Virtual Reality Assignment 2 Report

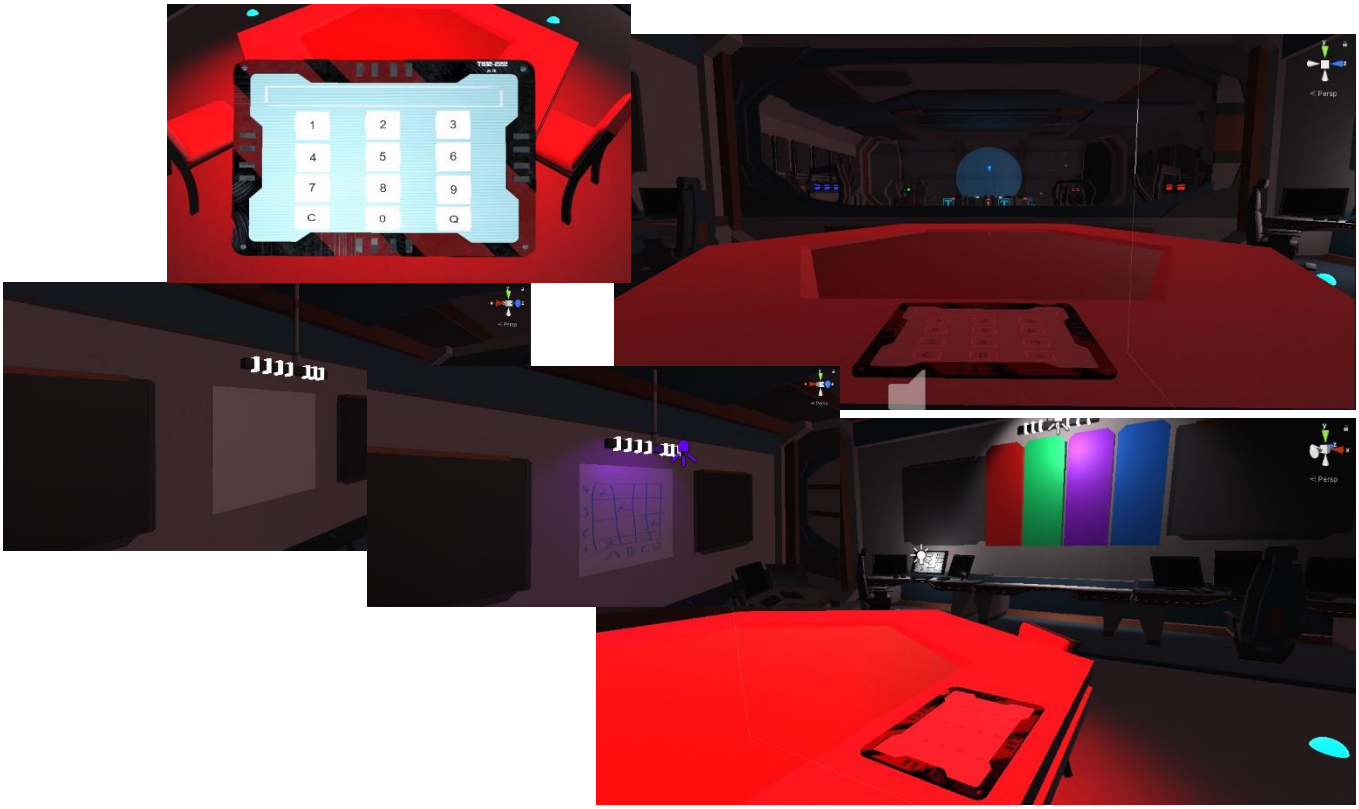
About the game

The game I have designed it a puzzle based sci-fi VR experience. The user loads into a spaceship bunker with the first objective of leaving the bunker and navigating towards the upper deck of the ship where a final room contains a puzzle to stop the ship from failing. There are 2 major puzzles in the game, 1 is to open the last door into the final room and the 2nd puzzle is to finish the game.

Puzzles

1st Major Puzzle

The first puzzle uses a UI keypad that can be selected as a prefab game object in one of the rooms, the keypad requires a 4 number pass code to open the last door. The passcode can be found by revealing a point graph, this is done by picking up a bulb earlier on in the game and placing this in the light fixture above the point graph which is hidden. After revealing the picture on the wall, the user can then find a grid on one of the computer screens in the room where they will find numbers relating to the points on the graph. Finally, to find the pass code the user can see on the wall a set of coloured panels which show the order of which the numbers are placed on the keypad.

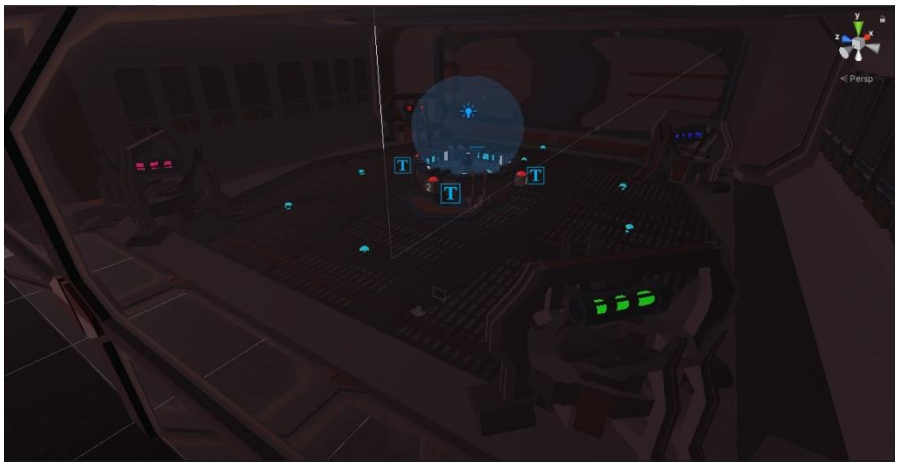


The above screenshots show the process of which the user can access the next room.

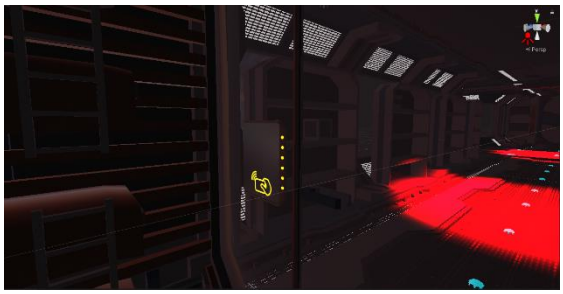
2nd Major Puzzle

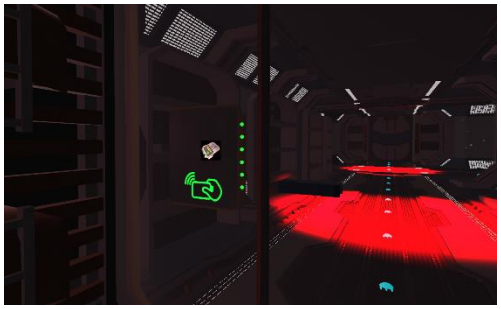
The 2nd puzzle is in the final room of the game where the player is met by 4 spinning batteries in each corner of the room, the player should take note of this as in the centre of the room is a console surrounded by 4 buttons. Each of these buttons can be pressed by the player to finish the game.

When the player presses each button the colour of the button changes, when all of the colours are in the right order via the number they have associated with them, the player will have completed the game. The number on each button relates to the speed of which the batteries are spinning in the room. 1 being the fastest and so on.



The other puzzle, though just simply picking up an id card and scanning through to the corridor is featured in the first room of the game.





Here I have cast out a ray from the player which interacts with objects on different layers.

The layers and interactions are as follows:

- Teleport nodes: Moves the player to the node selected around the map
- Access Module: When colour is green the player can click to open the door of which the module is next to
- UV Light Fitting: Turns on the light revealing the hint for the passcode
- Final room console buttons: Change colour every time the player clicks, when all are the correct colour the game ends

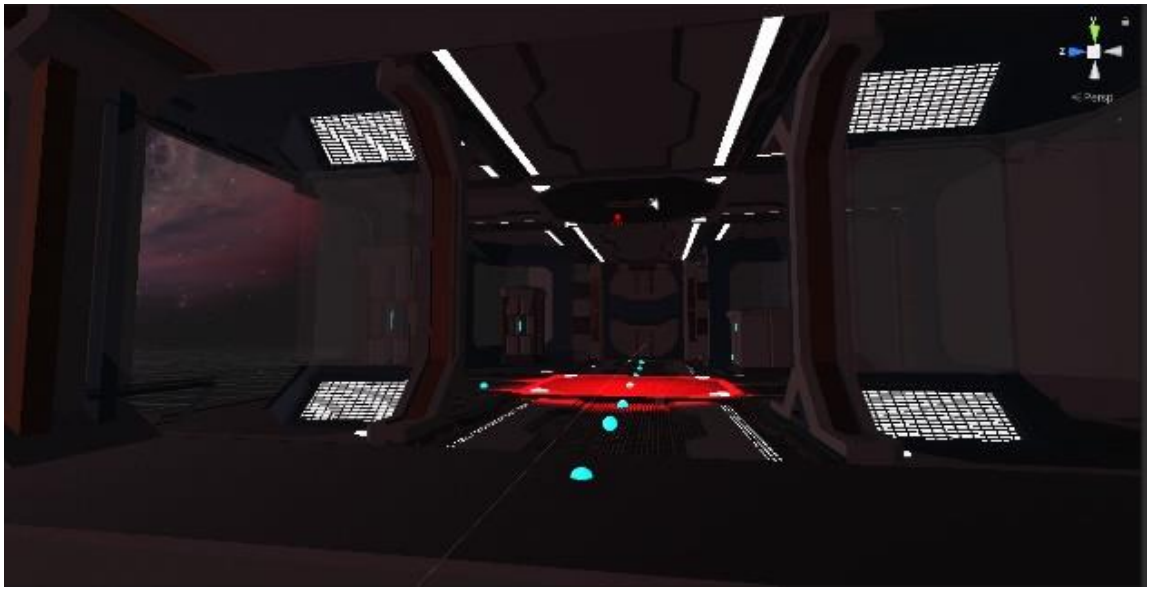
Design of the game

The design of the game comes from a package found on the asset store which gave me a bunch of space looking walls and doors to use. The keypad I have used is from a tutorial found online however I have edited the design to incorporate a tablet style design that I found on the asset store.

Upon awakening in the level, the ship features 2 main indicators that the player is in danger and needs to move fast. One of which is the lighting of the level, until the user finishes the game, red flashing lighting appears from the ceiling. The other indicator for this a background sound effect.

Travel

For travelling through the game, I decided to use a teleportation system white teleport nodes throughout the ship. One challenge I did face was navigating on the upper levels of the ship and the lower levels, for this I had to make sure that the Y axis value was set to the correct height depending on which level the player was teleporting to.



To achieve the affect of moving between the levels I wrote the following function

```
else if (Physics.Raycast(ray, out hit, 10f, LayerMask.GetMask("TeleportSpawn")))
{
    hitTarget = hit.collider.gameObject;
    if (hitTarget != currentTarget)
    {
        Unhighlight();
        Highlight(hitTarget);
    }
    if (Input.GetButtonDown("Fire1") && cursPos.equipRet == true && hitTarget.CompareTag("GoUpstairs"))
    {
        player.transform.position = new Vector3(upPoint.transform.position.x, 14.0f, upPoint.transform.position.z);
    }
    else if (Input.GetButtonDown("Fire1") && cursPos.equipRet == true && hitTarget.CompareTag("GoDownstairs"))
    {
        transform.position = new Vector3(downPoint.transform.position.x, 4.0f, downPoint.transform.position.z);
    }
    else if (Input.GetButtonDown("Fire1") && cursPos.equipRet == true && hitTarget.CompareTag("UpStairs"))
    {
        transform.position = new Vector3(hitTarget.transform.position.x, 14.0f, hitTarget.transform.position.z);
    }
    else if (Input.GetButtonDown("Fire1") && cursPos.equipRet == true)
    {
        transform.position = new Vector3(hitTarget.transform.position.x, 4.0f, hitTarget.transform.position.z);
    }
}
```

In the function above I have specified the height of the player depending on which level they are travelling on.

Another challenge regarding the movement of the player was ensuring that the player could not teleport through doors. To counter this I set all doors layers to “Door” and check for this in the script to make sure the player could never cheat and skip ahead without completing one of the tasks.

```
if (currentTarget != null)
{
    Unhighlight();
}
else if (Physics.Raycast(ray, out hit, 10f, LayerMask.GetMask("Door")))
{

}
else if (Physics.Raycast(ray, out hit, 10f, LayerMask.GetMask("TeleportSpawn")))
{
```

Inventory

One of the key aspects to the game is the inventory system that allows plays to travel around the space with the reticule cursor and objects such as the access card and light bulb. Moving through the inventory was a challenge as it would change throughout the scene. With the bulb and access cards potentially being added and removed through, for this I decided to use a range of if statements to change the inventory item on a button press. As I wanted to spend time focussing on the puzzle aspects of the game I didn't want to spend too much time working out a data structure, although with further development to this game, using a linked list would be an ideal method allowing me to iterate through the list when an item needs to be selected.

Menu

The menu is a basic design with some animations and VR world interaction. The player floats in space while a ship moves around in the air. The menu which comprises of a button to start the game floats in the air, the user can move their head to see the skybox that surrounds the area.

Performance

Some attention was paid towards performance by ensuring that all audio files were not played too high. Some sounds in the game were increased based off relative position of the user.

One slight issue with the performance of the game would be the travel system, as the user moves along the ground without actually moving in the real world there could be some motion sickness related issues with this method. It was however decided to go with the teleport method as it would make clue gathering easier with the user fixed to a certain position when they move.