
Thinking Emoji

Drexel University
3141 Chestnut St, Philadelphia, PA 19104

Covid Crossfire

4th September 2020

Introduction

Team Members:

Alex Hung, Lester Chacon-Posada, Jacky Chen

Code Repository Link:

https://gitlab.cci.drexel.edu/ah3472/cs375_project

Overview:

COVID Crossfire is a 2D roguelike, bullet-hell web game. The user will play as a white blood cell that travels through blood vessels while fighting off a particular disease that originates from bats, COVID-19. The user has to create an account to play. This system allows the preservation of the user's high score which will be displayed in a global ranking system.

Architecture

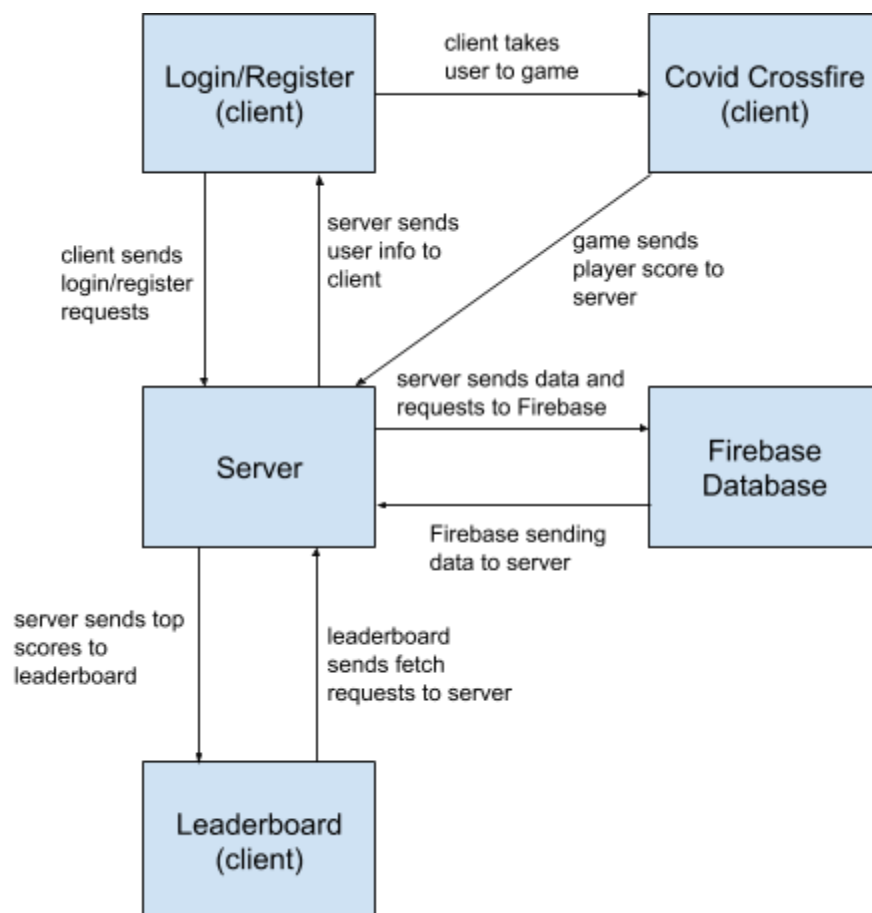
Technologies Used:

- **Javascript, HTML, CSS:** used to program basic UI, game mechanics, and server code
- **Firebase:** used for user authentication and the storage of high score data of players
- **Pixi.js:** used for rendering and animating images on the canvas
 - One obstacle was figuring out why the resources in the application loader was not rendering a specific sprite.
 - Pixi.js works by setting the attribute of the script source to the Pixi.js file. By doing so, we were able to gain access to render sprites and animations using their API documentation.
- **Node.js:** used to locally run our server

Architecture Overview:

Upon opening the application, a login page will show. If the client does not have an account, they can create a new account with an email, username, and password. This information would be stored in our database using Firebase. This information will associate that selected account with a unique ID and any highscores the client makes will be reflected upon that account. If the client already has an account that is in the database system, they can log in with their email and password they had created.

One of our main communication systems from client to server is our game's scoreboard feature. In every game, the client will have a default score of 0 and points will be given to them upon defeating enemies. This occurs up until the player dies. Upon death their score is then sent to our server, where their score and their username will be stored and shown in a ranking system.



Reflection

Original Planned Features:

Player sprite	Enemy sprites	Tilemaps	Different selections of weapon
Collision Detection	Transition between scenes/maps	Hit points implementation	Defeating enemies/Pick up loots
Music/Sound Effects	Damage implementation	Dynamically generated rooms	Score implementation
Animation	Bullet Projectile	Create multiple enemies	Boss fights
Create multiple environmental settings	Deployment	Create account/login authentication	Saving player's highscore in a database

Features Implemented:

*Green = Features Implemented

*Red = Incomplete Features

Player sprite	Enemy sprites	Tilemaps	Different selections of weapon
Collision Detection	Transition between scenes/maps	Hit points implementation	Defeating enemies/Pick up loots
Music/Sound Effects	Damage implementation	Dynamically generated rooms	Score implementation
Animation	Bullet Projectile	Create multiple enemies	Boss fights
Create multiple environmental settings	Deployment	Create account/login authentication	Saving player's highscore in a database

Incomplete Features

Create multiple enemies	Boss Fights	Transitions between scenes/maps	Deployment	Create multiple environmental settings
-------------------------	-------------	---------------------------------	------------	--

- We did not make these features simply due to time constraints and we decided to prioritize key features that show the main points and the theme of our game.

Things to do differently next time:

Alex:

- Come up with an game architecture before implementing because our codebase was very messy because we did not have a thought out plan.
- Find another way to implement colliders because our method used a lot of rectangles so it had to loop through each rectangle and check for collision. Especially with the number of bullets.

Lester:

- Be more clear on the theme of the tilemap I made. The enemies did not really fit the theme of the map (The blood cell inside the body). I had covid sprites but they were all bad in my opinion. Going back to make a good model for a COVID enemy is something I would do.
- Many ideas and sprites that I created were not used in the final product for various reasons. Next time I would have taken a step back and see what was needed rather than making a lot of sprites that would not even make it to the game.

Jacky:

- Find a better method on implementing the rotation of the player's weapon based on the position of the mouse cursor. The gun is able to turn from the left to the right but cannot turn from the top and the bottom direction, which makes the shooting portion of the game inefficient.
- Implement a better movement control for the player. When the player goes diagonally, the speed is much faster compared to moving in 4 directions.