

**Clash of Champions**

**Web Games Development 2 (COMP10057)**

**Documentation**

**Manuel Menor Flores**

**B00356825**

**Neoptolemos Papadiofantous**

**B00267675**

**Thomas Nicholas Mackrell**

**B00287948**

**28th April 2019**

**Lecturer: Mario Soflano**

# Introduction

As part of the project for the Web Games Development 2 module at the University of the West of Scotland we were required to make an online multiplayer game of our choosing. This game would then be embedded in a multifunctional webpage, also of our making, designed and built to promote and host the game and present it to the users with a design document assisted by pictures and a video demonstration. After considering various themes and game styles, we decided that we would create a 2D fighting game. The game will have an arena-style combat system set in a map inspired by traditional platformers, with a retro feel to it.

Throughout this document the game designed for the project will be discussed. This will be done using similar titles as examples to compare and draw influence from after a brief overview of the genre that was chosen. This will then be followed by an explanation of the thought process behind what elements the game will contain, as well as a short story to accompany the game and give it more depth. The next section will present the software and methods used to create the game, including the multiplayer database and shall be accompanied by code excerpts and explanations that show how the more significant parts of it work. Finally, to close off, a short description of ideas for future implementation and improvement will be made.

# Game Overview

## Genre

Platform fighting games are a subgenre of fighting games blended together with platforming games and adopting aspects from them. They maintain the combo-based combat system that traditional fighting games utilise but also add platformer aspects to it, such as collectibles and a variety of platform focused stages, and stray from the usual knock-out win conditions that fighting games use; rather opting to use other unconventional scoring methods. The games that are styled after this genre are generally designed to have party-oriented elements, ideal for casual play in groups of up to four players that can fight and cause mayhem in a small arena for fun.

## Similar Games & Influences

Once the game genre was decided, similar other games had to be found to draw inspiration from. Multiple options were considered as there is a large variety of 2D fighting games with varying levels of complexity and a wide range of different gaming styles. Focus was mainly put on platformer fighting games, and the most prevalent results of the search are presented in the following subsections.

### Super Smash Bros. Series

When it comes to platform fighting games, the main title to look to for influence is Nintendo’s Super Smash Bros. This encompasses five different games, released from 1999 through to 2018. An innovation, the title brought together various characters created by Nintendo and a number of its second-party developers over a series of games to fight in different arenas, also given themes that drew inspiration from other Nintendo titles. The aim of the game is for the player to knock their opponents off the map by doing enough damage to them, unlike the tradition health bar style that most fighting games adopt. This can be done by utilising an extensive melee combat system with a large variety of combos and special abilities – built and expanded throughout the series of games, increasing in complexity with each one. If an opponent is knocked off the map they will lose a life and then respawn, and the game continues until only one character is left alive.



*Fig. 1 – Super Smash Bros.*

The game is playable both against the computer AI as well as against up to four other players but was predominantly designed as a multiplayer game, something which garnered Nintendo a lot of praise and even resulted in a large competitive community being built around it.

### Rivals of Aether

A game heavily influenced by Super Smash Bros., Rivals of Aether was released for Microsoft Windows and had gameplay mechanics heavily based on those of Nintendo’s title. The game has ‘lower’ graphics quality, opting to go for pixelated characters and environment, compared to a higher definition, ‘cartoony’ look and has basic combat mechanics. What makes it stand out is the interesting and unique feature that appears in the combat. The characters of the game each get their theme from a different element and so each one has its own special elemental attack. The interesting part is that – and subsequently part of what made it a successful game in its own right – these elemental attacks can also affect the game’s stage, a mechanic which is enticing and can make game matches less repetitive.

*Fig. 2 – Rivals of Aether*

### Brawlhalla

Following in the steps of Super Smash Bros, Brawlhalla is a title owned by Ubisoft that plays similarly to the previous two titles examined. The aim is for the player to deal enough damage to their opponents so as to knock them off the map through the use of simple controls and one-button special moves, as well as various player mobility options such as double-jumping and different kinds of dodges and dashes. Brawlhalla is worth noting for two reasons. The first is the fact that the game has a large and ever increasing roster of unique characters (over 40) – each of which has different amounts of the four statistics the game has (Strength, Dexterity, Defense and Speed) assigned to them – a factor which is key for games of this type in keeping the player base’s interest by providing them with new and fresh aspects to play with. Secondly it has an interesting weapon system, where different weapons drop from the sky for players to pick up, with each player being able to hold 2 weapons from a pool of 12. This makes for an interesting challenge, as the players must formulate their game strategy around what weapons they manage to pick up, as each one has its own bonuses and drawbacks.



*Fig. 3 - Brawlhalla*

### Nidhogg 2

Produced by indie developer Messhof, Nidhogg 2 takes a different approach to platform fighting games. Instead of using a traditional scoring system of point scoring or kill count, two players are placed in the arena for a pre-specified amount of time. During this time the players must sway the ‘balance bar’ in their favour by getting kills on the other player, making use of the multiple weapons – both close combat and ranged – that can be picked up around the stage. The one who has control of the ‘balance bar’ when the time runs out is the winner. The artistic style of the game is also interesting as, despite being mostly minimalistic and ‘lower quality’ than other similar games, it has depth and pays attention to smaller details that make it unique.



*Fig. 4 – Nidhogg 2*

### 2.1.5 TowerFall Ascension

An interesting case, TowerFall Ascension does not use the traditional approach to platform fighting games. The title, produced by the indie company Matt Makes Games, is a platform, fighting, and battle royale game where up to four players are placed in an arena with a bow and a limited supply of arrows and the last one alive is the winner. Arrows can be replenished by picking up the ones that have already been shot around the map and there are also various buffs (shield, speed, etc.) that can be found in the form of pickups scattered throughout the stage. The stage style used in TowerFall Ascension is also different; compared to traditional platformers, the maps are separated into rooms that are conjoined rather than having multiple separate smaller platforms to break up the playing space.



*Fig. 5 – TowerFall Ascension*

## From Research to Game Design

After looking through a number of examples a theme for the game was decided, drawing inspiration from many of them. The game that was created for the project is a platformer fighting game, where the two or more players are placed in an arena and they must fight to eliminate each other. The players utilise ranged abilities that fire projectiles which take away a life if they collide with the opponent. Each player has three lives in total and loses once all of them have reached zero. Each player also has their own kill count. After the first kill the normal projectiles – which do one point of damage – are replaced by an empowered version that do two points of damage, making it easier for them to take out opponents.

The players can move and jump around the various platform of the map to dodge their opponent and gain a tactical advantage over them. While doing this they must also avoid suffering environmental damage from the rocks that fall from the ceiling at regular intervals. These rocks will do a single point of damage if they graze a player but will take off two lives if the player does no move out of the way in time.

The art style chosen for the game was simplistic. It was evident from the research that high quality visuals are not what makes a game and many people find the 32-bit style adopted by most platformers endearing. The level is built to look like an underground dungeon and the characters are magic wielding heroes.

## Lore

In a different reality, humans have been forced underground due to a massive surge of magic that decimated the planet’s surface. The surviving populations have formed underground nations to survive, but have now realised that resources are limited. Rather than going to war for them, each city has put forth a powerful champion; humans who somehow did not get lost in the magical storm, but emerged from it with immense power gained from harnessing the magic. These champions are pitted against each other to fight for their city and gain the much needed resources for their people to survive. As one such champion, the player must enter the arena and fight for the honour of their people.

## Game Assets

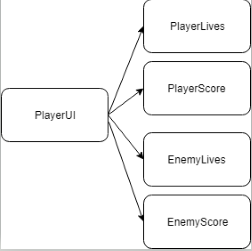
All the game assets used in this game were taken from another game created by one of the developers on this project. Tiled is a map editor and was used to develop the map of the game. The player character was taken from this website that shared this character for free. <https://rvros.itch.io/animated-pixel-hero>.

# Game Design

## Use Case Diagrams

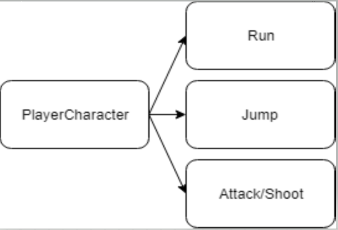
Once our team discussed which were going to be the requirements for our web game, some use case diagrams were created to describe functions it will be divided into. Hence two different use case diagrams are shown below showing the functions that the player and player character can do respectively:

* Player UI:



Hence the person who is playing will be able to see player lives and score. On the other hand he will be able to see enemy lives and score too to contrast the information with itself.

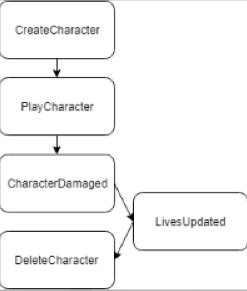
* Player Character:



These are the agreed actions that the player will perform, it will be able to move in left and right (running), jump and shoot bullets to fight against other opponents.

## State Diagram

Since state diagrams show the number of states which an object is composed, it is a useful tool to know lifecycle of them, representing events making changes between states possible. Therefore, a state diagram was elaborated showing the different states on which a player character can be:



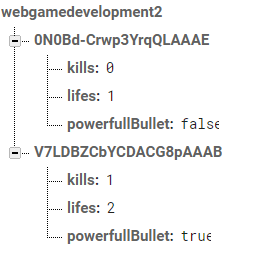
# Implementation

## Phaser

As proposed in the labs, Phaser was used for creating the actual game. Phaser is a free 2D game framework software, used for creating HTML5 games. It is ideal as it can be run in most browsers (any that support the <canvas> HTML tag) and is renowned for its fast rendering speeds. The games that are made through Phaser can be made with the use of JavaScript, which is what was used to simplify its addition to the realtime database.

## Firebase

Firebase is a web application development platform with multiple different uses, one of which is that it provides a free real-time database and backend as a service. It provides an API which can be used to synchronize data, which is stored on Firebase’s cloud, across multiple clients. Unlike other SQL databases, Firebase is asynchronous meaning that, despite the fact that multiple processes are running, the browser will be responsive and get updates at regular intervals whilst still allowing the user to perform other actions and all the information used is structured with JSON files as follows:



The structure of the firebase real-time database is a JSON tree as shown. So users are identified using its own id. As children, they have the information related to that player, that is to say, number of kills, lives remaining and if they have the powerful bullet mode enabled.

## Socket.io

Socket.IO is the JavaScript library used, instead of the proposed Eureca.IO from the labs, for implementing the methods which call to the server in a realtime web application. It contains two event-driven parts; the client-side library which runs in the browser and the server-side library which utilises Node.js. Together these enable realtime, bi-directional communication between the clients and servers. It was used to make the transition of the project from Phaser into a working version hosted through Firebase easier to handle.

## Heroku

The game has been hosted online using this free platform. Heroku allows developers upload their projects in the cloud being able to run them online. Heroku interoperates with Github to upload projects on the web, it makes the process of uploading the project straightforward. Once the game has been hosted and running we got the next link that can be used to play it: <https://frozen-journey-61507.herokuapp.com/>

The reason because the game was decided to be hosted online is because it is easier for users to display the game, users just have to put previous link in the browser and play it. Otherwise, they will have to run the server locally using node.js and make sure they already installed needed packages i.e Socket.io.

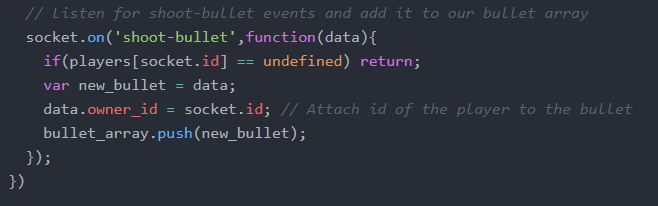
## Implementation of the game

Here, it is going to be explained how some mechanics were implemented to make the understanding of the flow of the execution easier. So, we are going to focus in the shoots of bullets and their update in the server to every player playing the game.

First of all, we need to capture inputs in the keyboard to detect that the users wants to make the character playing, so it has been done in the Game.js like this:



Depending on the orientation of the character the speed of the bullet will be negative or positive (left or right respectively). The flag shoot is set to true and only once we finish pressing X button (we do not want to shoot again), shoot flag will be set to false. As you can see in the previous image, there is a line where a socket.emit is done, in that line we are sending information about the bullet to the server. So the code in the server is the next one:



This main function of this piece of code is to store the information of a new bullet in the database, as you can see, the bullet will have x and y position, x and y speed, orientation of the player and its id.

Once the bullet has been stored a system has been developed to update the bullets in the game of every player broadcasting these updates to them:

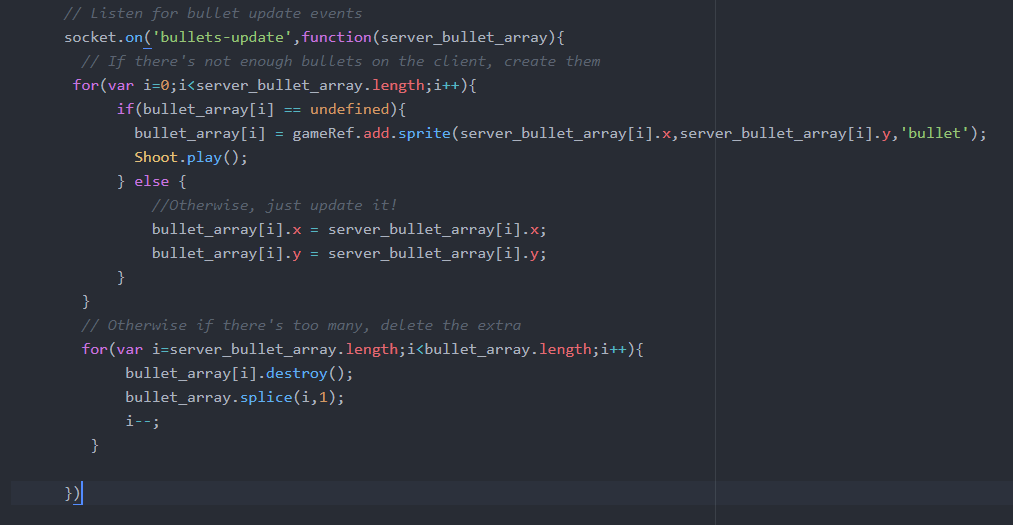


One of the functions of the server, is to call ServerGameLoop function 60 times per second, using an internal interval:



Hence, the server will run ServerGameLoop() updating every position of the previous array were information for bullets were stored. Therefore, in the line of code bullet.x += bullet.speed\_x the position of the bullet is incremented. Afterwards, since we have an array with players’ positions in the server too, we can check if the position of a bullet is near enough to a player to know if the bullet hits him (and let them know sending a broadcast to every player using io.emit(‘player-hit’) and deleting the bullet). The last checking before sending the new information about the bullets to all players, is check if the bullet is out of the bounds of the map, in this case it needs to be destroyed (bullet\_array.splice(I,1).

When it all has been checked, we are ready to send this information of the bullets updated to all player using io.emit(‘bullets-update”) broadcast to all players’ game:



Here, positions of bullets are updated in players’ screens, creating the bullet if it is the first time it appears in the screen or just updating its position if it already was in the player screen.

# Future Expansion

The game shows promise and could in the future be expanded upon. The first steps to take would be to focus on specific features that should be worked on first. A possible major change would be a character roster system could be implemented, giving the players more freedom with what they decide to play as. Finally, multiple stages could be introduced, to avoid the monotony of only playing in one. After the more straightforward improvements have been made, further gameplay changes could be made. This would include additions such as different types of magic attacks and possibly even special attacks – unique to each character – as a game such as this is limited only by the imagination.