Design Document
COL290 : DESIGN PRACTICES
Assignment 3: POCKET TANKS

April 16, 2018

Vikas Gola(2016CSJ0023)
Deepak Rai(2016CSJ0021)

1. Overall Design

In this project, we made a game namely **Pocket Tank**. The game has been made in html, css and javascript. We used different libraries which are Createjs, Nodejs, Angular and SocketIO.

2. Implementaion of Specifications

We have divided the game in different parts and explained as the division is. First, we describe the part and then we described how we implemented that part. The implementation of the game is as follows:

Tanks

The tanks have different textures and multiple type of weapons which can be fired and changed during gameplay. The set of weapons have to be select before start of game and the player will be able to select from that during gameplay. The tank will also be able to move on the terrain.

- **Tank**
  Tank has different parts which are main body, nozzle and wheels. All parts of tank have been made using image. Images have been load in canvas using **CreateJS** library. We change the position of tank by changing positions of image and angle of nozzle by rotating the nozzle image.
- **Weapons**
  We draw weapons using the simple draw functions provided in CreateJS and change there position according to action by user.
- **Motion**
  We handle the motion of tank by checking the position of terrian and then accordingly move the tank. To do this, We give the some defined power to tank with which it moves and then we applied the mechanics with it position and terrain angle.

Physics

We applied different physics on the different weapons on the basis of weapon's features, however there main motion is projectile and we applied the projectile physics which is explained here as follows:

- **Projectile Motion**
  Let the projectile be launched with an initial velocity $\mathbf{v}(0) \equiv \mathbf{v}_0$, which can be expressed as the sum of horizontal and vertical components as follows:
  $\mathbf{v}_0 = v_{0x}\mathbf{i} + v_{0y}\mathbf{j}$.

  The components v_{0x} and v_{0y} can be found if the initial launch angle, θ , is known:

  $$v_{0x} = v_0 \cos\theta, \qquad v_{0y} = v_0 \sin\theta.$$

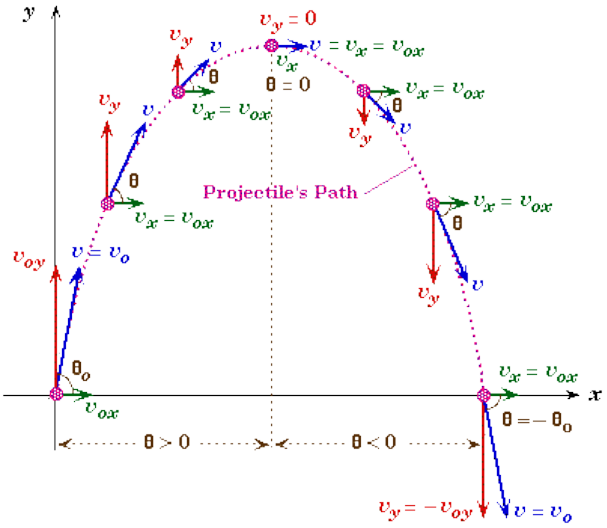

Figure 1: Projectile Motion

Velocity: The horizontal component of the velocity of the object remains unchanged throughout the motion. The downward vertical component of the velocity increases linearly, because the acceleration due to gravity is constant. The accelerations in the x and y directions can be integrated to solve for the components of velocity at any time t, as follows:

$$v_x = v_0 \cos(\theta),$$
$$v_y = v_0 \sin(\theta) - gt.$$

The magnitude of the velocity (under the Pythagorean theorem, also known as the triangle law):

$$v = \sqrt{v_x^2 + v_y^2}\,.$$

Displacement: At any time t t, the projectile's horizontal and vertical displacement are :

$$x = v_0 t \cos(\theta),$$
$$y = v_0 t \sin(\theta) - \frac{1}{2} g t^2.$$

- **Inclined Plane**

Fn = N = Normal force that is perpendicular to the plane,
Fi = f = input force,
Fw = mg = weight of the load, where m = mass, g = gravity

A load resting on an inclined plane, when considered as a free body has three forces acting on it:
1. The applied force, Fi exerted on the load to move it, which acts parallel to the inclined plane.
2. The weight of the load, Fw, which acts vertically downwards
3. The force of the plane on the load. This can be resolved into two components: First:-The normal force Fn of the inclined plane on the load, supporting it. This is directed perpendicular (normal) to the surface. Second:- The frictional force, Ff of the plane on the load acts parallel to the surface, and is always in a direction opposite to the motion of the object. It is equal to the normal force multiplied by the coefficient of static friction $\hat{I}ij$ between the two surfaces.

Using Newton's second law of motion the load will be stationary or in steady motion if the sum of the forces on it is zero. Since the direction of the frictional force is opposite for the case of uphill and downhill motion, these two cases must be considered separately:

Uphill motion: The total force on the load is toward the uphill side, so the frictional force is directed down the plane, opposing the input force. The mechanical advantage is

$$\mathrm{MA} = \frac{F_w}{F_i} = \frac{\cos\phi}{\sin(\theta + \phi)}\,,$$

where $\phi = \tan^{-1}\mu$,. This is the condition for impending motion up the inclined plane. If the applied force Fi is greater than given by this equation, the load will move up the plane.

Downhill motion: The total force on the load is toward the downhill side, so the frictional force is directed up the plane.
The mechanical advantage is

$$\mathrm{MA} = \frac{F_w}{F_i} = \frac{\cos\phi}{\sin(\theta - \phi)}$$

This is the condition for impending motion down the plane; if the applied force Fi is less than given in this equation, the load will slide down

the plane. There are three cases:
$\theta < \phi$: The mechanical advantage is negative. In the absence of applied force the load will remain motionless, and requires some negative (downhill) applied force to slide down.
$\theta = \phi$: The 'angle of repose'. The mechanical advantage is infinite. With no applied force, load will not slide, but the slightest negative (downhill) force will cause it to slide.
$\theta > \phi$: The mechanical advantage is positive. In the absence of applied force the load will slide down the plane, and requires some positive (uphill) force to hold it motionless

Terrain

The terrain will generate randomly each time and may contain different height and shapes. The terrain will have texture of mountain to give it realistic feel.

To implement terrain, we will make function which gives the random points on the canvas with some constraints so it looks like terrain and then we will connect these points to generate the terrain. We will apply some gradient color or shading on the terrain to make it realistic. The terrain destroy on the basis of weapon damage value which will be specified to each weapon. The damage cutted from terrain will be shown by changing the value of points at that place where weapon has been fired. There will be animation whenever damage happen to terrain to look it more realistic.

Menu Screen

At the start, the game have start menu where players can select the game mode in which they want to play. The game modes which will be provided are **single player mode, multiplayer mode**(on the same computer or phone) and **LAN multiplayer mode**(to be played on different phones or computers). The start menu contains options like sound on or off. To play LAN multiplayer mode, the one player has to send the invitation to another player with their login Id and the another one needs to accepts that invitation.

The main screen of the game will have a image or a video which will give overview or big idea of how game is, which will be shown on the screen by loading an image or video using the createjs. The modes buttons and mute on the screen will be shown as buttons which will also be an image.

Registration and Login Screen

This screen is what its name describes. Here, the person has to login to their account if he/she already has an account else he/she has to create a new account. The account contains information of users which are age, UserId, password, friends(with whom he/she played in past) and high score.

The registration and login part will be handled using SQL on the server. The data given by user will be sended to the server where nodejs code will handle the input and respond accordingly. If the user give wrong password/wrong id he/she will be informed by showing the text on the canvas which says 'wrong password/wrong id'. We also added a loading animation which will be shown at processing time.

Game Over Screen

This screen is last screen which will be displayed after the game finishes. This screen will have option of **Play Again** and **Exit**.

This screen will be draw as Registration and Login screen.

Play Screen

In this screen, the main game will be played where tanks and terrain part come together. This screen will contain information of players name, health, the number of consecutive wins of each player and also exit option from the game, pause option, angle of tank, weapons change option and score of each player.

This screen contains a terrain, two tanks which we have already explained and the some controls buttons. The angle changing will be draw using the createjs and the changes the user does like moving tank, chat and weapon changing will be send to server where nodejs will handle it and send it to another player(in the Lan mutiplaye mode and for the single player mode it just not needed). Angular will be used to make game dynamic so that it can reflect all changes to another player in real time. The background sound will be added using SoundJS.

Keyboard Controls

The game will have controls with keyboard for changing the angle of attack for weapons, fire, power and moving the tanks. The other controls from keyboard will be pause, exit and chat.

The controls of keyboard will added using the event listner on objects(tanks) and buttons for different activites.

Video, Text and Audio chat

The game will have options to chat. Player can message, video chat and audio chat which can be accessed from the screen.

The chat section will be implemented using the SOCKETIO, CreateJS and NodeJS mixture. The chat section will be open on the call from screen button on from keyboard. The main part of this section will be handled by SOCKETIO.

Smart Player

The smart player implemented using the strategy where the opponent is. Single player will have three levels of difficulties(easy, intermediate and hard).

In easy mode, the computer will attack with random weapon and with accuracy 40% which will be attain by the combination of velocity and angle.

In intermediate mode, the computer will attack with little brain like which weapon can destroy the other tank early or the other tank make not to move so it becomes difficult to move. The accuracy of this mode will have 60%.

In hard mode, the computer will attack with smartest weapons and with best accuracy of 90%.