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.

2 Graphical User Interface:

2.1 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.

2.2 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.

2.3 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.

2.4 Start gameplay screen:

This screen contains the start button which will call the main function and start the game.

2.5 Gameplay screen:

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

2.6 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 Functions:

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

3.1 draw_terrain():

This functions takes input from 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.

3.2 draw_tank1():

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

3.3 draw_tank2():

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

3.4 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.

3.5 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.

3.6 draw_weapon1():

This function draws the trajectory for weapon of player one.

3.7 draw_weapon2():

This function draws the trajectory for weapon of player two.

3.8 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.

3.9 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.

3.10 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 loses 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:

3.11 draw_login():

This function is used to login the player and retrieve the basic statistics of the player so that he can began playing the game. If the users account does not exists : we call the draw_signup() function: which takes in the basic details of the player and creates a game profile of player.

3.12 draw_endgame():

This function shows which player has won and does he want to have a rematch.

3.13 make_pairkey():

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

3.14 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.

3.15 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 emited using the socket.

3.16 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.

3.17 muntiplayer()

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

4 Java Script Functions:

4.1 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.

4.2 generate_terrain()

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

4.3 update_terrain()

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

4.4 computer()

This is a function which will be called when user select to play with the computer. The functnality 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.

4.5 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.

4.6 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.

4.7 projectile_motion()

This is a function which decides the weapons.

5 Global Varibles:

terrain- stores the state of the terrain.
player1Data- stores the details of the player1.
player2Data- stores the details of the player2. (Either computer or other player)