VR Assignment 1

Contents

Introduction

In order to take full advantage of the VR environment, I want my game to be set in an environment with movements that users cannot make in non-virtual life. So, my game will be set in space where the player is able to walk completely around planet-like spheres. There should also be an element of danger that the player has to compete against and a goal to work towards.

Premise

You are a robot security guard in charge of guarding a system of ten spherical satellites. Magnetic booster boots allow you to jump from sphere to sphere and stick to them. Asteroids are constantly floating towards the satellites and it is your job to identify them and shoot them in order to stop them from colliding with the satellites. If half of these satellites are destroyed, you have failed your mission, and hence the game ends.

Environment

There are ten satellites, each within 30 units of their neighbour. Figure 1 contains a preliminary layout of what this system could look like.

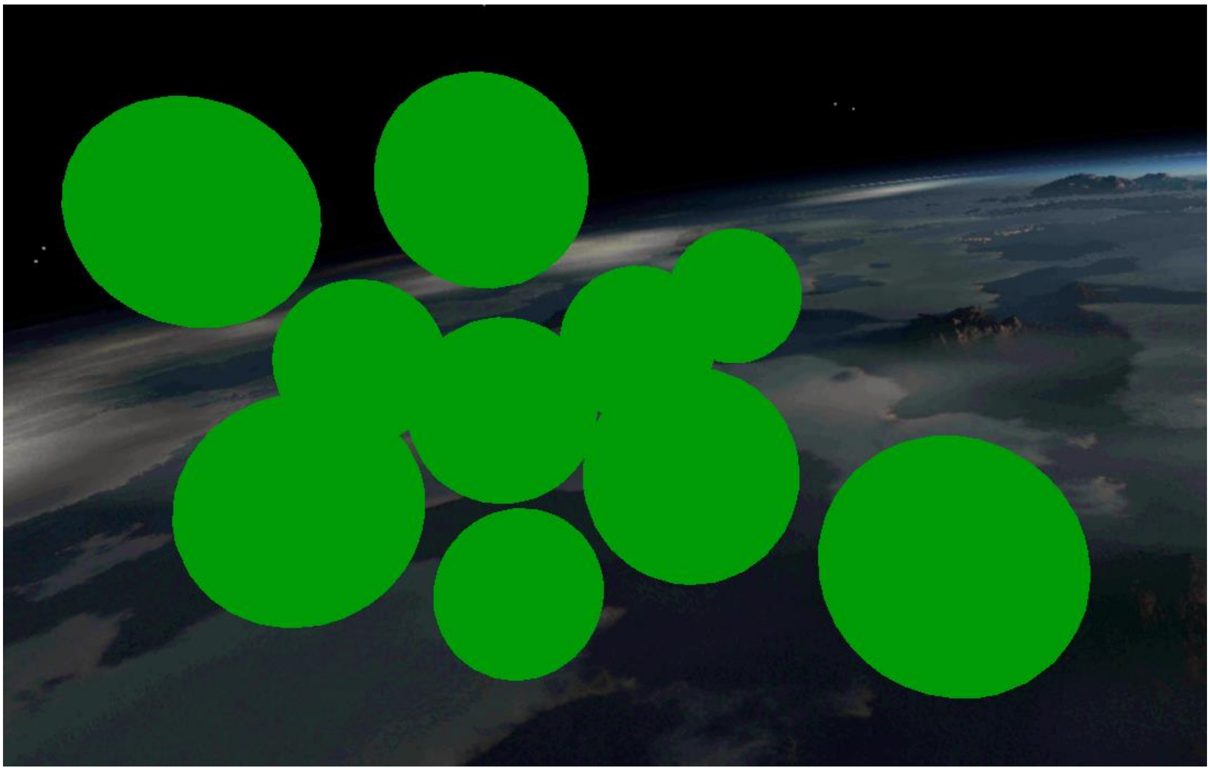

Figure 1 The system of satellites that the players inhabits

This entire system is rotating and floating up and down constantly to mimic an orbiting object in zero gravity. Each individual sphere will also be moving similarly within the system. The backdrop of the game will be a skybox with a view of planet Earth and a far-off sun.

Audio

There will be appropriate audio playing to signify certain events taking place. This will include the sound of footsteps on metal when the player moves, a propulsion sound when the player jumps from one sphere to another, explosions when asteroids collide with the spheres, and a laser sound when an asteroid is shot.

A looping track will also constantly be playing, something a bit spooky to set the ambience of the scene.

All of the necessary sounds are available as free assets on the Unity Store.

Goal

The goal of the game is to protect the satellites by shooting asteroids before they collide with the spheres. Once half of the spheres are destroyed, the game ends. The longer the player is able to protect the system, the higher the score achieved. As the highest score will be recorded, the player may aim to get a new high score upon each play.

Each satellite has a health of five points, losing one point each time it is hit by an asteroid.

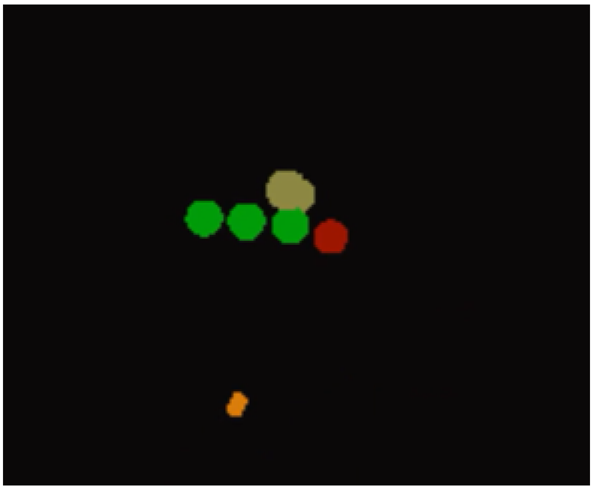

Figure 2 Minimap showing health of satellites and an approaching asteroid

The player will have access to a minimap that will represent the health of each satellite based on its colour, ranging from green (full health) to red (one health point remaining), as seen in Figure 2. Asteroids will also appear as orange capsules on the minimap so that the player can shoot them down before they collide with their targets.

Character

Mixamo.com has a downloadable character called ‘Exo Gray’ (Figure 3) that resembles a futuristic astronaut.

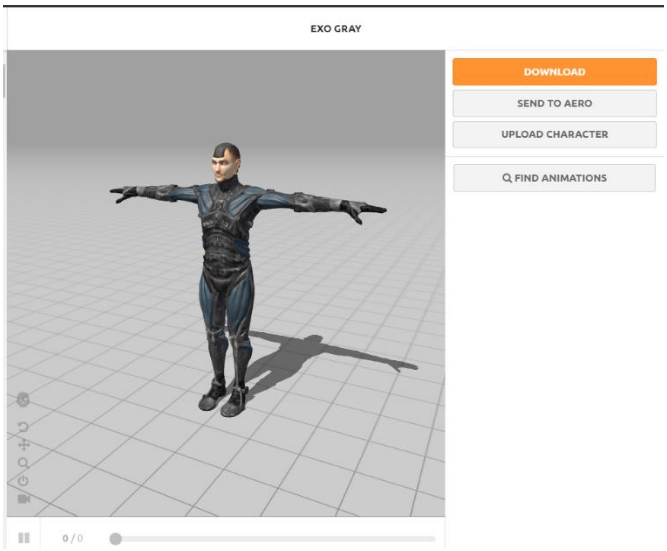

Figure 3 The character 'Exo Gray' on Mixamo.com

Using this character will also allow animations to be downloaded in order to make movement look more natural. The character’s head can be shrunk down to appear missing in order to avoid conflict with the VR camera.

A crosshair is necessary so that the player can aim at asteroids and see which satellite he is jumping to. This crosshair will be projected forward using raycasting so that it is not right in front of the player’s face, which would cause double vision in the VR environment.

Movement and Controls

Forward, left, right, and backwards movement will all be achieved using the left joystick on the controller, accompanied by appropriate animations.

Jumping will be tied to the ‘A’ button and the Unity ‘Lerp’ and ‘Slerp’ functions can be used to make positional changes and rotation smoother.

Shooting will be tied to the ‘B’ button, and the minimap will be visible while the ‘X’ button is held down. Controls will be restricted to the joysticks and letter buttons on the right of the controller as preliminary tests show that some other button configurations sometimes do not port over exactly from the Unity development environment to the corresponding Android builds.

High Score

Scores are calculated according to the time spent protecting the satellites. The highest score of all games is saved in a text file. On Android, this text file can be saved in a phone’s persistent storage using ‘Application.persistentDataPath’. If a new high score is not achieved, a player’s current score is displayed (but not saved to persistent storage) alongside the current high score.

Minimap

The minimap will allow the player to detect oncoming asteroids as well as to check the health of the satellites. The minimap will only be shown while the ‘X’ button is pressed down as the mobile VR environment does not provide much room to show extra information on the screen that is easily viewable by the player.

Minimaps can be made in Unity by attaching a camera to the player, looking down from a few units above. This camera can be set to only view objects on the custom ‘Minimap’ layer. The image seen by this camera is then output to a UI image.

This means that each object to be viewed on the minimap, namely satellites and asteroids, need to be duplicated in the exact same positions and these duplicates must be set to exist on the ‘Minimap’ layer, with the appropriate materials in order to have the right colours for the minimap.