

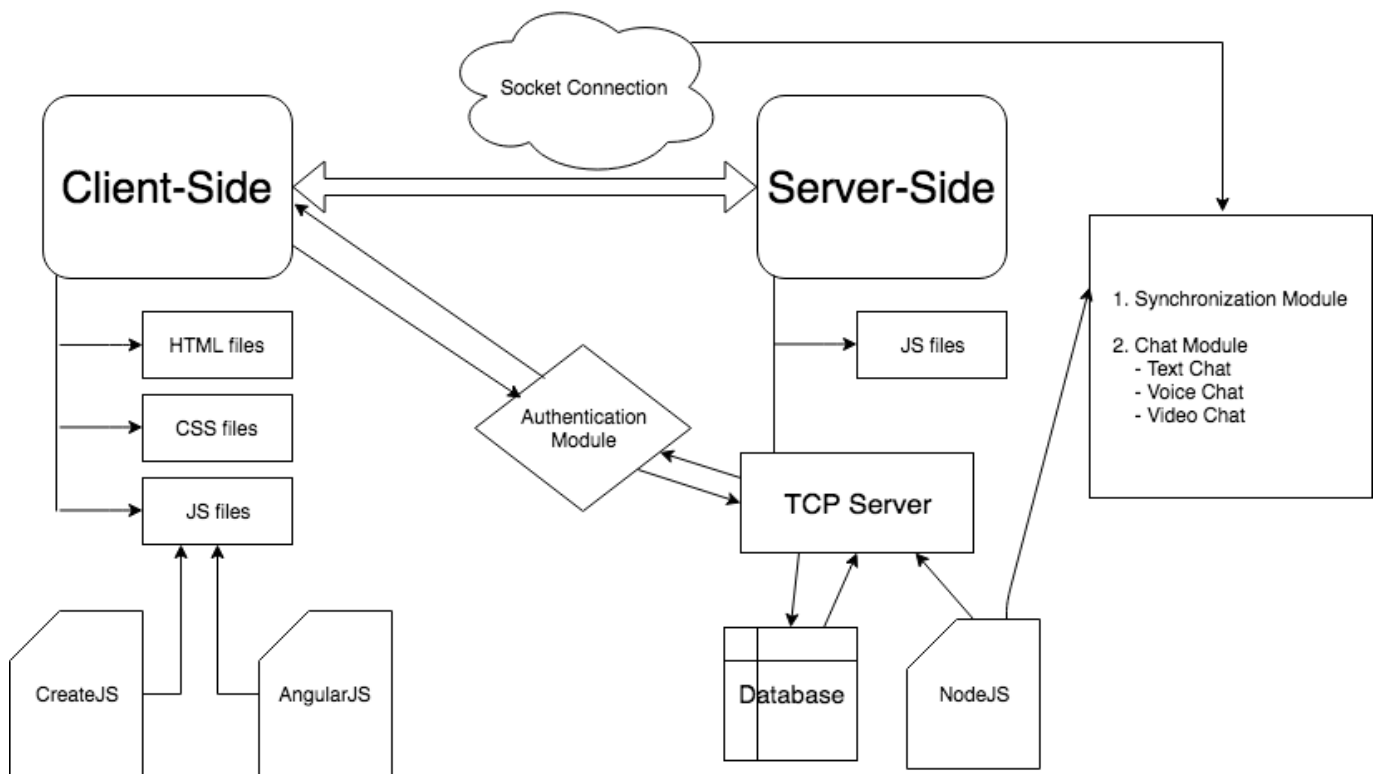
COP290 : Assignment 3 - Pocket Tanks Design Document

Sahil Bansal (2016CSJ0008) & Sahil Singh(2016CSJ0025)

April 15, 2018

1 Overall Design :

The following picture shows the overall design of the game: A brief explanation of



each of the components is given in the following section.

2 Sub-Components :

2.1 Client-Side:

- It will contain the HTML, CSS and Javascript files required for the game including other images, sounds and other non-code files.

- The two frameworks used for the front-end design of the game are **CreateJS** and **AngularJS**.
- **CreateJS**: It has four libraries.
 - **EaselJS** will be used to easily handle the functionalities of the **Canvas** HTML element.
 - **TweenJS** will be used for handling the animations.
 - **SoundJS** will be used for music and all other sounds in the game.
 - **PreloadJS** will be used for managing the loading of game assets like tanks, terrain and the weapons.
- **AngularJS** will be used for handling events like button clicks and login screen, and also for several user interactions in the game.

2.2 Server-Side:

- A **TCP server** will be created with the help of **NodeJS**.
- **Firebase** will be used as the database which will store the user's game statistics and also the login credentials of the users.
- **Socket.io** will be used to establish a two-way connection between the client and the server. Whenever one of the uses fire a weapon during the game, there will be a flow of data from his end.
- **Modules**:
 - **Synchronization module**: will ensure that the game is synchronous. It will require both socket and node.
 - **Authorization module**: will be used to handle login.
 - **Chat module**: will also be implemented with socket.

3 Testing Sub-Components :

3.1 Client-Side:

It will be tested by first designing the Offline Game and ensuring that all the functionalities of the game work properly. The following points will be checked:

- Terrain and the tanks are generated properly.
- All the buttons (changing angle, power of the weapon, move tank) are working properly.

- The weapon is being fired and **projectile motion** is properly implemented.
- The terrain is being distorted when the weapon hits the terrain.

3.2 Server-Side:

The following points need to be checked:

- The authentication module works properly and connects to firebase.
- The chat module works properly and text messages are sent and received. After checking for text messages, further features like voice and video will be worked upon.
- Synchronization module works properly i.e. the game is updated on both the players screen as soon as one of the player does some action.
- Some debugging functions will be made that just print some text to make sure that the functions are executed when requested at one of the end from the other end.

4 Automatic Intelligent Attacking Opponent:

The three modes provided are:

- **Easy:** It will be a random attacking opponent. So, it won't optimize the power and angle but they will be chosen at random.
- **Medium:** Some of the attacks will be intelligent while some random to give the appearance that the game is of the moderate level. Also, it won't always make optimal choice of the weapons.
- **Hard:** All of the attacks will be well decided so that the player gets a high enough challenge. Weapons would be chosen with a proper strategy.