Module Code: CS3VR16

Assignment report Title: VR Individual World

Student Number: 26011251

Date (when the work completed): 28/01/21

Actual hrs spent for the assignment: 25

Assignment evaluation (3 key points):

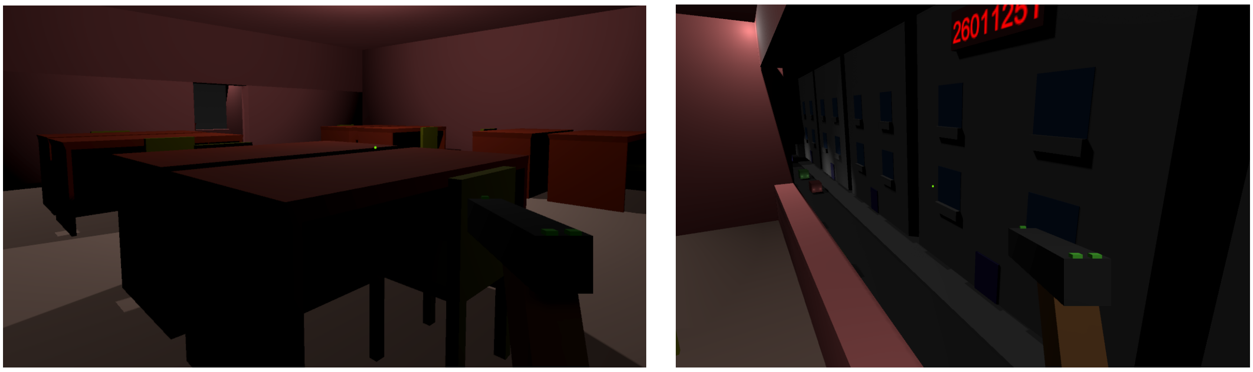
1. Develop skills using Unity and Blender
2. Think about simple solutions to problems
3. Think of how to combine several elements into one whole

Section 1. The World

The world constructed in Unity is an office building located within a downtown setting. The office comprises of two floors, lower and upper, with a fully functioning lift connecting the two. The player starts on the lower floor, taking the lift up to the top floor, to two office rooms containing desks and chairs. The far room has a window through which the player can view the downtown scene; cars follow each side of the road, lighting up the buildings with their headlights, before disappearing into tunnels at the opposite ends. The world makes use of two invisible walls to hem in the player; the first is on the lower floor to prevent the player from walking into the darkness, the second is on the upper floor to prevent the player from jumping out of the window onto the road.

The purpose of the world is for use as a level in a first person shooter game. The scene is set at night, with the use of lighting and colour intending to evoke the feeling of being in an important office building within a big city, perhaps raiding a gang hideout or hunting down a corrupt politician.

(Below: office with furniture and downtown view from window)



Section 2. Player Navigation

Controls:

WASD - Player movement

Space Bar - Jump

Mouse - Camera

LMB - Shoot Gun (Single Press or Hold)

E - Operate Lift (Move Lift)

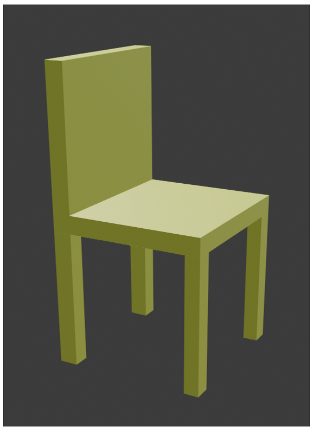
F - Operate Lift (Open/Close Doors)

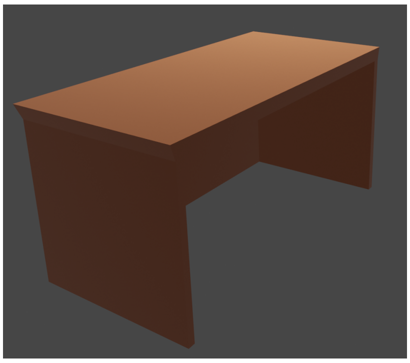
The features of player movement are as to be expected from a typical first person shooter game. The player can run, jump and fall, shoot their gun, and operate wall buttons. The lift actions are separated into two controls, one to open/close the lift doors and one to activate the lift once inside. This is partly because it was easier to code it this way, but also because games have a problem of assigning too many actions to the same key/button press (most being made for console controllers in mind); as the world is built for PC, there seems no reason not to make use of the extra keys.

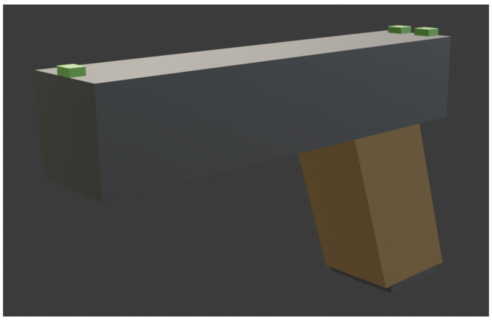
Section 3. Resources

All building objects within the world were constructed using simple cube objects found within the Unity builder, each manipulated into shape. The cars travelling on the outside of the building are also made of single cubes, mimicking the practice of using low resolution objects, found within the background of game scenes, to lower rendering requirements.

The models used for the chairs, desks, and the gun were all made by myself within Blender. I thought it would be more beneficial to learn how to make my own rather than download pre-made assets. Though simple, they convey the objects well and feature suitably within the world. My student number can also be found on a billboard on the side of a building, viewable from the window on the top floor. (Below: chair, desk, and gun)







Section 4. Scripting

The world makes use of five different C# scripts, three for the player (including one for the lift operation), one for the gun, and one for each direction of car.

1. Player

MouseLook

The MouseLook script locks the mouse cursor to the centre of the screen, allowing the player to pan the camera horizontally, as well as vertically within an arc of 180° (between 90° and -90° from the normal). The script makes use of the player's transform position as well as the x and y positions of the mouse.

PlayerMovement

The PlayerMovement script manages the movement of the player forwards, backwards, left, and right. It also allows the player to jump and fall with acceleration according to gravity (though there aren't any great heights to leap from within the world so this feature isn't very dramatic). The script makes use of a player character controller, as well as a transform to check whether the player is on the ground (for use in calculating the player's downwards velocity).

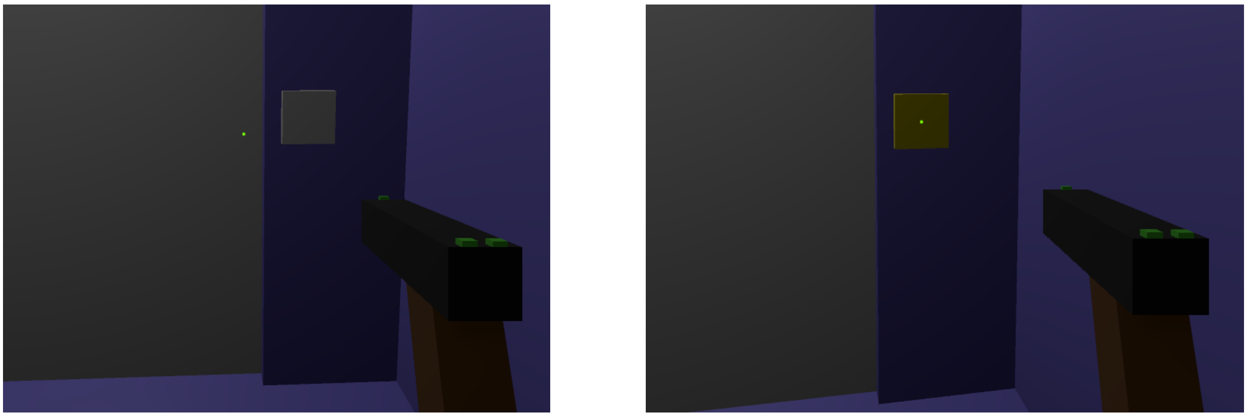
PlayerInteraction

The PlayerInteraction script is the most complicated and manages the player's interaction with the lift. Ray casts are used to determine whether the wall/lift buttons have been targeted up close by the player's central crosshair. The buttons will highlight yellow if interactive, remaining grey and inoperable when the lift is in transit. The transform of the lift doors and the lift itself determine whether the doors will be opened/closed, and which way the lift will move once activated.

The lift doors can be opened on the outside of or within the lift itself. The player can repeatedly open and close the doors, alternately they will automatically close after time if left open or the player activates the lift movement. The doors will automatically open upon arriving at the next floor. The player cannot change the direction of the lift while in transit, nor can the doors be opened.

Writing the script for the lift was the most time consuming element of the world's creation, with time taken to ensure its actions and behaviour replicated that of a real, working lift.

(Below: unselected and selected lift button)



2. Gun

The Gun object acts as the 'gamey' element of the world, (unless riding up and down a lift is to be considered more fun). The gun model follows just in front of the player, shown on a separate camera to the rest of the world, so it doesn't clip into objects or the wall. The script manages the ray cast targeting, making use of small, central crosshair. Upon clicking the LMB the muzzle flash particle system will play at the barrel of the gun and a spark/debris particle effect will appear at the end of the ray. The gun can be fired singularly or continuously through holding down the LMB.

3. Cars (Left and Right)

The simple car scripts move the cars along the road according to their direction and respawn them at the opposite ends upon reaching the tunnels, allowing for infinite 'traffic' within the downtown background, viewable from the office window.

Section 5. Reflection

I believe I have achieved the goal of creating a believable and consistent world in which the player can easily navigate and interact with the game objects. Novelty is shown through the implementation of a working lift and first person shooter ‘game elements’.

I have spent nearly double the amount of expected time on the project, partly because of wanting to achieve a high mark for third year, but also so I can learn more about using Unity and Blender. Coincidentally, my final year project is also based in Unity. This coursework has allowed me improve my understanding of Unity and Blender, the latter most important as it will allow me to create more professional looking models for use in my project.