

3D User Interfaces

Unity Virtual Reality Prototype



Julius-Maximilians-Universität Würzburg

Human-Computer Interaction

Chair of Computer Science IX

A project report submitted by

Manuel Calvo Martín

Supervised by

Dr. Prof. Jean-Luc Lugin

Prof. Andrea Bartl

Prof. Ronja Heinrich

Prof. Chris Zimmerer

Summer Semester 2022

Index

1. Introduction.....	3
2. Prototype launch	3
3. User interactions.....	5
3.1. Help Menu.....	5
3.2. Smooth movement and rotation	6
3.3. Snap Turning.....	6
3.4. High Speed Acceleration Mode	7
3.5. Jumping Technique	7
3.6. FPS Menu	8
3.7. Object Factory.....	8
3.8. Object Modification.....	9
3.9. Oculus Menu.....	10

1. Introduction

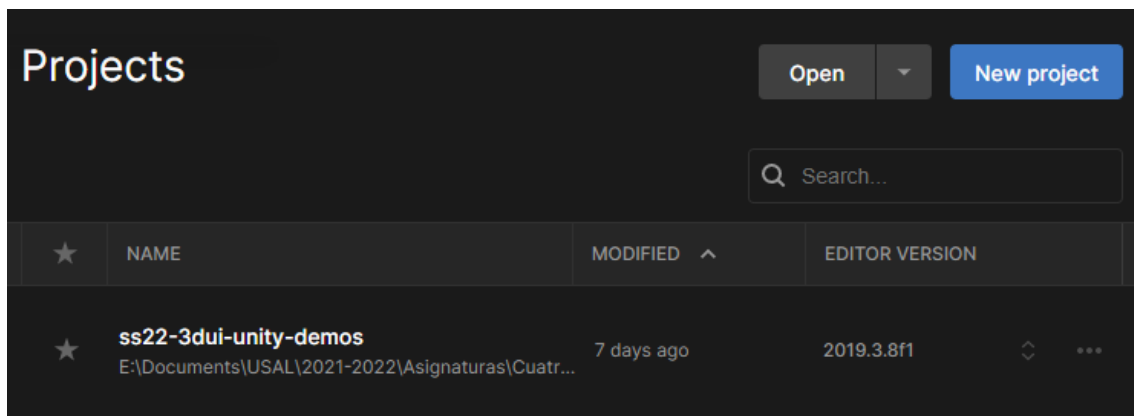
This project has been realized during the Summer Semester of 2022, under the module 3D User Interfaces taught by the Julius-Maximilians University of Würzburg.

It consists of a series of tasks and exercises presented in the lectures, in which a virtual reality prototype is developed.

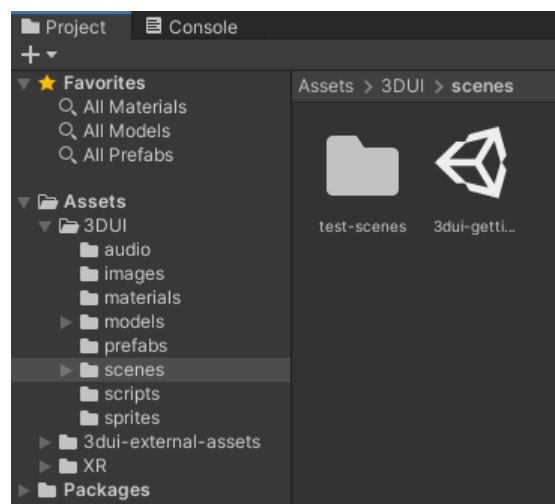
For its completion, the hardware used is an Oculus Rift S headset along with two Oculus Touch controllers. On the software side, Unity Engine 2019.3.8f1 and C# is used, in combination with a 3D Environment and Asset package already provided by the supervisors.

2. Prototype launch

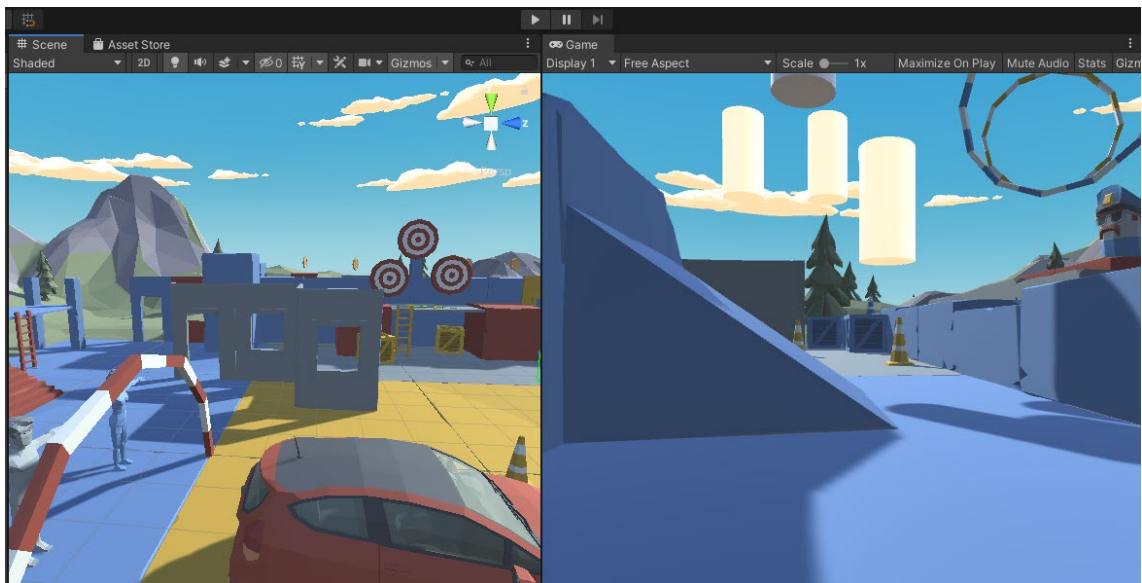
In order to launch the prototype, first the Oculus app needs to be executed. Once logged in, Unity Hub must be launched, and from there, open the folder containing the prototype.



Click on it and when it finishes loading, under the project tree, navigate to Assets, 3DUI, scenes, and double-click on the 3dui-getting-started.unity file. The scene should load.



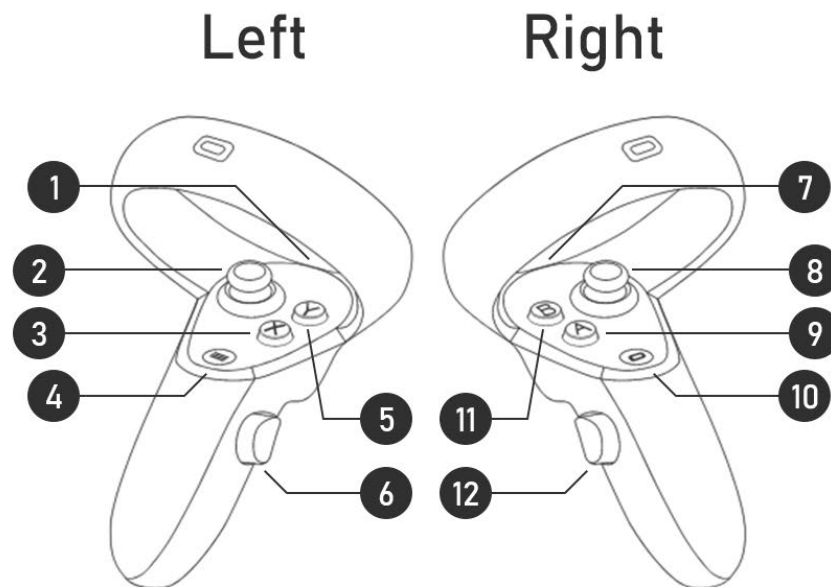
After this, the final step is to click the play button marked by the triangle on top of the window. In order to exit the simulation, it is necessary to click again the play button.



3. User interactions

3.1. Help Menu

The list of possible user interactions is shown in the help menu, which can be opened using the menu button (4) on the Oculus Touch left controller.



1 | Left Trigger
Toggle High Speed Mode

2 | Left Joystick
↑↓ Movement on hand pointing direction
↔ Turning

CLICK Toggle Snap Turning

3 | X Button
Open Object Factory Menu

4 | Menu Button
Open Help Menu

5 | Y Button
Open FPS Menu

6 | Left Grip
Jump to Red Indicator

7 | Right Trigger
Interact with GUI

8 | Right Joystick
↑↓ Move object along ray or rotate in Y-axis
↔ Scale object or rotate in X-axis

CLICK Toggle Rotation Mode

9 | A Button
Pick and drop object

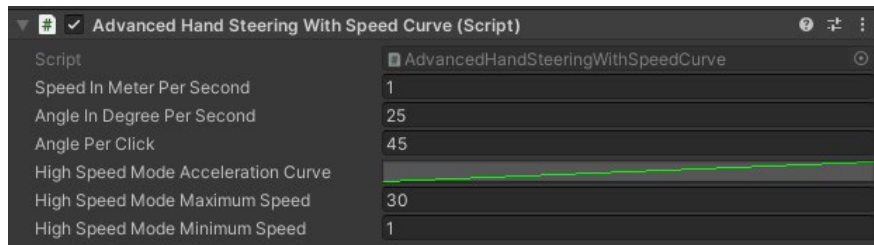
10 | Oculus Button
Exit to Oculus Menu

11 | B Button
Destroy picked object

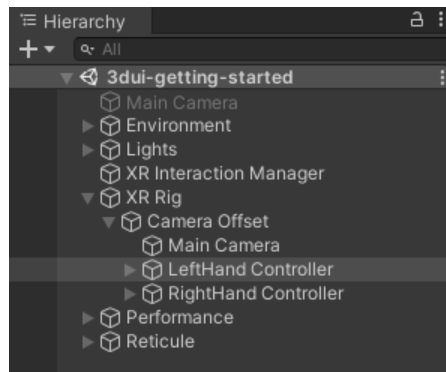
12 | Right Grip
Toggle gravity of object

3.2. Smooth movement and rotation

The user can move smoothly through the VR environment using the X and Y axis of the left controller's joystick (2). The speed and angle of rotation per second is fixed in this case, although there are public fields where to change those default values inside the Unity app.



In order to access those, under the scene hierarchy, we access the XR Rig, Camera Offset, LeftHand Controller section. When opened, a menu with all the attached components to the left-hand controller is shown. Inside this menu, all the default parameters can be modified as seen on the image above.



The user moves in the direction the left hand is pointing, virtually shown as a red ray coming out the left controller. This technique is called hand-steering.

3.3. Snap Turning

Snap Turning is an enhancement to hand-steering, as it allows the user to faster rotate on itself, without making any physical movement.

For the user to access Snap Turning mode, it must click the left controller joystick (2). In this mode, instead of turning smoothly at a fixed speed, the X axis of the left controller joystick is turned into a button. The user acknowledges its toggle by 2 vibrations of high and low intensity.

When the X axis passes the 75% pressure in one of the directions, it turns, per default, 45° in that direction once. A turn does not happen again until the joystick is back under the 75% pressure, locking it from constantly turning until that point.

While the user is in Snap Turning mode, the Y axis is still used for smooth hand-steering.

3.4. High Speed Acceleration Mode

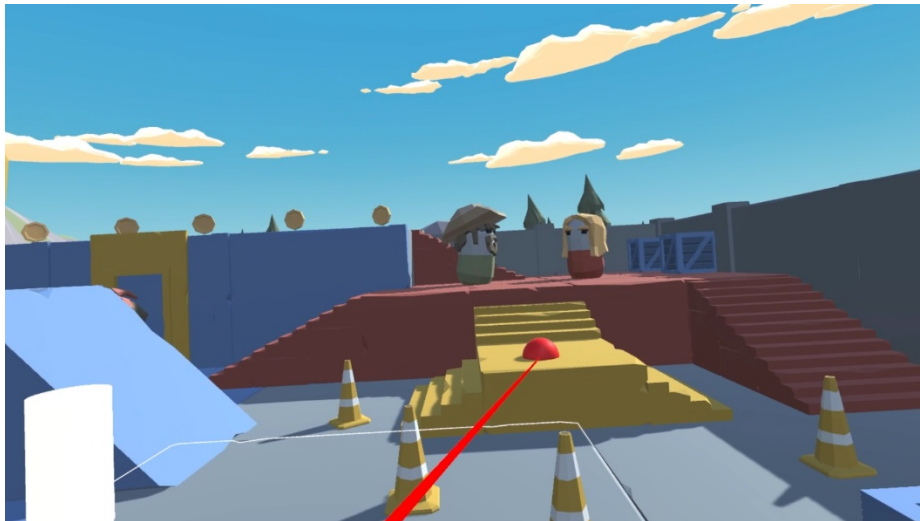
The user can toggle another enhancement of hand-steering, which implements dynamic speed. For it, the user must press the left controller trigger button (1). On toggle, the left controller vibrates three times.

The speed of the front-back movement then is accelerated, between two preset values, in the span of 5 seconds. The speed can also be controlled by the Y axis pressure.

3.5. Jumping Technique

Another navigation technique implemented in the prototype is Jumping.

In this technique, the user jumps to the location marked by a red indicator or reticle at the end of the pointing ray.

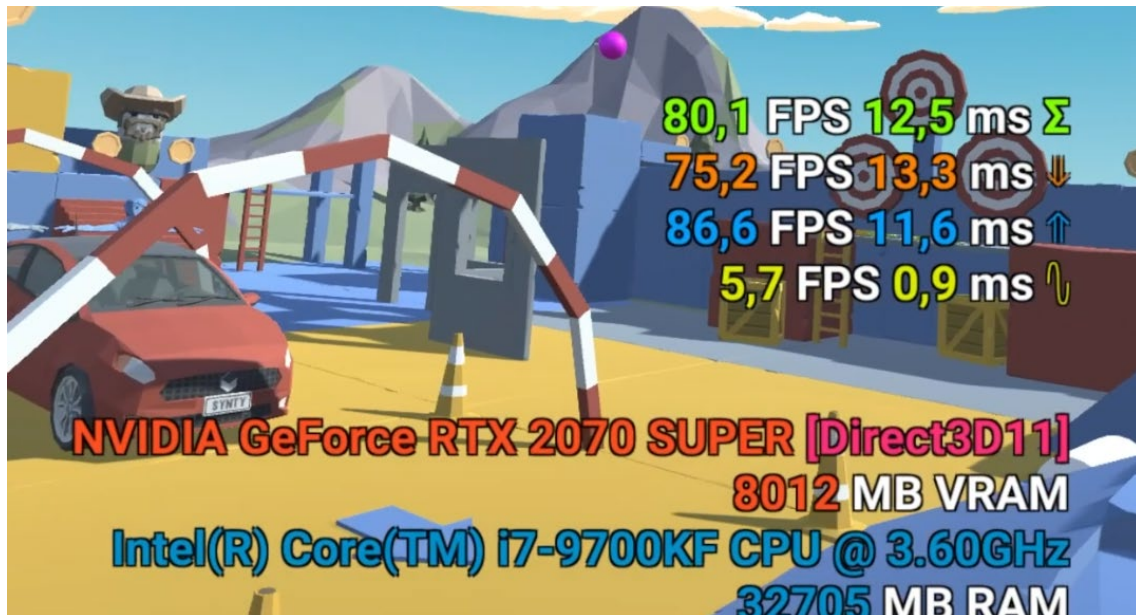


For the user to jump to this point, it must touch once the left controller grip button (6). After it has been touched, the vision fades to black for 0.3 seconds, moment in which the user is located on the desired location, and a violet indicator is placed on the original point of travel. Once the vision comes back, the user is on the desired point of travel looking in the same direction.



3.6. FPS Menu

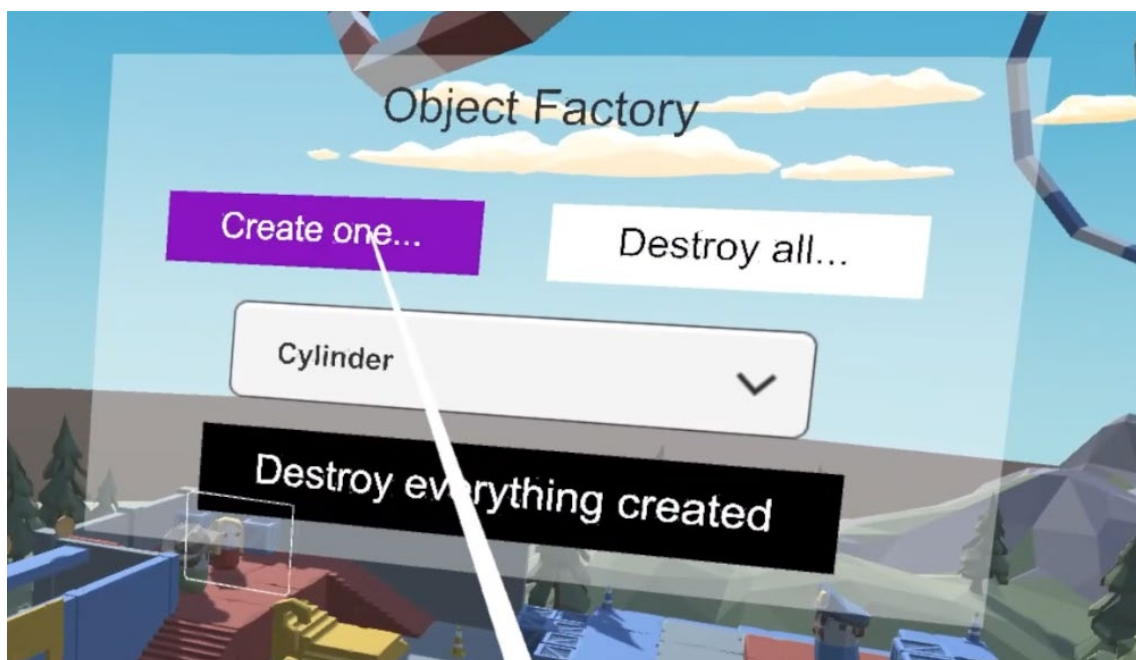
Using the Y button on the left controller (5), the user can display a performance menu in front of its vision, which will continuously follow where it is looking. It shows the variation in frames per second, as well as the delay in milliseconds and the computer characteristics.



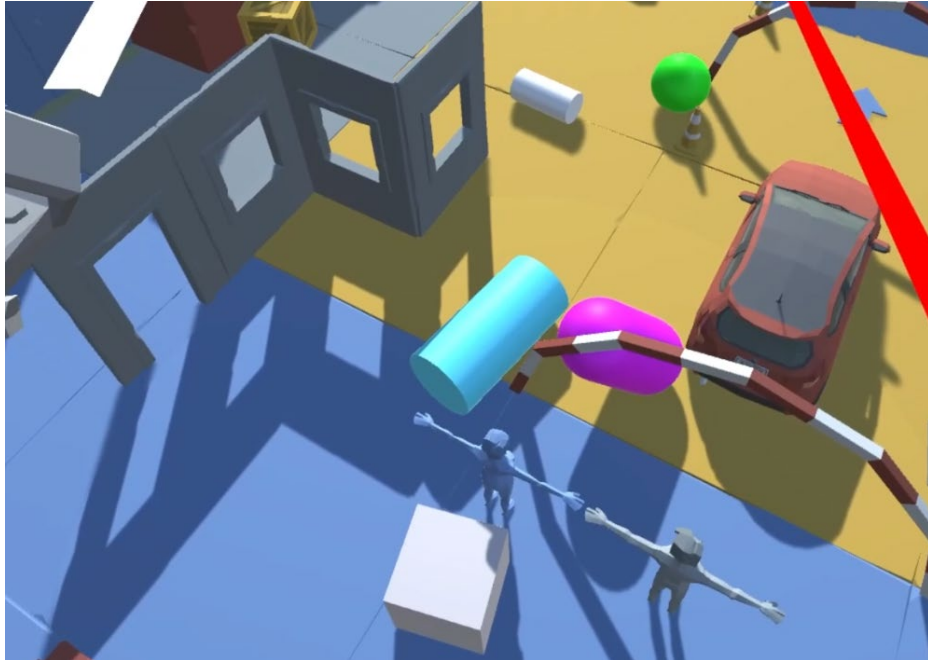
3.7. Object Factory

The user can open a menu in which to generate default prefabricated objects or prefabs. To do so, the X button on the left controller (3) must be pressed.

When opened, a 3D menu opens in front of the left controller ray location and stays static in that location until the X button is pressed again.



This menu has a series of buttons and a dropdown menu. The user can interact with them using the right controller trigger button (7). Under the dropdown menu, the user can find the four types of prefabs it can generate, cylinder, sphere, block and capsule.



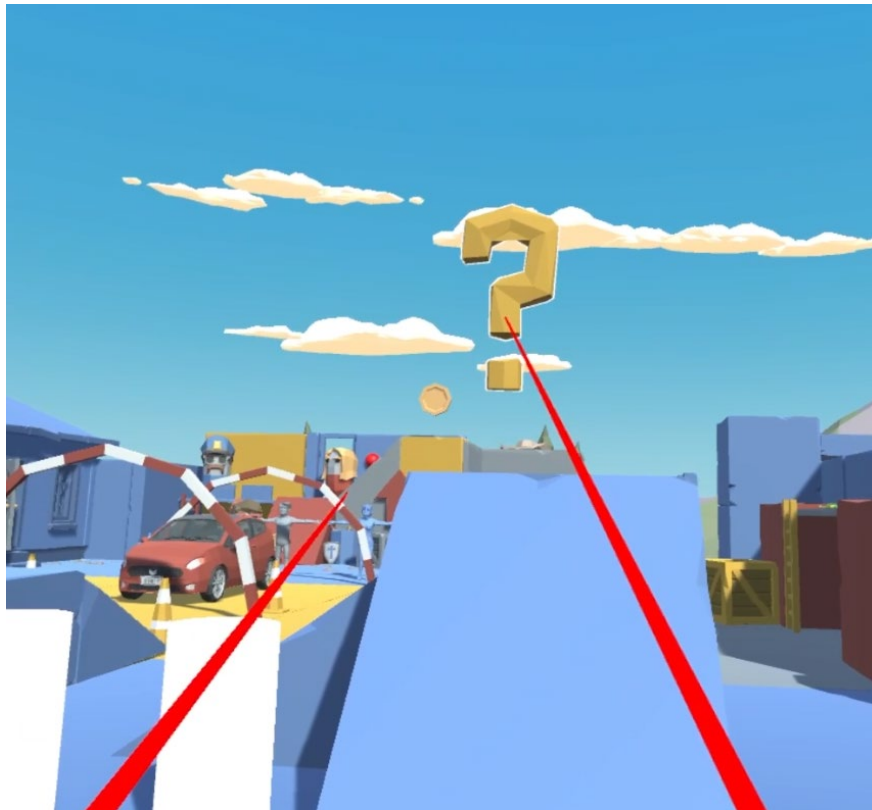
Once one of them is selected, the user can click on the create one or destroy all buttons. The create one button creates one prefab of the selected kind in the middle of the menu screen, with gravity. The destroy all button, however, deletes all prefabs created of the selected kind.

Finally, the destroy everything created button deletes all generated prefabs at once.

3.8. Object Modification

The user can modify and manipulate practically any element in the environment. To do so, the user must point at the object with the right-hand controller ray and click the A button on the right controller (9). If the object is modifiable, a white outline will appear around the object, and when selected, both a sound cue (click) and a low-high short-long vibration play.

In that moment, the object is picked up and the user can move it around with its ray. This technique is called ray-picking.



A picked-up object can be modified in various ways:

- The position can be changed along the ray axis using the right controller joystick (8) Y axis.
- The scale can be changed using the same joystick X axis.
- The object can be rotated in its X and Y axis. In order to do so, the user must toggle rotation mode, pressing the joystick. A fast click sound cue plays along with 2 short vibrations, marking its toggle.
- Rotation is controlled then by the joystick's axis. The user can return to the position/scale mode pressing the joystick again.
- An object can be deleted using the B button of the right controller (11).
- An object's gravity can be switched using the right controller grip button (12). When this happens, three short vibrations occur on the right controller.

For a user to drop an object, it must click again the A button on the right controller (9). A reverse click sound cue plays and a high-low long-short vibration occurs.

3.9. Oculus Menu

The user can access the Oculus menu using the Oculus button (10). From here, the user can see its surroundings as well as exit the prototype.