COMP30540 – Game Development

Product Name: SpaceBall PaintBall

Version: v3.0

Student: James Dorrian

BSc. (Hons.) in Computer Science

Lecturer: Arthur Cater

UCD School of Computer Science

University College Dublin

Completion: May 2017

Student Number: 13369451

Gameplay Video Link: https://www.youtube.com/watch?v=_pbaO1zus6I

1 Introduction

The game involved 2 players in space. Each player is equipped with a paintball gun and must shoot at each other. Shooting the spaceship/colliding with the spaceship at speed/taking too much damage ends the game. To win the game you must either kill the enemy (2 paintball hits) or make him shoot the spaceship, therefore putting yourself between the enemy and the spaceship is a good idea.

2 Summary Of Work

2.1 The First Person Character

The obvious first step in creating the game was to create the character that the user will control. I originally used a standard asset prefab for a FirstPersonCharacter. The script controlling the First person character were written in Csharp. I removed all four of them as they were designed to control a character on ground and not in the sky and had limits on things like velocity, torque etc. I replaced these scripts with a single script to control movement called 'thrusters.js'. As the game developed this class takes on a whole new array of functions including sound, display distortion, collision detection, animations and more.

The character moves as described in the assignment brief (A/D controls rigidbody rotation around theY-Axis, W/S control rotation around the X-Axis and J/K control rigidbody rotation around the Z-Axis). Return stops rotation and velocity and the 'X' key controls forward thrust. Removing world gravity and rigidbody gravity was necessary to achieve the weighlessness of space. Once I had this movement established, the next step was to get my character to fire the gun. My first approach was to use raytracing to calculate shot collisions but I quickly realised this would be counterproductive to the game and that part of the enjoyment would come from being able to see and avoid paintballs. I created a bullet prefab and assigned it the bullet tag (to aid in collision detection later), When the 'FirePaint' command was executed (Space key as configured in edit > project settings > input) a bullet was created and propelled at the speed of the character + added velocity of propulsion from the gun. This relative velocity stops the user from running into their own bullets when fired and is a physics principle applicable in both space and on earth.

2.2 The Enemy AI

After movement and shooting were dealth with an enemy was needed. To do this I created a capsule collider. The enemy was assigned a health of 10 and a bullet takes away 5HP per hit. Next I started to design the enemy AI behaviour. I identified 3 key bahaviours which any AI would need to participate in a FPS: lookAtPlayer(), moveTowardsPlayer(), shootPlayer(). Now I had a functioning game (even though we were just capsules) where the player could shoot me and damage health and I could shoot him and damage health.

I experimented with what range the AI should stop moving toward me and start shooting and I found that beween 150 and 600 was a good stopping range for the AI. His accuracy was 100% perfect so I added some inaccuracy (this is controlled by the chosen difficulty in the main menu) which involves the enemy shooting above the target. I imported a blender model of an M4 machine gun for the enemy and an astronaut suit and positioned them so they look realistic. I also added a spacesuit for the player which will be animated to remove paint from the visor.

2.3 The SpaceShip

I added my own blender model of a spaceship, however, I am not very artistic and in the end I imported a second ship (both are still in the gameScene). Impact with the ship results in loss of health and shooting the ship ends the game. I used a preconfigured skybox to control the atmosphere, sun and moon and give the game a space-like feel.

The skybox I ended up using was the purple nebula – a nebula situated in the milky way just south of Orion's belt in the constellation of Orion. As one of the brightest nebula it serves the game well by keeping the light and shadowing pleasant.

2.4 The Main Menu

The main menu system has 4 options: start, difficulty, audio and quit. Intuitively the 'Quit' button immediately exits the game while the 'Start' button immediately starts the game by entering the game Scene.

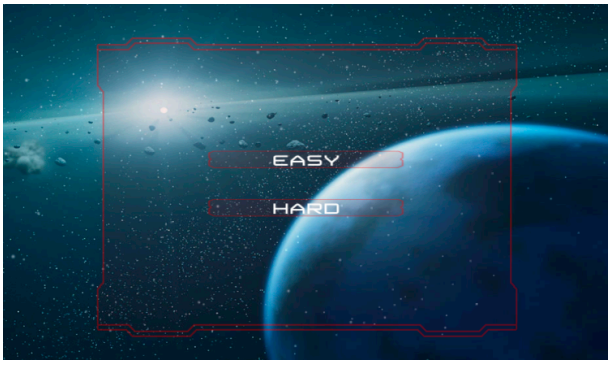
Figure 1. The Main Menu Panel

The difficulty and audio buttons hide the main menu panel and display their own UI Panel. The audio options are simple and only allow on/off as shown in figure 2. The selection here accesses a static boolean variable which is assessed before any sound should be played. Once an option has been selected the audio panel is hidden and the user is redirected to the main menu panel seen in figure 1.

Figure 2. Audio Panel

There are two difficulty options: Easy and Hard. The player's choice of difficulty affects 3 things: the % liklihood of the player being hit by the enemy's paintball, the rotation speed and the movement speed. Again these are being stored as static variables which, in UnityScript equate to global variables.

Figure 3. Difficulty Panel



2.5 HUD

Displaying the HUD is an important aspect of the game. It displays the healthbar of the player and the enemy and their respective speeds.

Figure 4. GamePlay showcasing the health and the speed of enemy and player movement

2.6 Experimentation

When the player is shot by the enemy AI, the player has a 1/3 chance that the painball hit the visor (it always looks like it hit the visor as the enemy is targeting the gun, not the player).

I opted for a handgun for the player to allow a hand to be free to wipe the visor when a smashed paintball obstructs it. This movement (animation) is achieved using the tab key.

3 Problems Encountered

When I started the assignment I spent far too much time playing around with blender models. Blender is a difficult program to get a grasp of and I should have opted instead for imported images from the get-go. Especially considering that many game developers simply comission blender models or use free models available online.

I also started the learning process in unity by developing a FPS game on the ground. I did this using a Standard Assets prefab called FirstPersonController. This prefab served as a good mechanism for understanding animations, sounds, movement, gravity and scripts. However, when it came time to develop the FPS in zero gravity some major problems were encountered. The prefab scripts put limits on torque and force and disallowed continuous movement. I tried to understand the cSharp scripts and disable certain aspects of them but in the end I removed them altogether in favour of my own scripts.

4 Future Work

In the short term I would like to work on the floating debris, making them spawn automatically (as there are a fixed number of debris items atm) and have them moving in random directions (as opposed to now where they only move forward in the direction they're going).

In the long term I would like to make the game a standalone product and allow 2 players to go head to head in an online player-vs-player battle, rather than trying to play the AI all the time. I would also like to add a feature where waves of astronauts come and try and shoot you instead of just a single round.

5 Conclusion & Evaluation

I learned a great deal about game development during this course. Some comparisons can be drawn between the first and second assignment including the method of disabling/enabling sound, setting difficulty variables, building a main menu and linking different scenes in a game. The order of the assignments was logical and allowed me to understand some of the basic gaming principles when designing my 2d game using gameMaker before moving on to a more technically complex second game using Unity.

6 References

[1] – Unity Tutorials: [https://unity3d.com/learn/tutorials]

[2] – Unity Documentation: [https://docs.unity3d.com/Manual/index.html]

[3] – Unity Forums: [https://forum.unity3d.com/]