
Design Document

Assignment 3-Pocket Tanks



Author:

TABISH SHAIKH(14)
AAKAR SHARMA(66)

1 Introduction:

An interactive multi-player game implementation of the famous game of pocket tanks with additional features such as chat and voice chat embedded within the application. Pocket Tanks game involves 2 tanks on two opposite sides of a randomly generated terrain. The weapons obey laws of projectile motion of an object, therefore to fire a weapon you must set its angle and power so that the weapon follows a projectile path and hits the enemy tank. Each player gets 10 rounds of weapons and at the end of 10 rounds the player with the highest score wins the game. The game is intended for audience of all ages. This is an online game created using NodeJS, WebRTC and CreateJS. The server of this game is deployed on Heroku and Firebase is used as a database to store data. The game can be played with both computer or with another player online. To play the game first the user needs to login. Matches history and the achievement of every player are stored individually. The user can view his statistics such as game wins, losses and draws by clicking the statistics button on the top right corner.

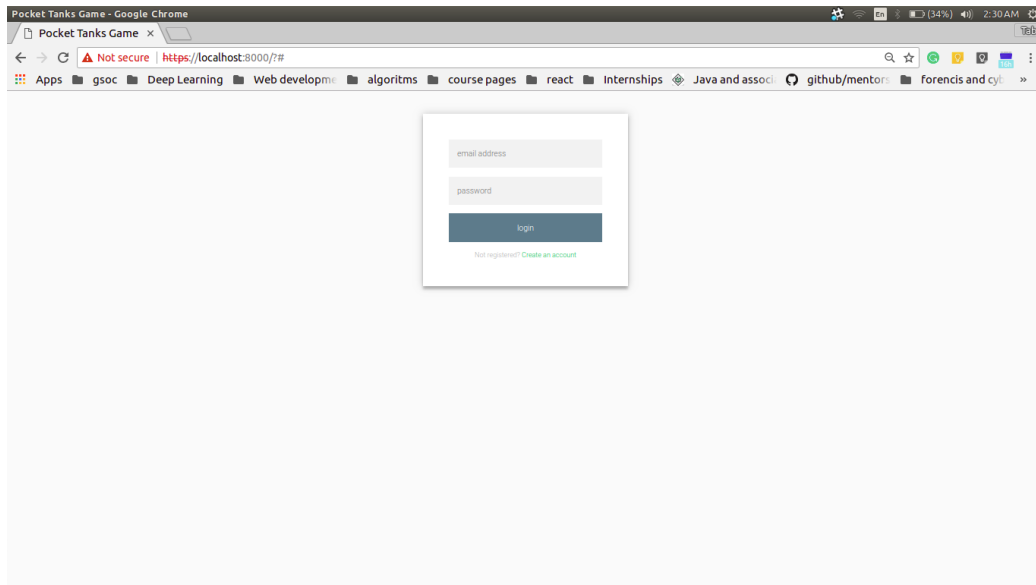
2 Features:

- 1 This is an online game in which more than 2 players can play game online. Any number of players can connect simultaneously and play multiplayer game with whomsoever he wishes.
2. We are also storing the statistics of all kinds of games, both between the computer and other players. Scores and match history of each player is stored online on Firebase.
3. An intelligent computer is also implemented which decides its move on the basis of your location and position.
4. In the chat-section we have implemented a simple text chat along with audio and video chat features which can be enabled or disabled as per the user's choice. You can also text chat or voice chat with other players in real time.
5. For security we have done AES encryption on the WebRTC packets. Also we have incorporated RSA security certificate in our site so that it will be secure.

3 Graphical User Interface:

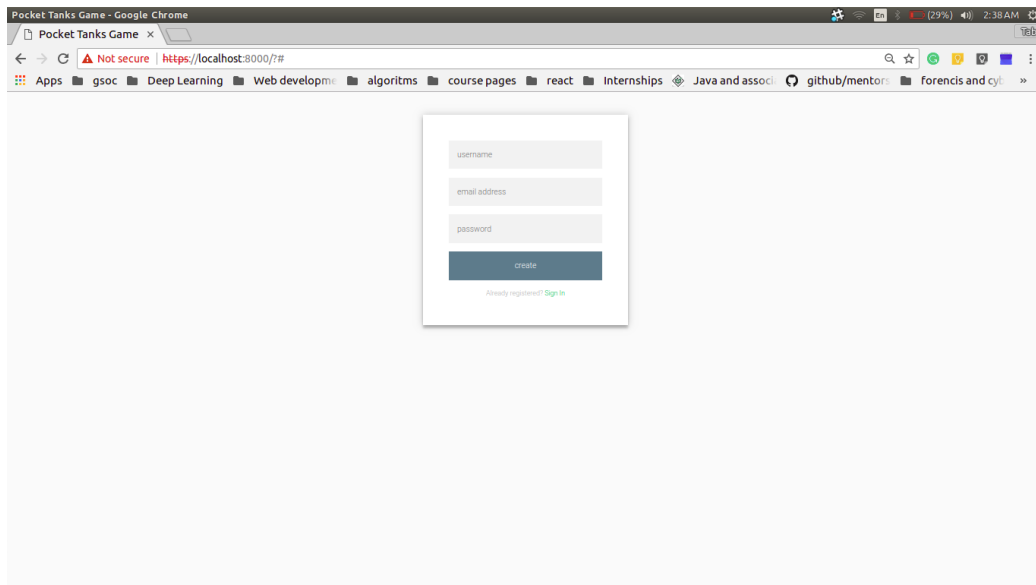
3.1 Login screen:

To play the game the user would be required to login into the game with an username and a password. If the user does not have an account then he will be redirected to the signup page.



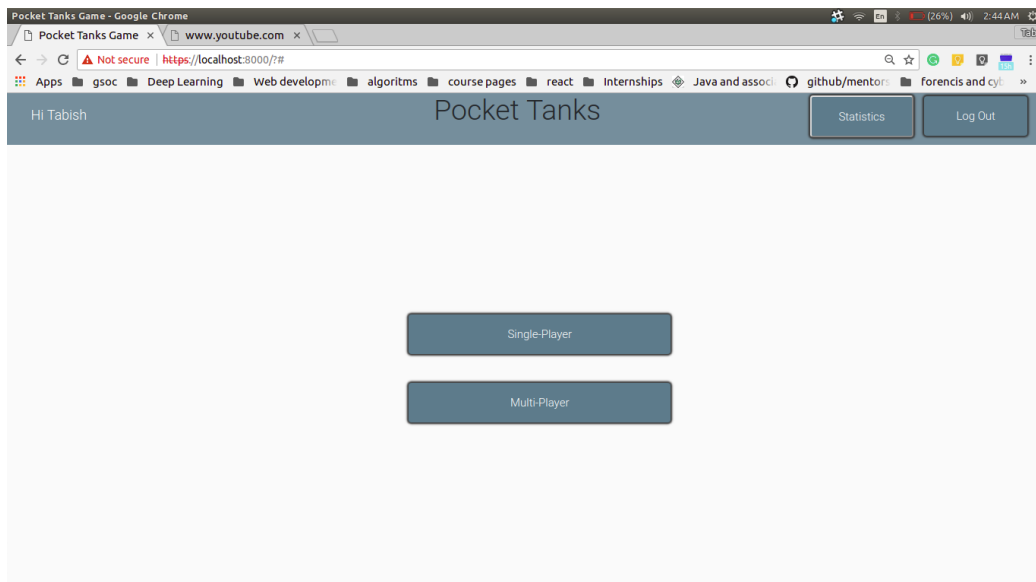
3.2 SignUp screen:

On not having the account of the game the user will be redirected here where he would have to fill in the necessary details such as user name , password.



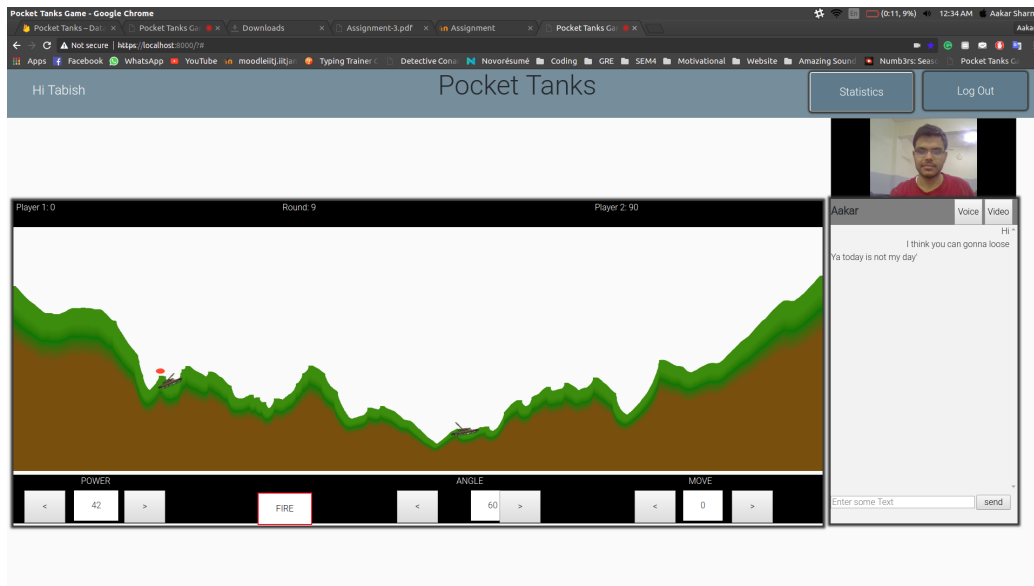
3.3 Options screen:

After login the user will face a choice whether to play the game one on one with computer or a multi-player game.



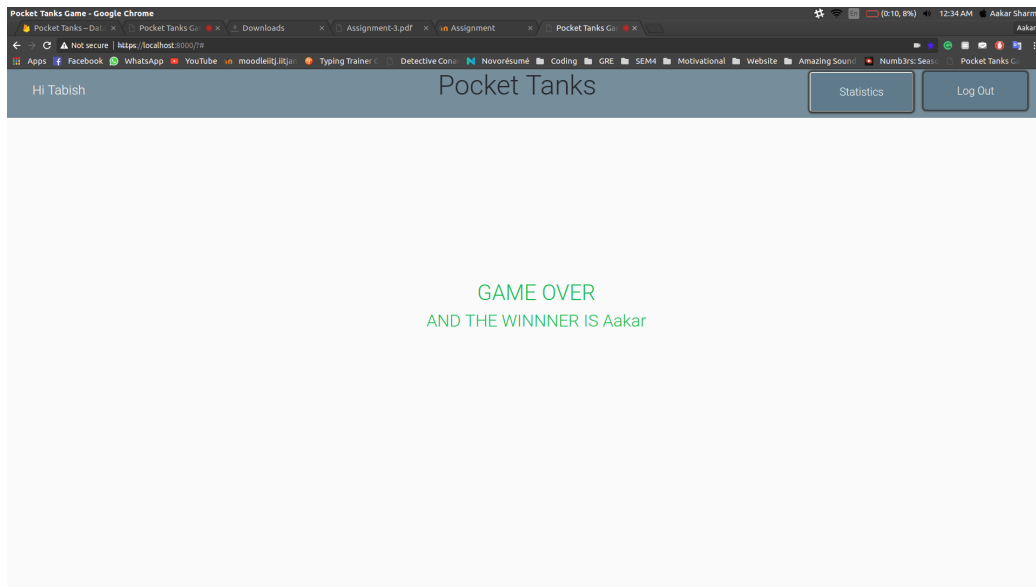
3.4 Gameplay screen:

The game will run on this screen. Terrain, controls, tanks and statistics will be shown on this screen.



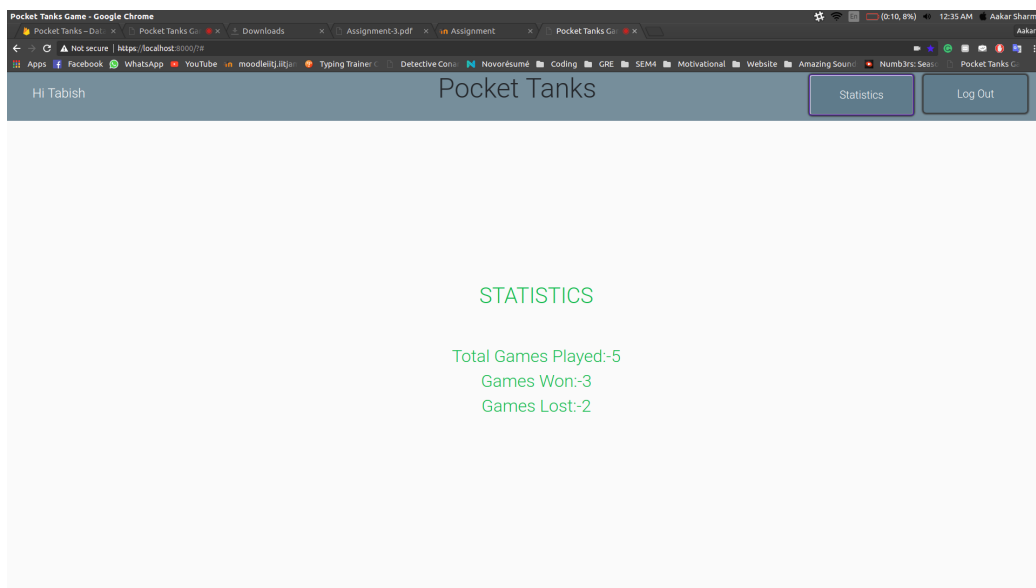
3.5 Endgame screen:

After the completion of the game this screen will show which player won the game and does the player want a rematch.



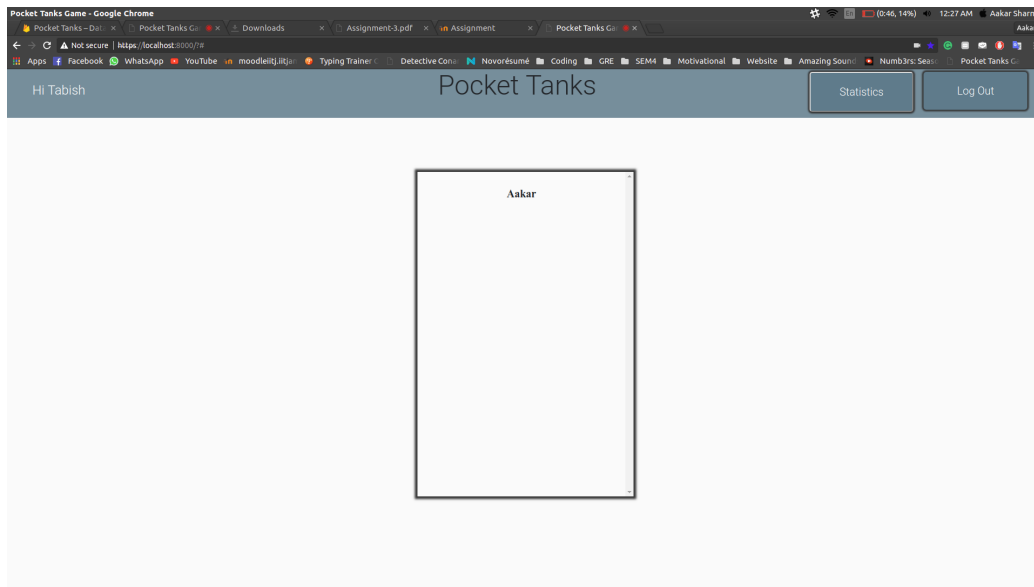
3.6 Statistics Screen:

Statistics of all kind of games, both between the computer and other players.



3.7 Multiplayer Screen:

This screen is used to show all the players online so that the player can send a connection request to the desired user.



4 Functions:

The code base will consist of many functions on both the client side and the server side.

4.1 draw terrain():

This function takes input from the generate terrain function and draws a fully random terrain. The random terrain is drawn using CreateJS and Canvas. The terrain is of default color green and the terrain will be updated only when the weapon damages the weapon.

4.2 draw tank1():

This function draws the tank for player one on the left side of the screen with default settings such as angle.

4.3 draw tank2():

This function draws the tank for player two on the right side of the screen with default settings such as angle.

4.4 draw controls():

This function draws the whole control panel for each player for controls such as power, angle, fire button, move button, chat button, weapon bag. The control panel will be grey in color and a slider for weapon selection will be provided. Each individual player can move only 4 paces to the left or right. On pressing the chat button a chat window will appear on the right side of the screen which will implement basic chat function. The controls will be disabled when it is not the current players turn. This will be controlled by a variable in the structure of the player named turn which will be set to false when it is not the current players turn and will be set to true when it is his turn.

4.5 show statistics():

This function shows the player name and score of the player. The statistics of player one will be shown on the top left corner of the screen and the statistics of player 2 will be shown on the top right corner of the screen.

4.6 draw weapon1():

This function draws the trajectory for weapon of player one.

4.7 draw weapon2():

This function draws the trajectory for weapon of player two.

4.8 draw homie():

This function draws the first page of the application ie. the homescreen which includes the play button which on clicking by the user will redirect to the profile in which the player will signup or login.

4.9 draw chat():

This function draws the chat div to the right hand side of the game play. We have implemented a basic chat room in which a player can send and receive messages.

4.10 draw options():

This function is used to draw the options page ie. the player wants to play in single player mode or multiplayer mode. We have used two methods for authentication. We use firebase as a database to store each player statistics such as number of matches won, number of losses and total score. We have also stored the rank of each player among all the users registered for our game. If the account of player already exists we call:

4.11 draw login():

This function is used to login the player and retrieve the basic statistics of the player so that he can begin playing the game. If the user's account does not exist: we call the draw signup() function: which takes in the basic details of the player and creates a game profile of player.

4.12 draw endgame():

This function shows which player has won and does he want to have a re-match.

4.13 make pairkey():

This function asks the person who is hosting the game to enter a pairing key which the other player will enter to join the game and host the game.

4.14 enter pairkey():

This function requests the second player to enter the pairing key entered by the player hosting the game and then join the game.

4.15 send terrain():

This is a function which send the details of the terrain (if their is a change) to the client side. This details include the height and the damages in the terrain. This terrain is updated on both side of the players as it is emitted using the socket.

4.16 send player data():

This is a function which send the player data like scores or the position of tanks or the angle of the barrel. This data is used by the functions defined on the client side (draw tank1, draw tank2 , etc) to update the data in real time. This player data will be send only when there is any change in the player data.

4.17 multiplayer():

This function is used to set up a multiplayer game between two player. This helps in authenticating the player and establishing a secure connection between them.

5 Java Script Functions:

create game():

This function is used to create a new game with reseting the scores and recreating the terrain. This function is called whenever a new game is created or a player choose to play again.

5.1 generate terrain():

This function generate a random terrain and place the tanks on this randomly generated terrain. The state of this randomly generated terrain is stored in a global variable terrain which remains same for a single game. However this randomly terrain can be destroyed by the weapons.

5.2 update terrain():

This function is used to update the terrain whenever a weapon hits the terrain. The damage occurred to the terrain depends on the type of weapons. This change in the terrain is shown to both the players simultaneously.

5.3 computer():

This is a function which will be called when user select to play with the computer. The functionality of the computer is implemented using the physics. The computer analyse the position of the player and then computes the angle and power of the shot accordingly

5.4 auth():

This function check if the pair key entered by the second player is same as created by the first player or not. If the key is same it joins the second player on the same network otherwise not.

5.5 playgame():

This function is called whenever a game is played. This function is responsible of allowing turn by turn game. It freezes the control of the other player, if one player is playing the game.

5.6 projectile motion():

This is a function which decides the weapons.