2019

Interactions Example Scene Delivery

VIDEOGAMES AND VIRTUAL REALITY

JAIME RAMOS TORNERO
4º COURSE – DEGREE IN COMPUTER SCIENCE
YEAR: 2018/19
FACULTY OF CASTILLA LA MANCHA

Content

Table de illustrations

1. Introduction

The definition of virtual reality comes, naturally, from the definitions for both 'virtual' and 'reality'. The definition of 'virtual' is near and reality is what we experience as human beings. So the term 'virtual reality' basically means 'near-reality'. This could, of course, mean anything but it usually refers to a specific type of reality emulation.

The way we achieve it is by making some kind of interaction between the user and the environment, for instance, by using head-phones, head-mounted displays and, of course, a powerful computer. Furthermore, there is a tool called SteamVR that allows the user to be involved in the game by using its headset HTC Vive. This tool is connected via hardware to our computer and it will be in charge of simulating the interaction into the virtual reality.

With this in mind, this document is intended to attract the user to use virtual reality, and to give a clear explanation of how to implement our interaction system by using the assets that Unity includes. The one that will be given much more importance will be the SteamVR Plugin. Overall, the aim of this task is:

- Break down one of the Interaction System examples, describing it in a clear way. In our case, we will take the Throwables[1] example as a case in point.
- Once the example has been analysed, we will create a mini-game in which we are going to extend the functionality of the mentioned example.

2. Preparing the environment

First of all, in order to test our game in a real computer we have to know that there are some requirements that need to be fulfilled.

2.1. Hardware requirements

The testing of the game takes place in a Virtual Reality environment that our headset provides; in this case we are going to use the **SteamVR/OSVR-compatible Head-Mounted Display (HMD)**, which is called "**HTC Vive**" and **HDK 1.4** used in this course. Bear in mind, that all the **USB, HDMI** must be connected properly to the computer.

2.2. Software requirements

Our next step to be carried out is lunch the Steam program and inside it the **SteamVR plugin**. After all, if it is not installed on the computer we have to install the **Unity 2017.1.0f3**, concretely this version which is compatible with our **HTC Vive controls** and **head-set**. These are the links for downloading the programs:

- **Unity 2017.1.0f3**. https://unity3d.com/es/get-unity/download
- **Steam**. https://store.steampowered.com/?l=latam
- **SteamVR Plugin.** https://store.steampowered.com/app/250820/SteamVR/

[1] A throwable object means an object that the player can throw anywhere whenever he is playing.

3. Building the game

This section is going to be divided into two sections. Firstly, we are going to describe briefly the game that we want to implement. Secondly, we are going to add our game objects, so that we can add functionality: interactions, throwables and teleporting.

3.1. Game description

Our game will consist of a plane with some objects and some teleportation points. The plane includes a blue huge cube situated in the center of the plane and 4 yellow cubes. All of them are placed on a surface (red plane) which will also include 4 teleport points and a teleportation central area.

The aim of the user here is to put him in contact with Virtual Reality by walking around the map, teleporting him and the capability of grabbing and throwing objects. This game is developed with the purpose of giving knowledge for the new people that want to use Unity. For this reason, we are going to explain all the steps of how to implement it and test it with the user. In fact it is a great example for the new people to test it as it includes the throwables and teleporting interactions which are very straightforward to implement.

3.2. Game construction

Now, we are going to explain step by step how to implement our game in Unity. In fact, it is very straightforward.

The first thing you need to do is open **Unity**, go to **New**, name the project '**InteractionTutorial**' and pulse the button **Create Project**. In this step, Unity is going to load its packages to launch and prepare a new project. Now we are in the project:

1. In the hierarchy, **bottom-right button** of the mouse to create a Plane: **3D Object > Plane**. In order to center the plane we are going to introduce this values in the inspector:
    - **Position.** X:0 Y:0 Z:0
    - **Rotation.** X:0 Y:0 Z:0
    - **Scale.** X:1 Y:1 Z:1

2. Now, we create an object, in this case it will be a **Cube**: **right-button > 3D Object > Cube**, you will need to also set its **values** in the inspector as the same we have described before.

3. We are going to give some colour to our **Cube Object**. Locate yourself within the **assets directory** and then **right-button > Create > Material**, we name it **BLUE**, go to **Albedo** in the inspector and select the **blue colour**. After that, **drag** the material and **drop** it into the cube so that you can notice the cube is now **blue**.

4. What we have to do now, is download the **SteamVR Plugin** from the **Asset Store**, we will need it to add the interaction to the objects we are going to create. For this purpose, go to **Asset Store tab** and write **SteamVR Plugin**, **download** and **import** it.

5. Now that we have downloaded and imported the plugin we look for the **Player prefab** in the **Assets browser**, we **drag** and **drop** it into the hierarchy (illustration 1).

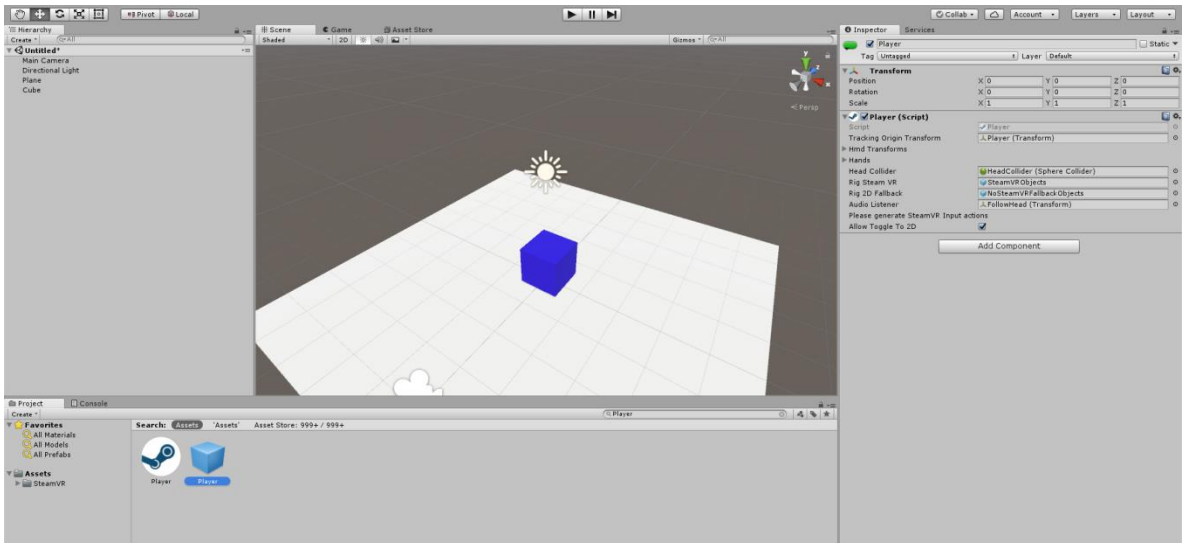

Illustration 1: Adding the Player into the scene

6. Delete the **Main Camera** as we are not going to use it anymore. Place the **Player** a little bit far from the Cube with the mouse.

7. If we would like to test our game, what we have to do is pulsing the **Play button**, some windows will appear, click on Yes every time. Then, a **SteamVR Input** window will appear, click on **save and generate**. Close all the windows, and pulse "**Play"** again. Now you can note that the Vive controls are in the scene.

If you are in this point it means you are going well. Congrats! Although it seems very easy. Now, we are going to add the interactions, first of all, the teleporting.

1. When we are placed in the **Assets browser directory**, we look for the **TeleportPoint** (illustration 2). We **drag** and **drop** it into the hierarchy.

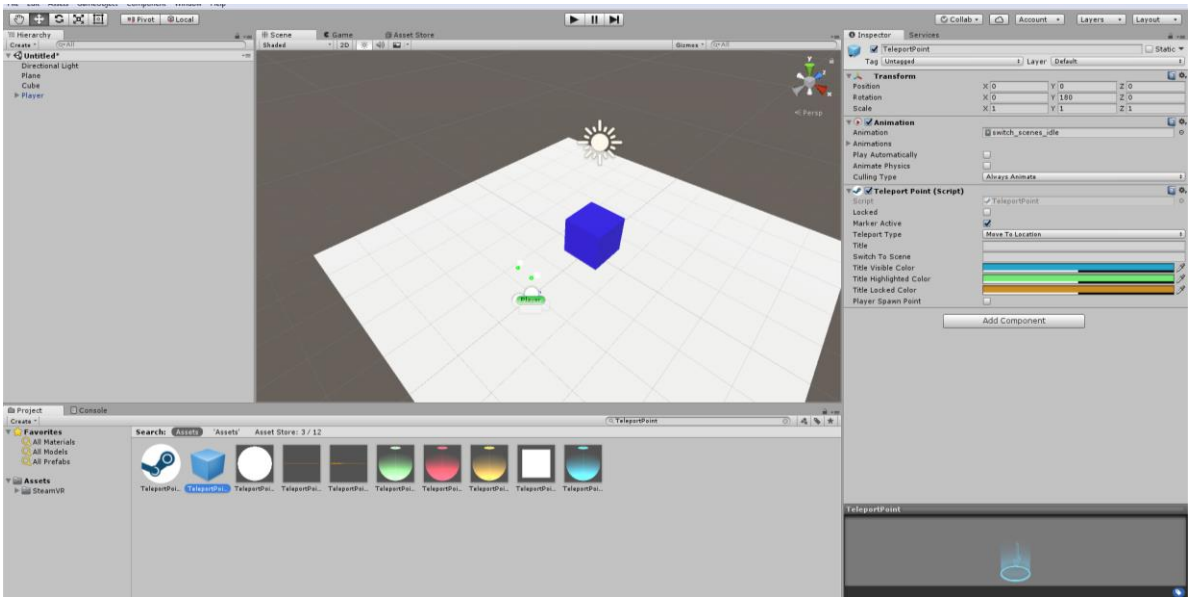

Illustration 2: Looking for the TeleportPoint

2. Create again a **new material** in order to colour the plane, choose a nice **red**. **Drag** and **drop** it into the **Plane**.

3. We place the **Teleport Point** in the corner of the **Plane**. We duplicate the **TeleportPoint Objects** so that we can have 4, and we place them in the remaining corners (illustration 3).

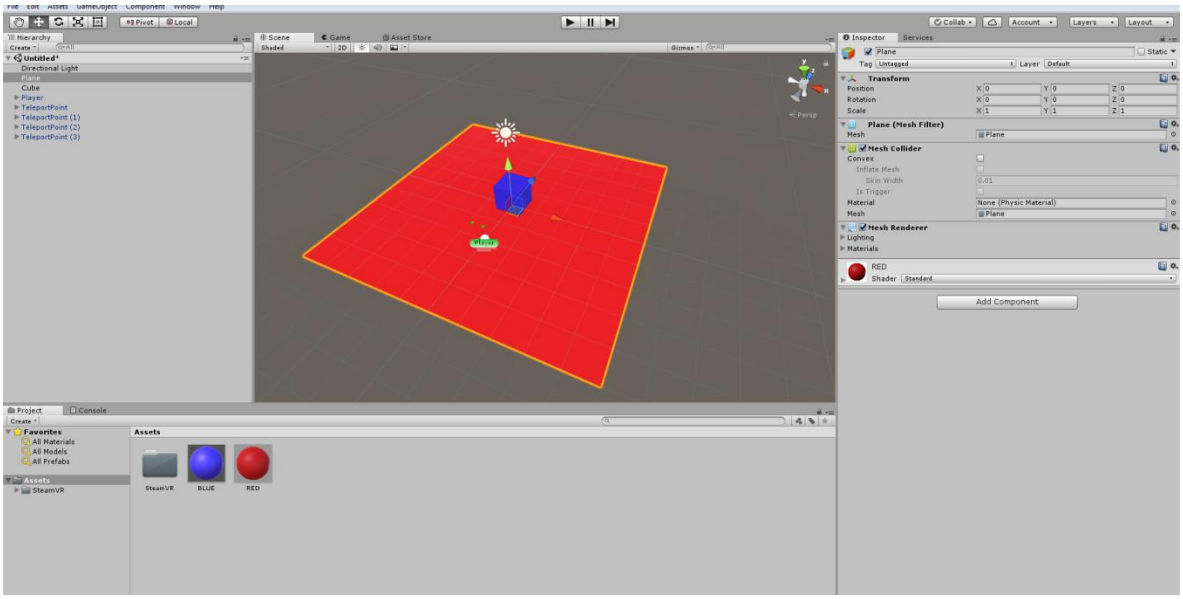

Illustration 3: Distributing the TeleportPoints in the corners

It seems difficult to see that the **Teleportation Points** are located in the **corners** but they are. If you get a closer look you see a **white dot** in all the **corners**. In my case it is because I do not have the SteamVR opened and the HTC Vive controls and headset connected, but in reality they are shown.

4. Our next step is creating a **Teleportation Area**, for this reason we are going to create another **Plane**. We are going to place it in the middle of our cube. Put these values into the inspector:
    - **Position.** X:0 Y:0 Z:0
    - **Rotation**. X:0 Y:0 Z:0
    - **Scale**. X:1 Y:1 Z:1

Make the **plane smaller** as in the illustration 4. Now, our second plane (the one that will allow us to teleport to) is centered with our original plane (the one the player is going to move, allowing the character to play).

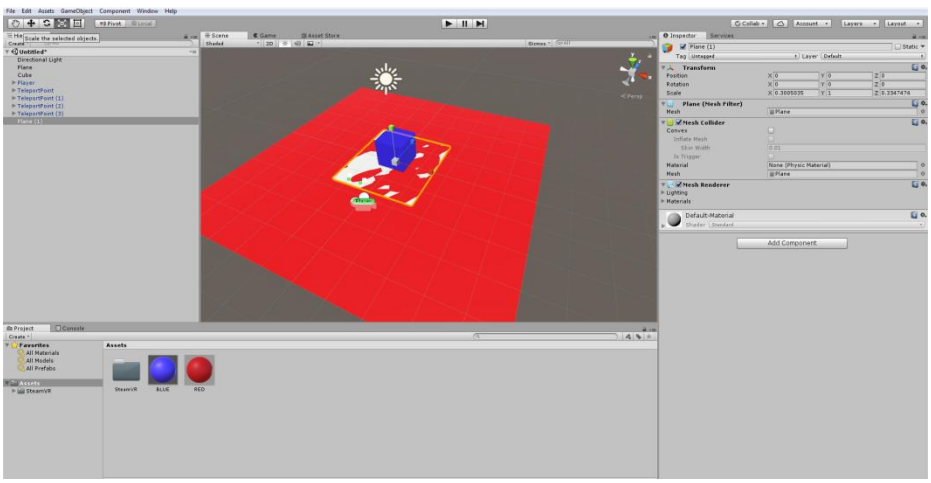

Illustration 4: Making the plane smaller

5. With this plane correctly located, we go to the inspector **Add Component > TeleportArea** (illustration 5). This script will allow the player to teleport to that **Plane**.

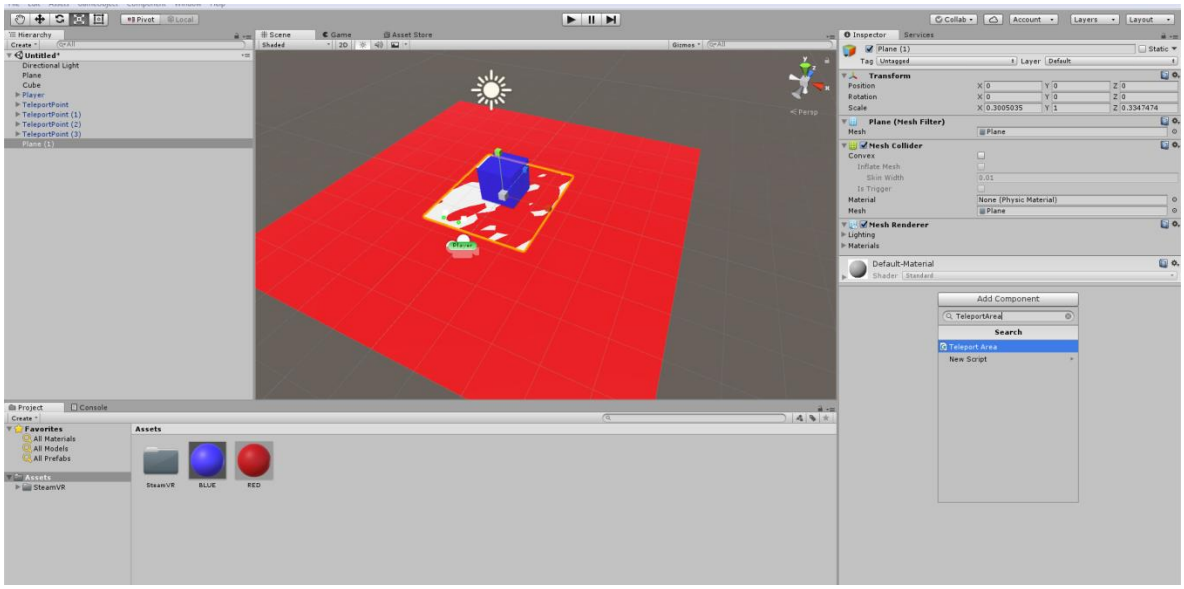

Illustration 5: Adding the TeleportArea to the plane

6. With all these created, we can now add new cubes which will be the ones that the player will throw. **Right click button** in the hierarchy, **Create > 3D Object > Cube**. Make it smaller and place it near the middle cube. Create a new material and colour it **yellow**.

7. With this new yellow cube, go to the inspector **Add Component > Interactable**. After that, **Add Component > Throwable**. These scripts will allow the player to **grab** and **throw** the cube (illustration 6). This scripts will allow the cube to have some **physics** used to know that the objects have gravity so that when you throw them will go down. Also, when they collide with other cubes the effect of this collision will be more realistic.

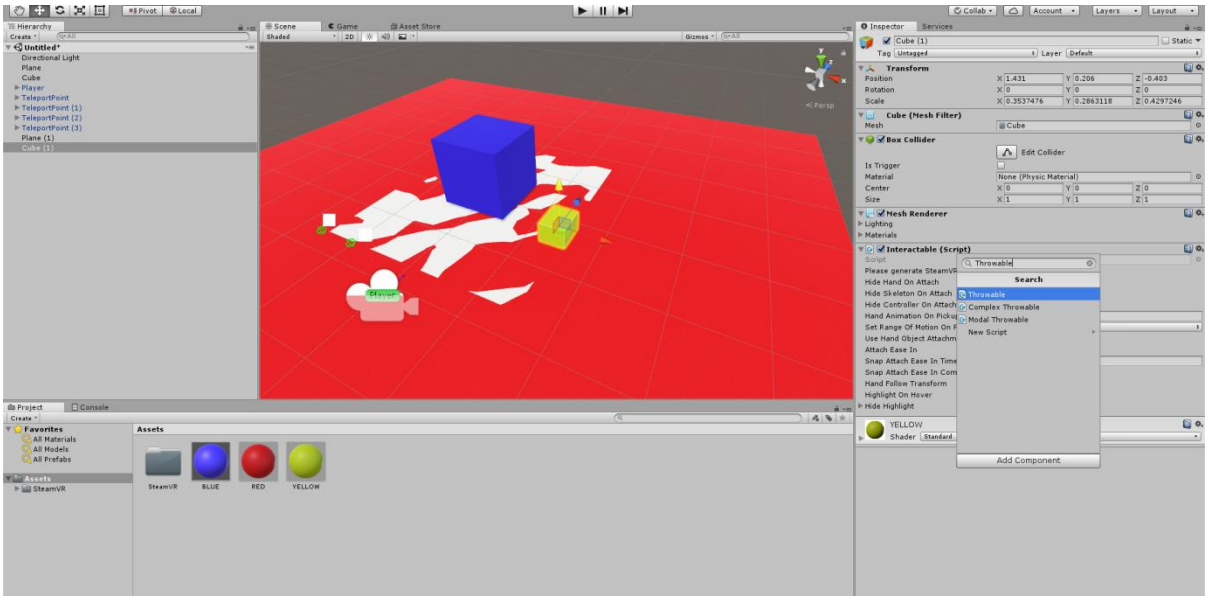

Illustration 6: Adding the Throwable script

8. We duplicate this cube, so that we can have more cubes, we can place all ones over them, we can make like a tower of cubes (illustration 7).

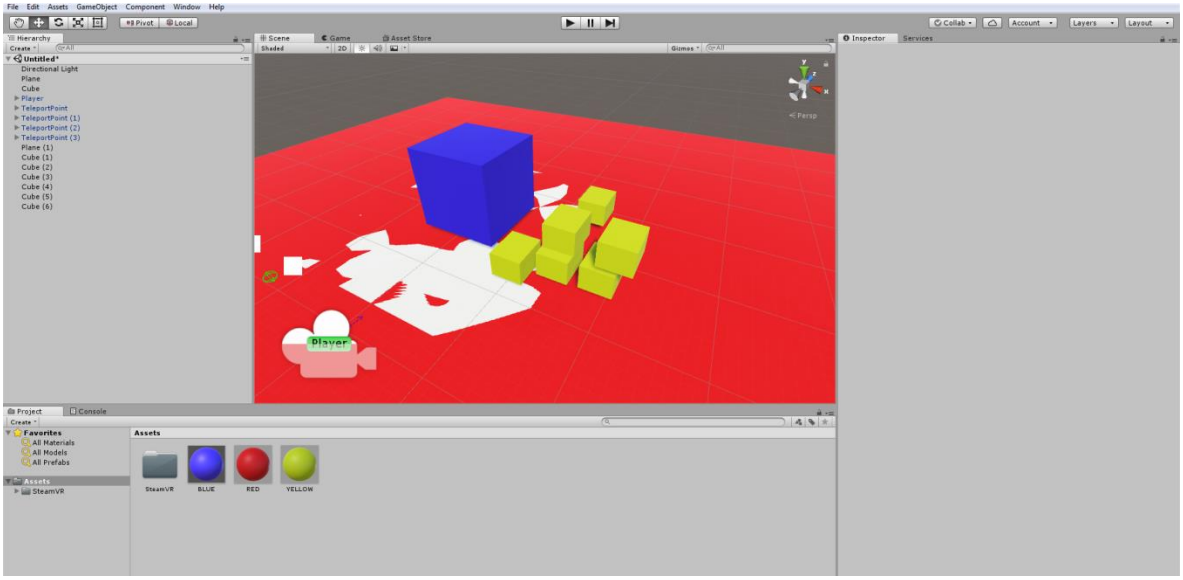

Illustration 7: Duplicating the cubes

Notice that if we want to reduce the size of the cubes we can go to "**Gizmos**" in the upper part and change a bit lower the **3D Icons**, by moving the bar it has.

9. Finally Save the Project and the Scenes. **File > Save Scenes**, name it **Interactions,** and **File > Save Project.**

We have finished, if you get to this part Congrats! You have implemented the game, now you will test it.

4. Testing the game

In this section we are going to explain how to test all the interactions we have added to our game. Therefore, first of all you need to pulse the "Play Button", take the Vive headset and controls and place yourself in the scene. First of all, you will have to move physically when you have the headset put on.

- For testing the **Teleportation points**, pulse the northern part of the trackpad and you will see that a laser pointer will be shown Right! Point with this laser to the corners of the plane, you will see that a point will be illuminated; release the trackpad button to notice the transportation.

- In order to test the **Teleportation Area**, pulse again the northern part of the trackpad by pointing to the cube that is in the middle of the plane, you will notice that the plane will be illuminated so; you can be transported there by releasing again the trackpad button.

- Finally, to test that we can **throw the objects**, we get closer to the yellow cubes, we grab the object by pulsing the trigger with the Vive and we throw it releasing the trigger. With the rest of the cubes we can do the same thing.

5. Final considerations

'*InteractionTutorial Game*' as you have noticed is a very simple game to create. It basically tries to introduce new users in **Unity** to gain some knowledge of how to throw objects and how to teleport. There are, of course, many things to be improved and much more original to be created as a result of the development of this game project.

6. Bibliography

Valem, 2018. *Unity HTC Vive Tutorial : How To Setup VR, Hand, Teleportation and Grabbing Object.* [En línea] Available at: https://www.youtube.com/watch?v=iJ0oNYIUFJo [Último acceso: 25 04 2019].