COMP30540 “Game Development” 2nd Game Programming Assignment “Drone Destroyer” Report

XUANHUI XU

1. Introduction

The game is called “Drone Destroyer”. The background is on an aircraft carrier. The player will be able to control a drone to defence other drones coming from a mystery space, protect those air planes from being destroyed by them. At the very beginning, the enemies are just sending some reconnaissance drone to get some information. After they finish gathering all the data they need, what will they do next?

2. Implementation

2.1 Controller

The drone can move by using keyboard keys: W for speed up, S for slow down, AD for moving left and right, Z for going up and X for going down. Meanwhile, player can use mouse to control the look direction of drone and fire a missile by clicking the left mouse button. The controller is modified from the standard unity asset, which saved me a lot of time. The missile is created as a prefab with an initial speed in the game, the firing direction is along the camera direction. As the rigid body component is attached on the model, the gravity will automatically affect the moving curve. Some box colliders are attached around the drone model as well. The reasons why not use one single collider are to prevent the missile collide with drone and make sure the drone can land on the ground without burying the camera. The drone has health script attached to record the health and trigger the event related to it, and the script listens to the collision event of enemy’s bullet to decrease the health if necessary.

The enemy is using almost same prefab as the player drone with same model but some other script. The AI behind the enemy is achieved simply: shoot the missile to the facing direction with random time offset; move randomly within a limited region; use a trigger sphere collider to check whether the player is coming close and look at the player if so. Enemy is designed have only one HP. As long as they get shot by the player, they will lose power and fall to destroy. For each level, the enemy prefabs are modified as a variant. Some parameters are changed in each case. For instance, the boss enemy (2.1.2) in the last level is unbeatable unless you push it out side the map and the size of it is 20 times bigger than the ordinary one.

2.1.1 Planes

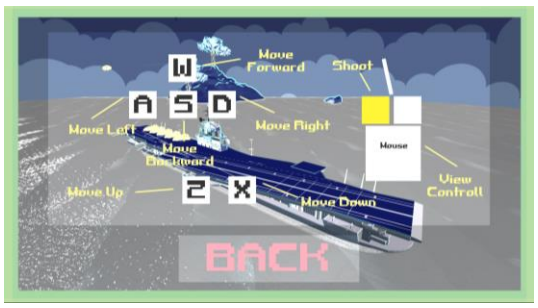
2.1.2 Boss Enemy

The enemies are generated in a random space within a sphere area, controlled by a script. Moreover, there are some plane generator appearing in the game to create flying or sliding plane (2.1.1). The plane flying in the sky has a power system as it is a prefab in the standard unity asset. By placing the generation point at a proper space, the plane would be able to land on the aircraft carrier. The other type of plane does not have the power system. All the behaviours including slide and take off are all finished by adding animations to the model, which also performs really well.

2.2 User Interface (UI)

The start scene (2.2.1) is quite symple, only having two buttons. One is to start the game, while the other is the instruction page. The instruction page (2.2.2) shows player the basic control method of the game and has a button to go back to the main page. What's more, those page use the actual game scene to decorate the background. Because of this, player would able to know the game background before they play the game, increasing the immersive feeling.

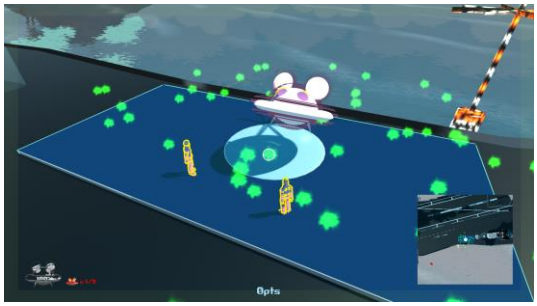
2.2.1 Start Scene



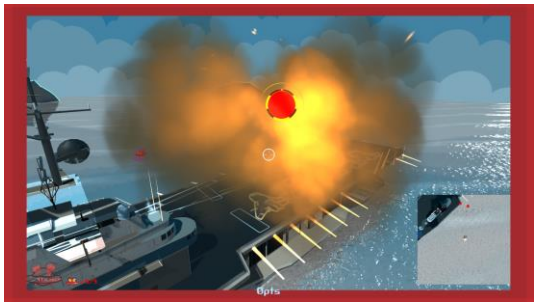
2.2.2 Instruction Scene

After starting the game, player can see the drone landing on the repairing station (2.2.3) and ready to take off. In the main canvas, the health of the drone is showed as a logo on the left bottom of the screen with the health percentage. When the health drops down, the colour of the logo will go red and dark to notice the player, they don not have enough health to keep fighting. Beside the health logo, the number of enemies is showed as well. Moreover, a mini map that can show enemies location is attached on the right bottom corner. By using a camera equipped by the drone, the view of the map is real-time and following the player.

When player is hit by enemy's bullet, the transparent frame will turn to red to warn the player that the drone is taking damage (2.2.4). Meanwhile, the camera should shake but the shaking effect is not that visible in this case. Player is able to repair the drone by driving it back to the take off point, which is known as repair station. The green particles tell the player that the drone is under repairing.



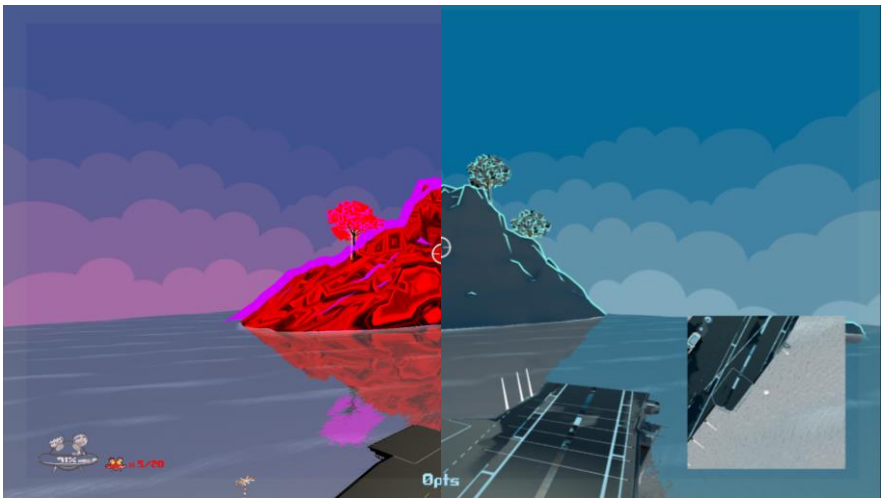
2.2.3 Repair Station



2.2.4 Taking damage

2.3 Artwork and Audio

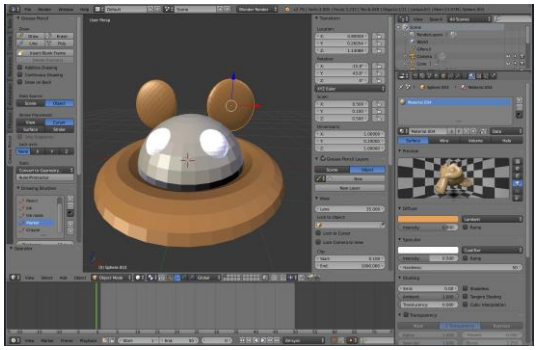
The main style of the game is kind of cartoon. By using the outline shaders with different colour, texture and outline width, the feeling of the whole game becomes much more vivid and interesting. In different level, I changed the basic tune of the colour to render different emotion of the gaming background (2.3.1). Comparing to the blue skybox, the red one shows more intensive and urgency, which force player to focus and play well.



2.3.1 Same Location with Different Style

There are lots of little details of the game that improved the immersive feeling of the game. The little audio for shooting, explosion and flying; The explosion particle showing after the drone is destroyed; The little portal that transfer the enemy into the battle ground.

2.3.2 Bullet Thread



2.3.3 Blender

The thread of the bullet (2.3.2) is not rendered by the trial render component as the TA said it should not be used. In order to achieve the thread, I have tried to use different joint component. However, the result of using joint is not ideal and wastes a lot of time. In the end, I decided to use initialise function to generate small spheres along the trail, which works really well.

The only model that I made by blender is the drone model (2.3.3). And I also made this character as the main logo of this game.

The audio source in the game are mostly 3D stereo, which creates a sound immersive environment.

3. Future work

There are lots of features that I would like to add to the game, but due to the time limit, I did not finish those features. So, let's add them into the future work.

The very first thing is adding health volume to the aircraft carrier ship and those planes. During the process of destroying enemies, the player should also consider the health level of the base, as the dead of the base will cause the game over as well.

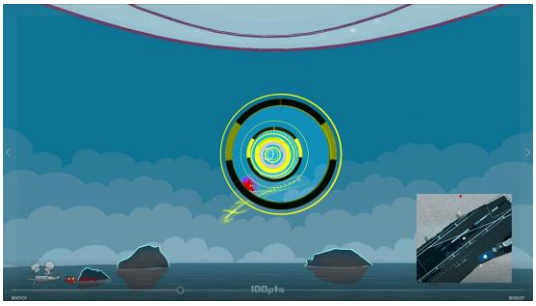
Then I would like to enhance the boss system with health as well. Because until now, the boss is unstoppable, unless you push it out of the map edge, which is quite challenging. When the player's drone enters the enemy's view sight, there would be a symbol to show that it notices player.

In the end, there would be a full background story, presented by the text or animation.

4. Screen shots

4.1 Win Page

4.2 Lose Page



4.3 Shoot

4.4 Plane Landing

4.5 Hit Enemy