**Game Description**

The game is a 3D third-person multiplayer spaceship shooter. The player controls a spaceship that is able to fly around and shoot. The player’s objective is to kill all other players by shooting while avoiding collisions with environmental objects and other players, as well as not being shot by other players. The player can only shoot one missile at a time that will detonate on impact.

**How to Run the Game**

The server must first be running to open the game. The server is run from terminal with the command:

node Server.js

When the server is running, it will listen to port 8080. The open a client, open a web browser (Chrome is suggested). Open the URL:

localhost:8080

To open up another client, use the same URL in a different window or tab.

**Controls**

|  |  |
| --- | --- |
| **Key** | **Action** |
| W | Fly forward |
| S | Fly backwards |
| A | Turn left |
| D | Turn right |
| Q | Strafe left |
| E | Strafe right |
| Space | Shoot |

**Implementation**

There is a server and a client to the game. The server controls the connections using socket.io, Node.js and Express.js. It holds all the players in a JavaScript array. Each player has attributes for their player id, if they are alive, ship position, ship rotation, missile position and if their missile has been fired. There are functions that control/manipulate the players array. The server will check if players have been hit by a missile or have collided with another player’s ship. If this happens, a message will be sent to the clients that they have died, and the client will handle the rest. The server takes updates from the clients and will update the player array when data is sent from the clients. The server will send back the information about all the other players. When a player has died, it will emit a death message to the users who have died.

The client side of this uses Three.js to render the scene. It will initially create the environment with a skybox, the sun, randomly generated asteroids. The player ship is rendered at a random spawn point. For the sake of testing purposes, there is commented out code where the ships will spawn relatively close to each other. The ship is given keyboard controls to move and shoot. A chase camera has been implement to follow the ship around from behind. When a missile is shot, a red sphere is created that will constantly move forward. The client checks if the player has collided with an environmental object, and will die if it has collided. The other player’s ships and missiles will be rendered if they are new and translated/rotate if they exist. When an enemy ship dies, it will be removed. The client will constantly send updates to the server about the position and rotation of the ship and missile. It will take the information provided by the server and update the other ship locations and other missile locations. When the user dies, the client will send the player to another html page that notifies the user that they have died, as well as a button that lets the player try again.

**Issues**

Issues that have the game has is that the game seems to run slow and there is a delay in the movement of other ships. This I think is due to the implementation of the rendering of the other player’s ships and missiles. There are many loops that check all the conditions and causes the game to run slow. This could be changed with a different implementation. Another issue is that the renderings of the other ships seem to flicker. This is due to the implementation as the ships are removed from the scene and returned a while later. The new implementation that would solve this issue would be to remove the ships and then immediately add the new ship to the scene. The game seems to run slower on Firefox compared to Chrome. I suggest running the game on Chrome.