# Assignment 3
# POCKET TANKS
# COP 290
# Design Document

May 8, 2018


| | |
|---|---|
| Date Performed: | April 15, 2018 |
| Group Partners: | Atul Rai (2016CSJ0074) |
| | Dwij Upadhyay (2016CSJ0009) |
| Instructor: | Professor Huzur Saran |



**Abstract**

This Document is an overview of the design that we are going to implement in our project. As this is an open project and we are free to build our own version of game, we are going to implement many features in our game as being descripted further in the document.


## 1 Overview

In this assignment, you are supposed to create a web based Pocket Tanks game. We have divided the development of this project in three parts:

a. Client Side of the game

b. Server Side of the game

c. Adding Chatting Option in to the game

Tools to be used: JavaScript (ECMAScript 6), Angular JS, Create JS, Node JS, HTML5 Canvas, CSS3, Firebase and Sockets.io.

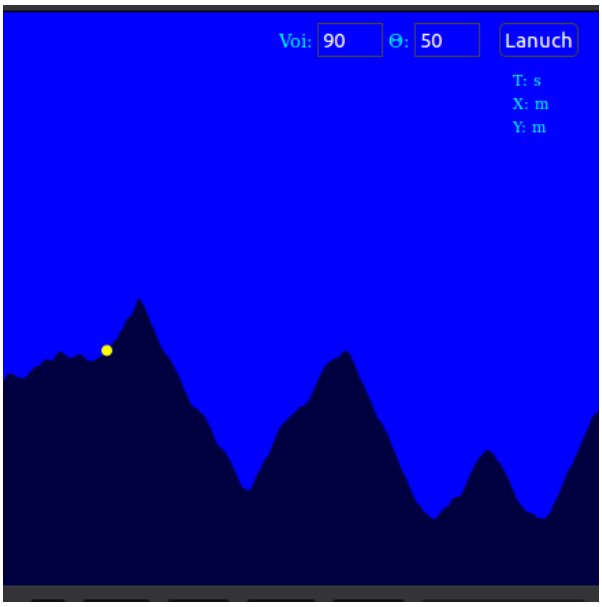



Figure 1: Prototype of the Game

## 1.1 Offline Game

a. The game starts off with a two tanks placed in the two halves of the screen over a randomly generated terrain. Note that while the terrain is generated randomly in each game, it remains same throughout the game. The upper left half of the screen will contain or show the data related to the Player 1 and the upper right half of the screen will show the data of Player 2/ CPU Player (If playing against CPU). Data shown on the screen of a particular player includes the name of player, his scores, two range bars to set speed and angle of projection of the projectile.The basic functionalities include:

- Movement of tanks in either direction..
- Selection of weapons
- Setting an angle for the tank nozzle
- Adjusting power on a scale of 1 to 100
- Firing of weapons

The functionalities that we are going to implement are:

- A dropdown for selecting weapons. When you will hover over the Weapons tab a dropdown will appear containing list of all weapons.

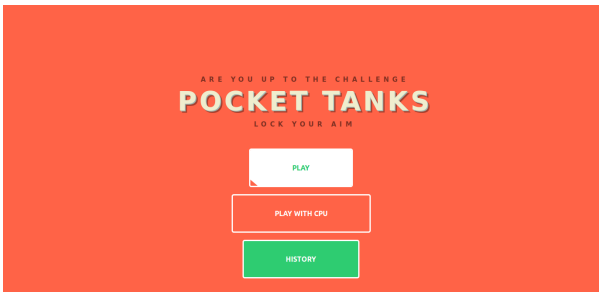



Figure 2: Home Screen

- Our GUI will have a chat window which can be minimised or maximised as required.
- We will also be having a exit button which will leave the game for us and will bring us to dashboard.
- We will be storing the scores and results of player's last three games.
- We will also implement a CPU Player.

```
1
2     TerrainShape.graphics.beginStroke("rgb(0, 150, 0)");
3     TerrainShape.lineWidth = 1;
4     for (let x = 0; x < max_w; x++) {
5         randdiff = rand(0,10);
6         TerrainShape.graphics.moveTo(x, h);
7         TerrainShape.graphics.lineTo(x, h - TerainPath[x]-
      randdiff);
8     }
```

Listing 1: Storing co-ordinates of Terrain

b. Following are the functions used for Controls:

```
1   start();  //Function to be triggered after Fire Button is
      pressed
2     go1();     // Forward Movement of Tank1
3     go2();     // Forward Movement of Tank2
4     back1();   // Backward Movement of Tank1
5     back2();   // Backward Movement of Tank1
```

Listing 2: Controls of Buttons on the Screen

We have our driver function named startGame();

```
1   function startGame(){
2       GenerateTerrain();
3       stage.update();
4   }
```

Listing 3: Triggers first and generate the Scene

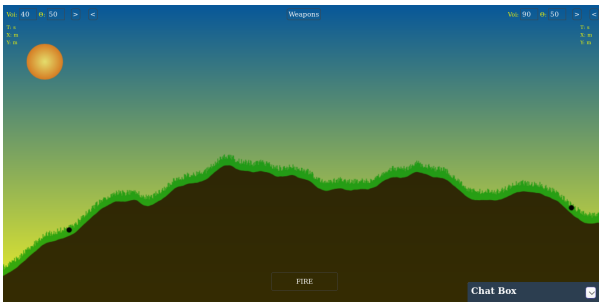



Figure 3: GUI of the Game

In GenerateTerrain() function, terrain is being generated and random positions on the terrain are assigned to both the tanks.

c. Information about the weapons used is given below:

   (a) **Sniper rifle**
   - 100 points if connected correctly; 0 if missed even slightly.

   (b) **Gamma Blaster**
   - A Way to play offensive even if caugt in a dirtball.How? This weapon doesn't get affected by anything it just makes it way through .It shoots lots of particles in a fan type way at the angle provided and then converge if the opponent tank is detected dealing a great damage.
   - Fire it after creating a dirt ball around yourself at last to create the best result out of this combo.
   - Shoot it as slow as possible to make all particles to hit.

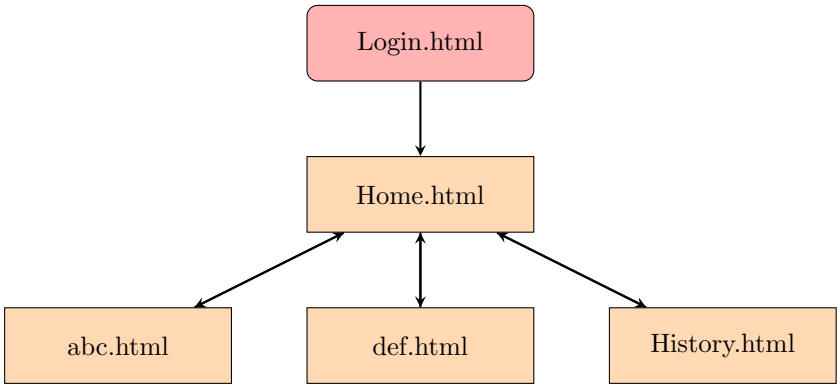
   (c) **3 shots**
   - Shoots three particles at different angle from the selected angle and then creates blasts when touch the ground and deals damage.
   - Shoot such that tank comes between any of two particles so it gets hit by both of them to deal max damage.

   (d) **One shot**
   - Shoots particle at selected angle and then creates blasts when touch the ground and deals damage.

d. For the above work we will be completely using HTML Canvas, Angular JS and Javascript. Animation and Interactivity in the page will be done using CreateJS and angular JS.For the Physics used in the project we will refer online to here.

e. Our base file is login.html which is for logging in to the game. After submission of login form we will be redirected to Home Page/ DashBoard of the game. It will have three buttons:

- Play : This button redirects us to our game where two players can play their game side by side.
- Play With CPU : This button will redirect us to our game where the second player i.e player against us will be CPU.
- History : This button will redirect us to our history of the game. It includes the details of last three games played each against CPU and against any other logged in player.




## 1.2 Server Side

We will be setting the server using Node.Js. Node.js gets more popular by the day.We will create a server side that listens for players wanting to join your game, keeps track of statistics in matches played over last 3 sessions, works as a mediator for the actions of the two players to be in sync. (There swill be login authentication for players when they join the game. This will help in tracking their history).
We will be most probably using Firebase as our database. Firebase has its google account authentication in it whic we are planning to make use of.
For joining a game there can be many ways of implementation e.g. we have thought of using a screen which will prompt of either joining a game or creating a game.

## 1.3 Chat System

At last we will implement/integrate the chat system into our project. For this we will be using Angular Js to implement its view. We will be using Socket.io to implement this funtionality of chat into our assignment.

## 2 Physics

a. **General projectile motion of particle in presense of gravity:-**

- The motion of an object under the influence of gravity is determined completely by the acceleration of gravity, its launch speed, and launch angle provided air friction is negligible.
- The horizontal and vertical motions may be separated and described by the general motion equations for constant acceleration. The initial vector components of the velocity are used in the equations.
- Note that the 60 and 30 degree trajectories have the same range, as do any pair of launches at complementary angles. The launch at 45 degrees gives the maximum range.
- **The horizontal motion:-**

$$a_x = 0 \tag{1}$$

$$v_x = v_{0x} \tag{2}$$

$$x = v_{0x}t \tag{3}$$

- **The vertical motion :-**

$$a_y = -g \tag{4}$$

$$v_y = v_{0y} - gt \tag{5}$$

$$y = v_{0y}t - {}^{1}/_{2}gt^2 \tag{6}$$

- **Time Of Flight:-** The basic motion eqn is given by

$$h = v_{0y}t - \frac{1}{2}gt^2 \tag{7}$$

Can be used to find the time of flight at height h given by

$$t = \frac{v_{0y}}{g} + -\sqrt{\frac{v_{0y}^2}{g^2} - \frac{2h}{g}} \tag{8}$$

By replacing h=0 we get the total time of flight $t = 0$ and $t = \frac{2v_{0y}}{g}$
$t_{peak} = \frac{v_{0y}}{g}$ is the time taken to reach peak

- **Range Calculation:-**

$$R = \frac{v_0^2 \sin 2\theta}{g} \tag{9}$$

- **Height Calculation:-** Max heigt of a given initial velocity and angle of projectile can be given by

$$h = y_{peak} = \frac{v_{0y}^2}{2g} \tag{10}$$