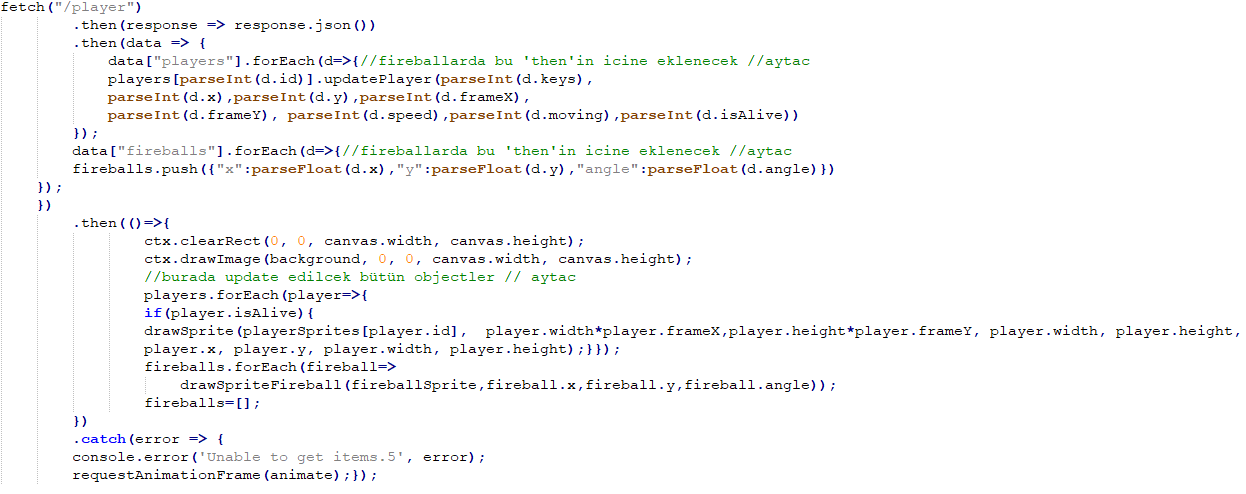
Front End

Browser part of the game is responsible for drawing the game state and relaying the user inputs to the backend. It consists of a html canvas and the underlying JavaScript code.

When the page is loaded, it asks the server to spawn a new player and saves the id returned by the server.

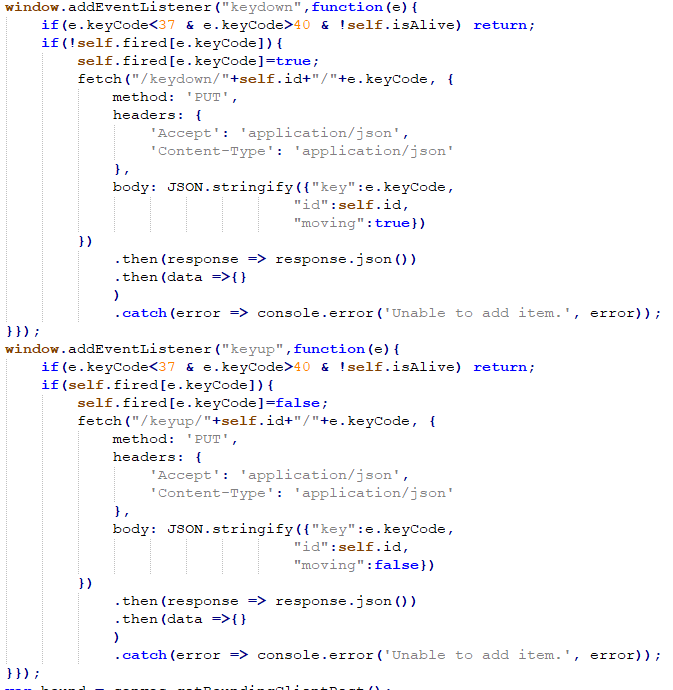


After the new player is initialized, animation loop begins. During every loop, a request is sent to the server in order to get the game state. Server returns a json file that contains information about the position and status of objects (players, projectiles etc.). This file is then parsed to update existing objects or to create new ones, and then objects are drawn. Attacks are drawn as simple png files adjusted according to their angle and players are drawn as sprites based on the direction they are looking.



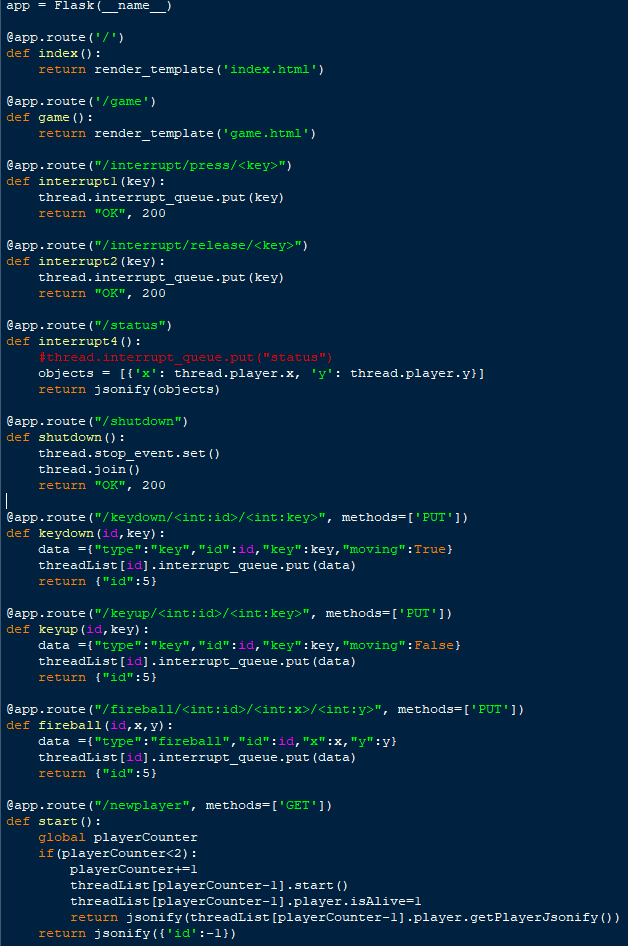
Additionally, three events exist to keep track of key presses, key releases and mouse clicks. This information is relayed to the server in order to coordinate player movement and in case of mouse clicks, making attacks.





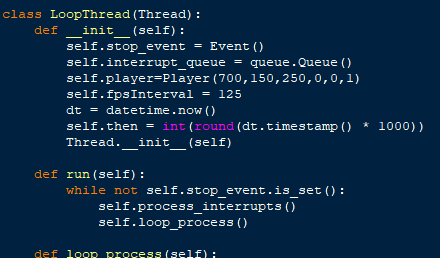
Back End

Backed code is written in Python and uses Flask library for browser communication. Additionally, threading library is used for thread making. Server side consists of threads corresponding to each player, a main handler and routes for communication.

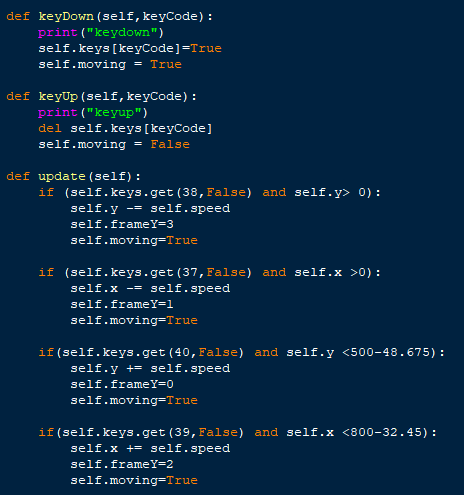


Keydown/keyup keep track of pressed keys for player movement, newplayer is used when a new player joins and fireball is used for creating a new fireball object with appropriate speed and angle values.

During the thread loop a collision check is run to determine if a player is hit. If so, damage is registered for that player and the attack is despawned.



When a request is sent from the browser, an interrupt event is made for the corresponding player based on player id and player status is adjusted accordingly. If an arrow key held, player speed is adjusted in the corresponding direction.



If a mouse click is sent, a new attack object is created, and its angle and speed are calculated. When a browser asks for an update, the objects are turned into json and sent back.

