Mega Mecha Fighter - Game Design Documentation

Mark Clyne(B00419911)

October 18, 2022

1 Introduction

Mega Mecha Fighter is a 3D fighting simulation game. The aim of the game is to defeat the opponents of the enemy mecha faction. You can choose from 5 different mecha fighters of your faction, each utilising a different style of fighting; some using guns, swords and laser weapons. The aim of this game design document is to inform the reader of the intention and overview of the game.

2 Story

The year is 2333 AC(After the Crash), humans have been at war with the Fernweh, a race of aliens that utilise manned robots to try and take over the Earth, which is just another planet in their domination of the galaxy. You are Rudd, a 22 year old human who was born in to the war of the legends, with a deep hatred for the Fernweh, you have been fighting them every day since you turned of age. You can pilot the earth-made robots, which were created by reverse engineering a fallen Fernweh robot. The earth-made robots are inferior but there are more of them.

2.1 A deep dive

You were instrumental in the last victory against them. You were assigned to protect a large train that was transporting a lot of vital war assets and resources.Everyone believes that you gave you life to protect the train and its assets, that you died a hero.

As of now you are in cold storage, and there is no one to take your place at the battlefront. A place that you are sure to die in. You are a veteran soldier and no one can replace you. The war between human and Fernweh is a close minded one. The Fernweh will not stop until the humans and all organic life on Earth is destroyed. The humans are not about to let that happen. The humans are using all their resources to fight the Fernweh and it is a hopeless cause.

Your body is hanging in suspended animation in a tube in a lab inside a hidden military base. You were supposed to die when the train was hit by Fernweh laser fire, but you escaped in an emergency evacuation pod. You are in a computer simulation

that is trying to create a replica of the last battle by seeing what you would have done if you had survived.

This is all to build you back up. You need to be in perfect fighting condition before the next war. The war will be a fierce one. There will be no mercy. You now need to prove your worth once again before the Fernweh try and destroy the main base. The only thing needed for you to succeed is for you to find motive for surviving. And you may need to ally with your fellow warriors to get you what you need to succeed. You can hardly think of a way out of the current situation but if you still have a fighting spirit then you have to try anyway.

That is the time to get victory.

3 Engine Information

The game engine that is going to be used for this project is BabylonJS, a JavaScript game development engine.

3.1 Engine Version

The project will be using BabylonJS version 5.3.0

4 Version Control

4.1 Type Used

The project used Git version control on a remote Network-Attached Storage(NAS). On top of this, Github is being used as a form of redundancy to ensure that data is being kept in more than one secure location.

4.2 Justification

It was decided to use two forms of version control because of issues in the past where either Github or a local NAS was out of comission, meaning that retrieval of data in some circumstances would not be possible. By using two different forms of version control and project secure storage, access to files is increased twofold.

5 Team Members

The team is made up solely by: Mark Clyne (B00419911).

6 Project Planning

The planning for this game was started in September.

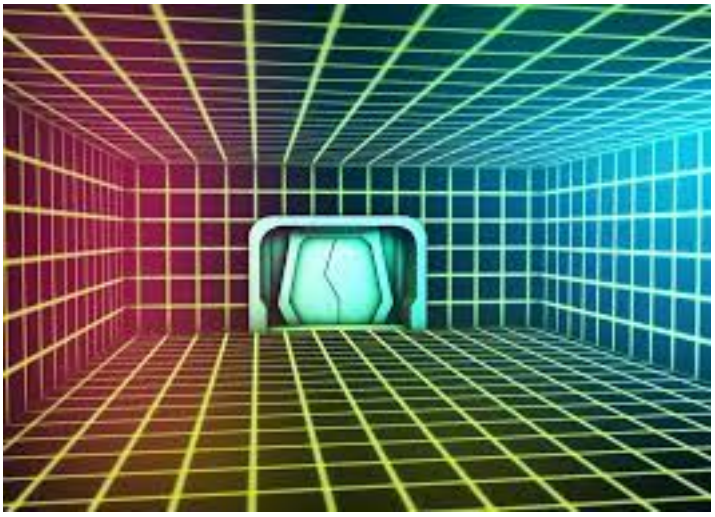
6.1 Project Management

Project management for this game was done using Trello, where goals were set for specific dates and times, could be updated, commented on and some times, removed completely. This helped to keep track of what has been done, what needed to be done and what features would ultimately not be implemented.

7 Game Implementation

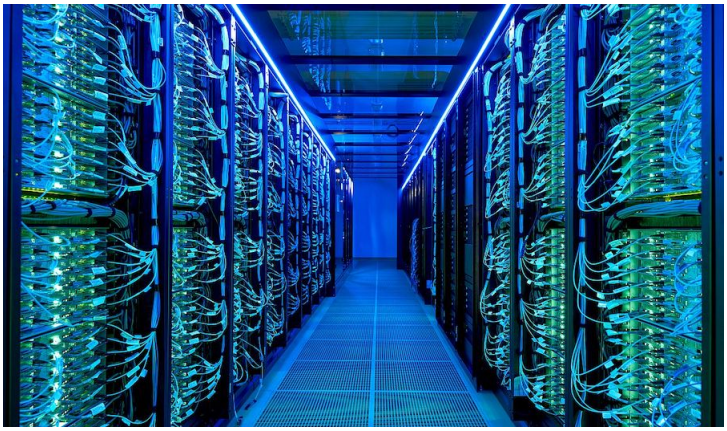
7.1 Level Design

The main area the player can explore is the hidden military facility which they are held. This facility has two major hubs; the Simulation Chambers and the Upgrade Chambers. The simulation chambers are where the player can load of battles with the enemy forces. The upgrade chambers are where the player can program updates and upgrades to the robots in the program. These are stored on a massive supercomputer.



The battlefield is a simulated environment where the player can battle different enemy robots, there are 3 different forms the battlefield can take.

The simulation chambers is an area the player can utilize to start simulation battles against the Fernweh soldiers.



The upgrade chambers is where the player can program upgrades and modifications to the robots they fight with during simulation. These upgrades are carried out by programming in the features using a high tech military computer. The player will write actual lines of *very easy code* to apply the modifications to the robots.

7.2 Visual Design

The visual design was heavily influenced by the rural areas of Scandanavian countries, e.g. Norways countryside
Plains in Norway

The design for all of the Foul are based on what I consider to be humanoid "abominations" that you would see in movies/games such as Resident Evil and Silent Hill. The mix in visuals of Horror and RPG are intended to invoke fear in the player when they are fighting them, the ambient visuals e.g. post processing and skybox are meant to back up this sensation.

The materials and textures for the landscapes were all chosen carefully to reflect the situation. The foliage color reflects the Foul occupation of the main map "The Hoarscythe Plains". It isn't quite the green you would expect from grass, instead has a brown tint to it. This is attributed to the Foul being present and leeching the life from the land, slow but surely.

7.3 Mecha Choice/Design

- Fighter Mk I:

- Cannoneer Mk II:

- Tank Mk II:

- Command Unit IV:

- Medic Unit III:

7.4 Audio Design

The music for the game will be futuristic synth music and deep emotional music, used to emphasize the fast paced and dystopian gameplay. Slow building and hyping music will be used to convey emotions during gameplay and cutscenes.
e.g.

- Gundam Battle Assault Complete OST
  `https://www.youtube.com/watch?v=X3HUyI3xspM`
- Darling in the Franxx - Vanquish
  `https://www.youtube.com/watch?v=Bfkc5bX4spE`
- Neon Genesis Evangelion OST
  `https://www.youtube.com/watch?v=A7zVgSZS4tQ`

7.5 User Interface Design

The User Interface is designed to be minimalistic but also modern looking with smooth animations in the menus. The main goal is to provide the player with a visually clear User Interface that still looks good. It also needed to be placed and the correct colors as to not distract the players when they are in combat or exploring.

7.6 Features To Implement

- Customised AI based on the AI Behavior Toolkit, changed to allow for more dynamic combat with enemies and bosses.
- Heads Up Display(HUD) that informs the player of their HP(in red), stamina(in green), and MP(in blue).
- Pause Menu which leads the player back to the main menu and an option to resume the game.
- Main Menu that has options to start the game, look at the settings, and exit the game completely.
- Upgrade Menu that has options to upgrade your robots with new programming.
- Simulation Menu that has options to choose the arena you wish to fight in, the difficulty of the enemy and what enemy you want to fight (can be randomized).
- Objectives HUD

8 Testing

8.1 Testing Methodology

Three different testing methods were used to test the game. These methods are *"Functionality Testing"*, *"Compatibility Testing"* and *"Play Testing"*.

- Functionality Testing: Functionality testing is a form of White Box testing carried out by the developer where generic issues such as broken assets, audio and video issues and system scalability for the desired system specifications are tested for and picked up before play testing begins.
- Compatibility Testing: Compatibility testing is a form of white box and black box testing where the game is tested on many different forms of hardware to ensyre the game will run on many different systems with different configurations. This includes but isn't limited to ensuring that all UI elements and the game scale properly with screen size.
- Play Testing: Play Testing is a form of black box testing where user engagement, opinions and collection of generic issues can happen. This is a form of testing where the play test end user can report to you what is, and is not, working.