilot of the prototype can use according to their needs (Fig. 2).

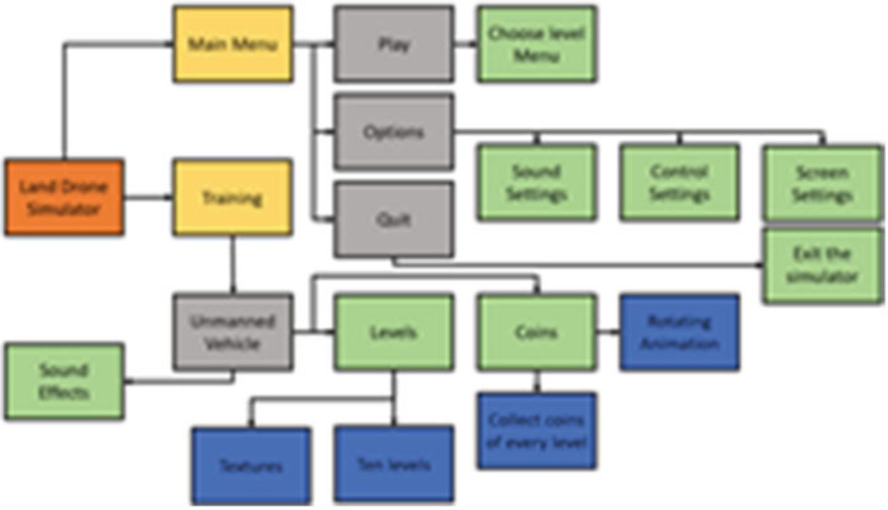

Fig. 2 Diagram of analysis phase

The next step is to consider the actions that each of the models will perform within the simulator environment so that the system meets its objective:

1. The vehicle must go in the desired direction, either forward, left, or right, and in reverse.
2. The collision property will be added to each one of the objects that require it; these will be those whose function is to mark a limit or must be immovable for training purposes.
3. The coins should be rotating to take advantage of their texture and be located more easily for users.
4. Each time that each of the located medals is collected, it should disappear both on the map and in the simulator objectives count.
5. If the user has not collected all the medals on the map, pad must not allow him to complete the level.
6. Wheels of the vehicle must move whenever the vehicle is directed in any direction.

2.3 Creation Phase

The simulated environment can be like the real world to create a lifelike experience, for example, in simulations for pilot or combat training or it can differ significantly from reality, such as in VR games [13]. Because of this, several factors must be considered when creating a virtual reality environment. The first thing was to think about how real the simulator should look, to make more dynamic the way in which

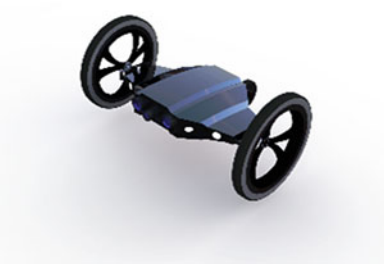

Fig. 3 Land unmanned vehicle model isometric view

the user develops and aesthetics; it was decided to be a more unreal environment and look more like a futuristic video game.

The development of the project was carried out in different phases and software, of which the first was to develop the structure of the unmanned vehicle that will be used for the tests in virtual reality.

The vehicle model and its parts were made in SolidWorks because of its ease of use and later the assembly was made. Levels were made in SolidWorks to design both walls and floors for each level with no materials selected, to be assigned later in another software.

2.3.1 CAD Design

The measures that were given to the vehicle were assigned considering an acceptable size for the vehicle to meet the applications that would be given (rescue, transport, exploration, etc.). The hitbox dimensions given for the side wheels were 172 mm, being the total height of the vehicle, besides being the largest measure of all. The width of the vehicle is 314.72 mm, and finally, the depth of the vehicle is a total of 210.64 mm. However, the total volume of the vehicle is 772269.89 mm^3 (Fig. 3).

The complexity of the design is justified by the fact that the physical prototype can carry out tasks in which human intervention is difficult and requires human intervention and that it has the tools to carry out the task. It consists of a model that has two wheels at the sides and one at the back to maintain balance; the front is shown with a camera, as well as a night vision sensor and LEDs to facilitate vision as shown in the figures below.

On the other hand, the design of the levels was carried out by means of the difficulty that these would suppose for the user. Each level was planned to require the user in different facets or challenges that must be faced. The difficulty varies depending on the level, with number one being the easiest and the level with the highest number being the most difficult. Throughout the ten levels, users are tested so that the right challenges are demanded from the user and better training is completed. General rules were maintained for the creation of the levels. The walls of all levels have a height of 3000 mm so that in case of a bug or glitch in the simulator, do not leave the assigned play area. Also, the walls are 100 mm thick and the floor is

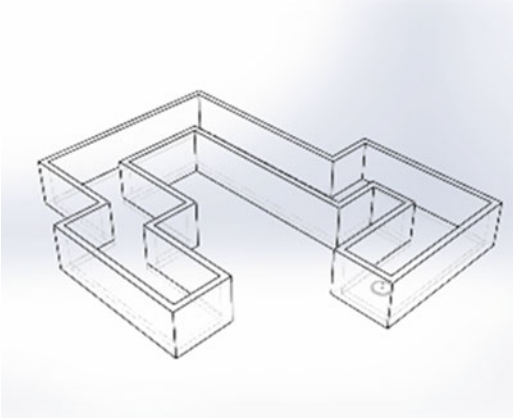

Fig. 4 First level isometric view (hidden lines visible)

Fig. 5 Pad final assembly (no color or textures given)

750 mm wide. All these measures were assigned to each of the levels regardless of the difficulty, to achieve a standard and not have variations between them (Fig. 4).

Finally, the design of the pad in which the vehicle must be placed at the end of each test was constructed. This pad consists of an assembly also designed in SolidWorks. The design is a circle with a diameter of 300 mm and a height of 14.66 mm, the latter in order not to be difficult to climb for the vehicle but nevertheless be prominent from the rest of the environment (Fig. 5).

2.3.2 Blender

Once all the models needed were made, both levels and the model of the vehicle, we had the need to export them to the unity software. For this, the first thing we had to do was to export each of the models in .STL format to be able to modify scales, rotate angles, and convert them to .FBX format for export to the Unity editor where the final project will be done. The image below shows the model of the vehicle in blender for later export (Fig. 6).

Once the models are processed to the corresponding format, we can use the models created in the Unity editor to add textures, place the objects in the corresponding place, and continue with the next step which is the creation of scripts, the addition of physics, and final programming.

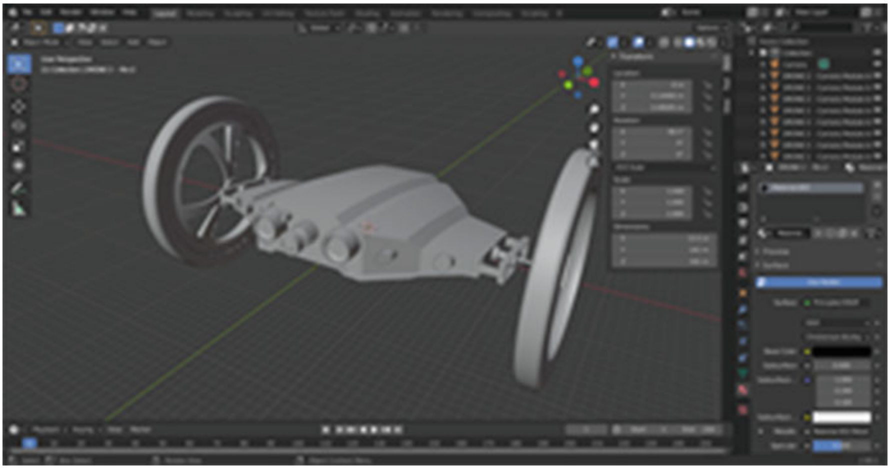

Fig. 6 Unmanned vehicle model in blender

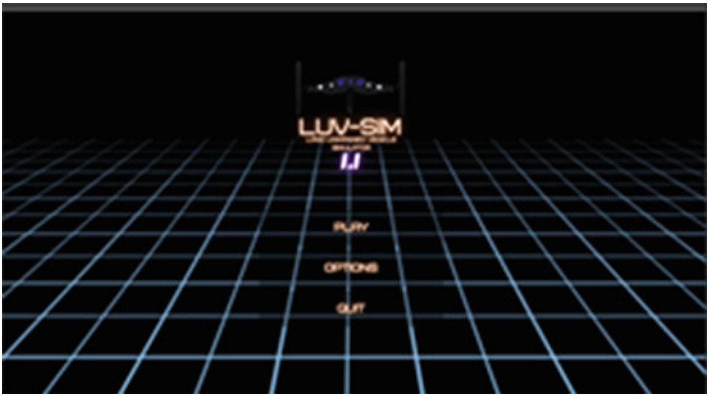

Fig. 7 LUV-SIM main menu

2.3.3 Unity Editor 2D

The first part, already concentrated only in the Unity editor, was the creation of the menu in a 2D project in which the logo was placed at the top of the screen with only three buttons so as not to make the user interface so invasive (Fig. 7).

The design selected for the background kept the idea of a futuristic design so the background being the first selection to be made had to convey this idea, and the grid was selected with a blur, imitating a little the look of the levels inside the simulator. We added the background by right-clicking on the scene where we will place our background, then select the "UI" tab where "Image" was selected to then select the parameters that fit what we wanted to achieve, such as resolution, size, etc.

Fig. 8 LUV-SIM 1.1 logo

Later, we selected the logo. The logo was created in the Canva web software. This design consists of an image of the vehicle with the title already mentioned, "LUV-SIM" adding its version (Fig. 8). This was added just like the background but with different parameters that were selected specifically for the logo, for example, the size and the coordinates where it will be placed. The buttons were added with the "Button Text Mesh" option in the "UI" menu.

By default, the Unity editor adds text to this type of buttons, but it was replaced to have a more uniform structure for the menu and respecting the same font and colors that were given with the logo, since all this is shown in the same scene. Due to this, we exchanged the text for a centered image that shows uniformity in the menu, since it corresponds to the typography of the title and respects the aesthetics of the futuristic idea.

Since now, an image was shown and not text on the buttons, we had to adjust some parameters such as the size of the buttons and find a way they could fit, since the size of the image and the location had to be modified because they were different. Also, it was necessary to select the positions that each button would occupy, so it was decided that the option of placing them in the form of a vertical line would be the most efficient way to not occupy much space on the screen.

However, the simplicity of the menu does not mean that no more options can be placed, since each of the buttons opens submenus in which, for example, the "PLAY" button will open a screen in which the desired level can be selected (Fig. 9).

This section, as its name says, gives us the option to choose the level we want to select, according to the ability of the user. For this, we use the same methodology as in the main menu. We kept the background and the typography handled, being monotonous for all the menus of the simulator and a logo in the upper part, with the difference that in the upper part would be placed the name of the menu in which we

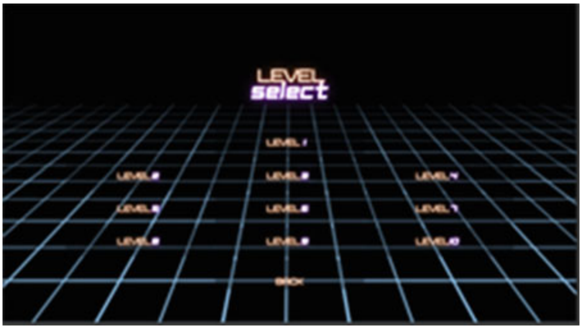

Fig. 9 Choose level menu

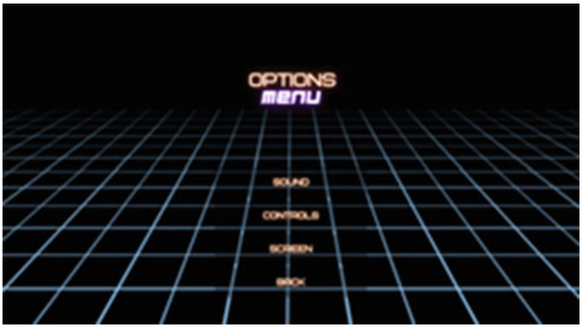

Fig. 10 Options menu

were, in this case the level selection. Each of the buttons maintains the size of the main menu but with a different distribution so that the ten level selection buttons and the button that would take us back to the main menu would fit. In terms of style, the typography was kept as mentioned above, but with the addition that we now had numbers, so it was decided that we would change the color of these to achieve a different aesthetic without losing the essence that was built with the main menu.

Also, the “OPTIONS” button will open a menu in which you can adjust different parameters such as volume, screen settings, and controls (Fig. 10). Keeping the style already proposed in the previous menus, the buttons were placed in a similar way to the main menu with the addition of a back button to the main menu.

And finally, we will have the “QUIT” button that will serve as the output of the program.

2.3.4 Unity Editor

The Unity editor part is the part where the most work was done, due to the development of the scripts and the programming done.

The first part of the process was to import all the models that were designed to be placed in the scenes, which will later become the levels of the simulator. The 3D Models were placed in a folder called “Models” in Unity that was created to put there everything necessary for the creation of each level; this includes the level

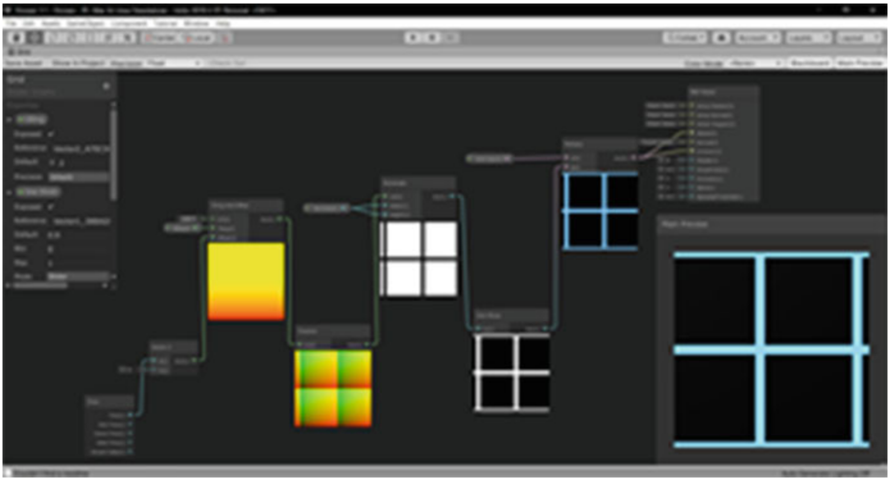

Fig. 11 Grid properties and parameters

itself, the medals that will be placed to complete the level, and if necessary, we must import the pad itself, because although in most levels is already included in the final assembly made before, in case of an error must be placed again and being a completely independent part of the walls and floor, we must return to place it manually. The explanation of the textures will be given by taking the example of a level.

The addition of the grid texture for the walls and floors was done from an animation made first by placing a square with the command rectangle and giving it fraction properties making it multiply along all the structures. Later, the tiling and offset parameters were added with a timer and multiply to generate the motion animation. The connection of these parameters is shown in the figure below (Fig. 11).

Since the vehicle textures had already been added previously in blender, we do not need to place another one for this one. The only ones that were added were to the missing structures that are the pad and the medals. These were not given a dynamic texture since they are moving objects. We just had to add a solid texture that would fit the look of the simulator. The medals, being the main objective of the simulator, were given a golden texture that could be easily seen by the user (Fig. 12).

Once the textures were finished, the scripts were configured with the physics needed to make the simulator as similar as possible to a model built in real life. Based on the considerations mentioned in the analysis phase, the necessary structures were given collision properties. As mentioned before, the structures that will need this type of property are those whose function is to stop or be an obstacle for the user, taking into account that the vehicle needs them in a mandatory way, since it is the object of training. In addition to this, they had to have considerations like those of gravity, since several aspects will be tested along the levels of the

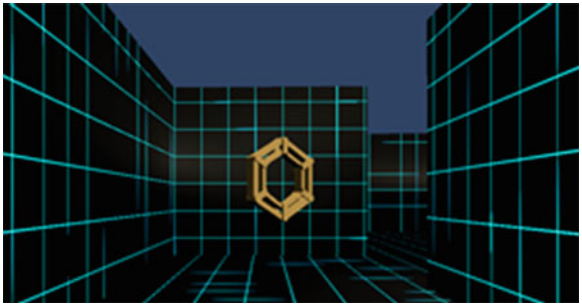

Fig. 12 Texture of the medal compared to the background

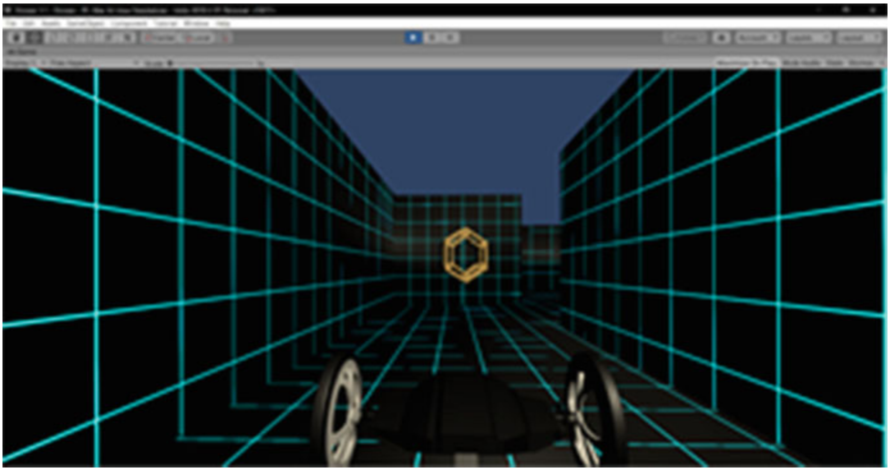

Fig. 13 Illumination and camera settings

simulator, for example, in the levels that have the difficulty that have more complex structures with different levels in the structure or in those that have edges in which the user could fall. Finally, about physics, we worked with the placement of the medals that will be found along the maps. To these, we added an animation that makes them go around, making it easier for the user to identify them.

After finishing with the physics of the simulator, we continue with editing the cameras and lighting. Given the atmosphere of the simulator, we opted for a low lighting to achieve a more striking immersion for the user. As for the work in the cameras, this was anchored so that the user's view was that of a third person perspective, since for a user who has never had a similar experience, the first person could have side effects on the user. Both illumination and camera settings are illustrated in the image below (Fig. 13).

3 Testing

A software product should only be released after it has gone through a proper process of development, testing, and bug fixing. Testing looks at areas such as performance, stability, and error handling by setting up test scenarios under controlled conditions and assessing the results. This is why exactly any software has to be tested. It is important to note that software is mainly tested to see that it meets the customers' needs and that it conforms to the standards [14].

As for the menus, in the simulator, each of the buttons fulfills the expected function, and each of the directions they should have were checked, for example, the level selection menu is opened as soon as the "PLAY" button is clicked.

Regarding the levels or maps, as far as the levels or maps are concerned, we first tested the work of the textures and that they have been placed correctly on each of the objects without causing interference.

After checking the textures in all the structures, the next step was to test how the simulator works in different facets. The first one was to check that the movements of the vehicle were as indicated, for example, that when a joystick or button that is programmed that the movement of the vehicle was to the right accomplished its function. For this purpose, a level was randomly selected and then analyzed the movements by pressing the buttons independently, as well as all the combinations, resulting in all cases was fulfilled the objective because in each of the buttons that were assigned to each direction achieved its task as well as the combinations, for example, forward and right and to change their direction.

The next part of the testing of this project was to test that the physics developed with the scripts were the most appropriate and in the same way that they worked correctly. The goal was to analyze that the given physics work together with the vehicle movements in the simulator to achieve the most realistic experience possible and to achieve a more complete training consequently. Different levels had to be studied to test the falls and the behavior of all the elements that make up a level; for example, if the vehicle falls from an edge to another surface, it would be necessary to check that this surface had the correct properties and that the fall of the vehicle was as close as it would be in real life. Given that several of the maps contemplated to test the user in situations where he must overcome obstacles such as bridges, false paths, etc. have two or more levels in their structure, it was necessary to check how he would behave in the case where there was a surface. It was also necessary to prove that when there was no surface, the vehicle would return to the part from which it fell, and an attempt would be discounted from those the user counted.

Another aspect to consider in the physics was the programmed movements that were considered when programming and developing the scripts. In this analysis, it was found that the medals performed the programmed movement correctly and that the wheels start moving when the vehicle moves forward, so this part also fulfills its purpose.

4 Conclusions

Once the steps that were considered for this project were completed, the corresponding tests were performed, and the results were analyzed, it was concluded that the main objective was partially fulfilled; this is because the operation of the simulator was desired in each of its facets, which makes it a viable option for training, since it tests several aspects to be taken into account in the development of a training software in a correct way. However, the tests conducted do not reflect the expected result, because the main idea is to train a user who would pilot a real prototype and, from this, analyze whether the training that is carried out with virtual reality technology is effective in this area, comparing the real training with this simulator; this will be the scope for the next paper. At this work, it is concluded that the simulator meets all the aspects that were taken into account for the development of it. Although there are details that could be improved to increase the quality of the user experience and the tests have been minimal due to external circumstances, the simulator meets the requirements, and it is estimated that it would be a viable option if at some point a software of this type is required, given that the tests are sufficiently demanding to improve the handling and that it would also require the most experienced pilots.

References

1. M.E. Porter, J.E. Heppelmann, Why every organization needs an augmented reality strategy. Harv. Bus. Rev. **95**(6), 46–57 (2017)
2. S. Benford, C. Greenhalgh, G. Reynard, C. Brown, B. Koleva, Understanding and constructing shared spaces with mixed-reality boundaries. ACM Trans. Comput. Human Interact. **5**(3), 185–223 (1998)
3. M. Kundalakesi, T. Swathi, B. Ashapriya, R. Sruthi, A Study of Virtual Reality. Department of BCA and MSc SS, Sri Krishna College of Arts and Science, Kuniamuthur, Coimbatore, India (2017)
4. R. Silva, J.C. Oliveira, G.A. Giraldi. *Introduction to Augmented Reality National Laboratory for Scientific Computation*
5. E. Gandolfi, *Virtual Reality and Augmented Reality* (2018)
6. A. Tocu, A. Gellert I-R. & Stefan, T-M. Nitescu, G-A. Luca, The impact of virtual reality simulators in manufacturing industry. https://doi.org/10.21125/edulearn.2020.0905 (2020)
7. Abi Research. Virtual Reality Devices to Ship 43 Million Units by 2020, with Mobile-Reliant Head Mounted Displays Foremost, Oyster Bay, New York (2015)
8. M. Suomalainen A. Nilles S. Lavalle. Virtual Reality for Robots (2020)
9. U. Marjanovic, S. Tegeltija, N. Medic, M. Lazarevic, N. Tasic, B. Lalic, Content Development for Virtual Reality Training University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovica 6, Novi Sad, Serbia (2018)
10. J. Polcar, M. Gregor, P. Horejsi, P. Kopeček, Methodology for Designing Virtual Reality Applications. https://doi.org/10.2507/26th.daaam.proceedings.107 (2016)
11. T. Takala, A Toolkit for Virtual Reality Software Development – Investigating Challenges, Developers, and Users (2017)

12. S. Oveisi, M. Farsi, A. Moeini, Software safety design in requirement analysis phase for a control system (2019)
13. S. Harms, J. Hastings, Enabling Student Innovation through Virtual Reality Development (2016)
14. D. Raghuvanshi, Introduction to Software Testing. 797–800 (2020)