# Course Project Documentation

## CS101 Project

# POCKET TANKS

**“The ultimate one on one artillery game”**

Shrey Gadiya, 14D070006
Ravi Sharma, 14D070055
Prabhav Mohta, 140040086
Ramswaroop Kumawat, 140040065

# Table Of Contents

# 1. Introduction:

Pocket Tanks is a fast-paced artillery game that is easy to learn and fun to master. You take turns with your opponent in firing shots. Each hit scores a certain amount of points, which varies based on the proximity of the shot. The player with the maximum number of points at the end of 3 turns wins.

The game features a fully destructible terrain causing the change in the position of the tanks thus allowing the probability of strategic play.

## 2. Problem Statement

Our goal was to design a simple yet entertaining game involving two player known as Pocket Tanks using OpenGL.

We aimed to create a game where a player has to judge the power and angle according to the distance between the tanks.

To generate a randomized terrain so as to increase the difficulty of the game.

Evaluating and displaying the scores after each turn according to the relative position of the opposing tank and the point where the shot touches the terrain.

# 3. System Requirements:

1. OpenGL - For a graphical interface.

2. Code Blocks - For writing the code and building the project.

3. Git Hub - To manage and update the project.

# 4. Implementation and Functionality:

1. **Creating the terrain-** The terrain was created by dividing the OpenGL window into 20 different polygons and randomizing the height of each new polygon. This creates a new and random terrain each time the game is played.

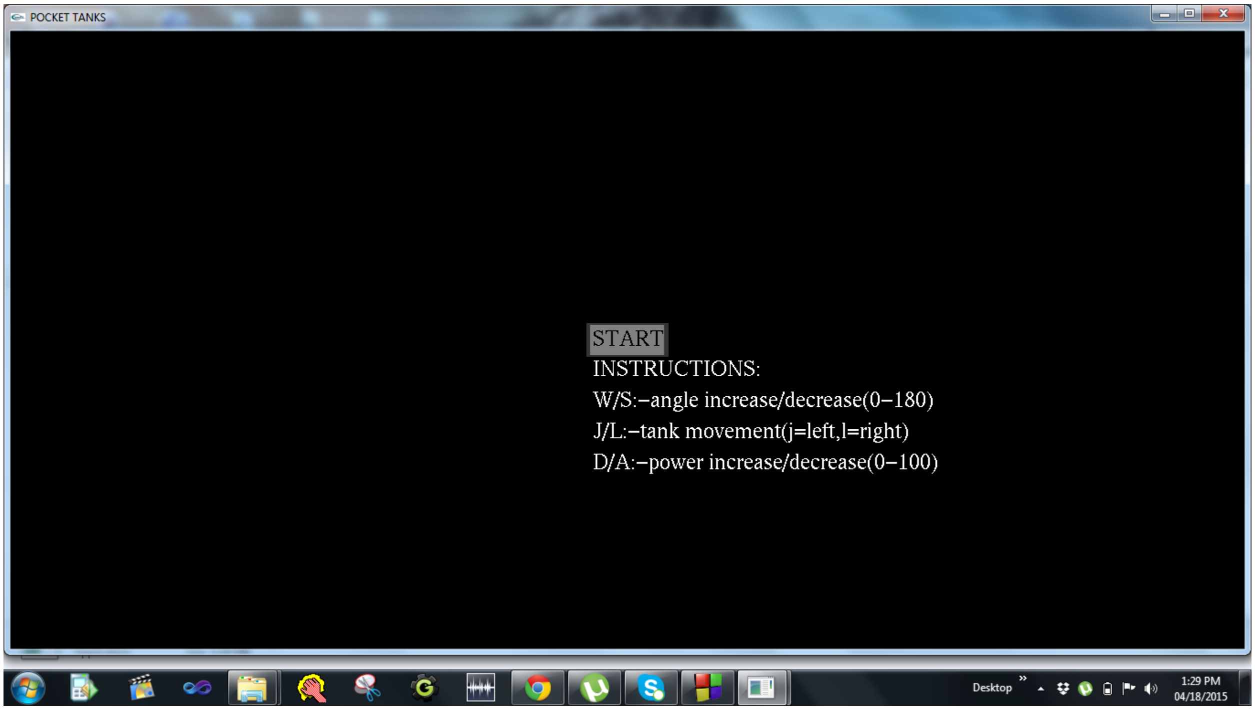
2. **Scoring the shots fired-** Scores are given based on the proximity of the point where the projectile touches the tank or the terrain. Since distance in OpenGL is measured from -1 to +1, if the point of contact is within 0.3 of the tank, a score of ((0.3-distance)/.3)*100 is awarded to the attacker. Otherwise the score doesn't increase.

3. **Power and angle-** There are two bar on the OpenGL screen representing these quantities. The power bar ranges from 0 to 100 and the angle bar varies from 0 The user can toggle these bars using keys on the keyboard. We did this using the glutKeyboard Func.

4. **Movement of tanks-** The tank can be moved on their half of the screen. Their position changes to the vertex of the nearest polygon (of those 20 used to make the terrain) on either side. This function is used on the keyboard using the glutKeyboard Func.

# 5. Testing Strategy and Data:

On Running the program, an OpenGL screen is generated (as shown below) displaying the controls of the game and with the option to start the game.



START

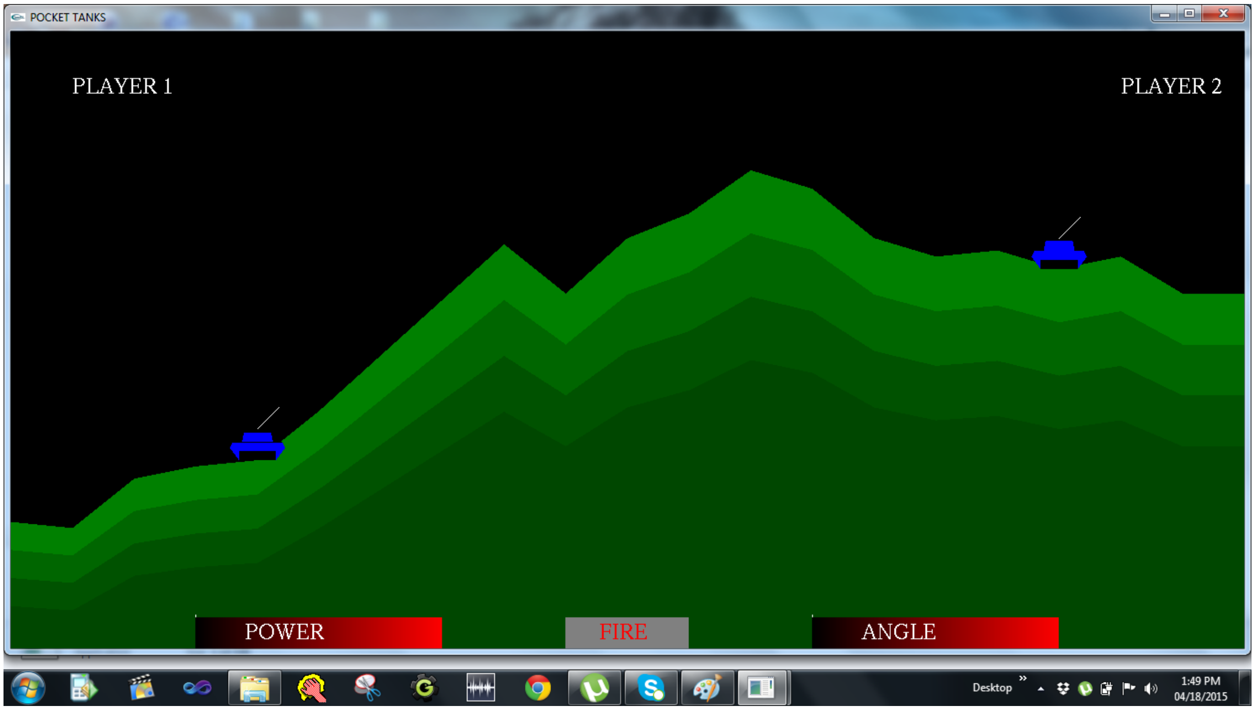
INSTRUCTIONS:

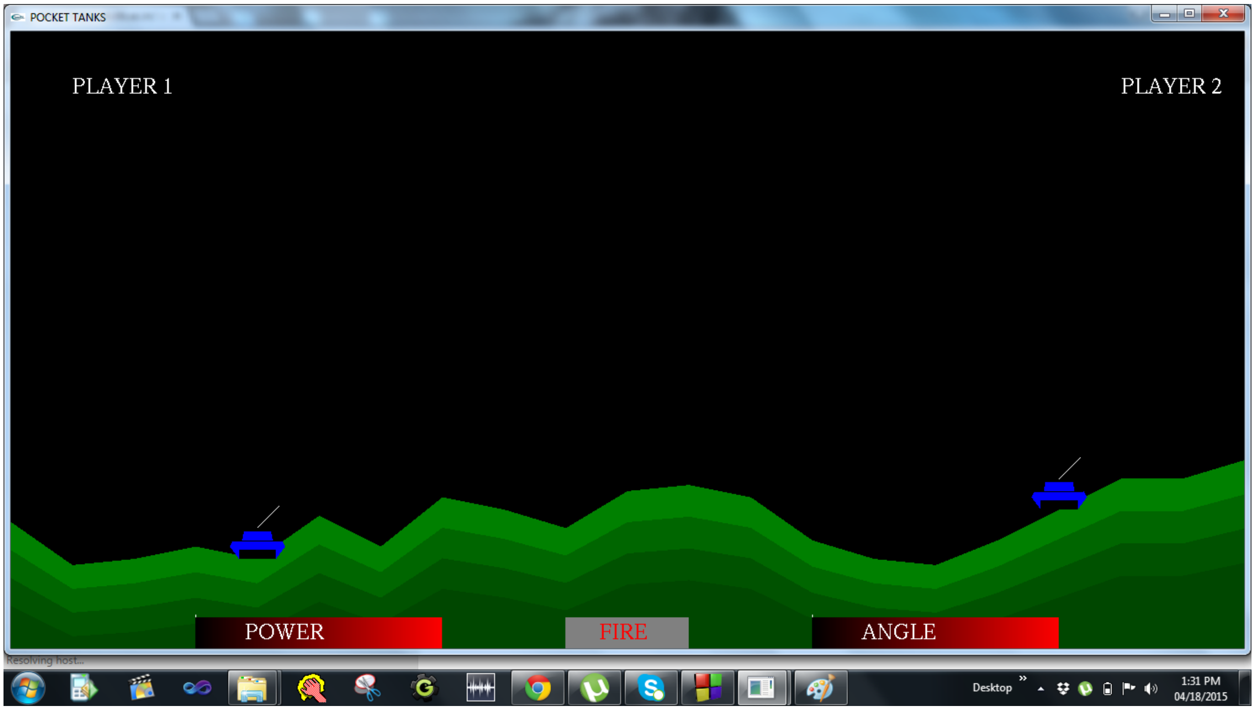
W/S:−angle increase/decrease(0−180)

J/L:−tank movement(j=left,l=right)

D/A:−power increase/decrease(0−100)

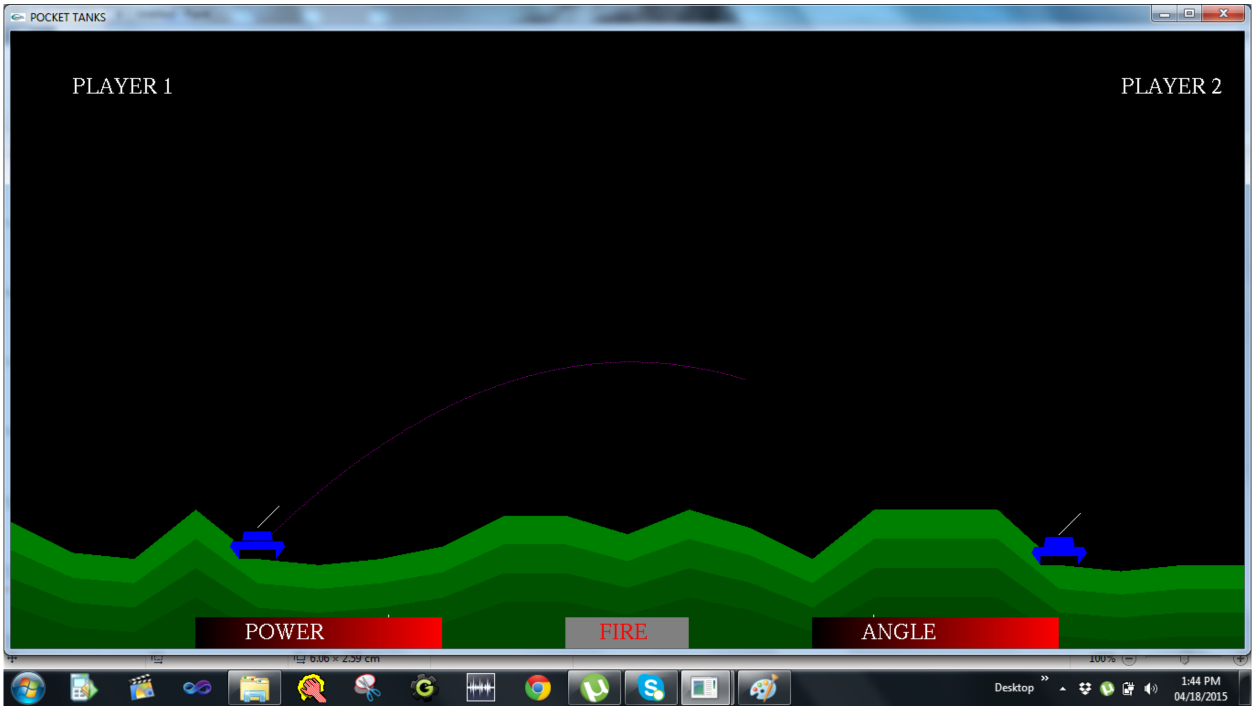
On Clicking on the ‘start game’ icon, the game screen is generated dispaying a random terrain everytime, with the tanks place randomly.



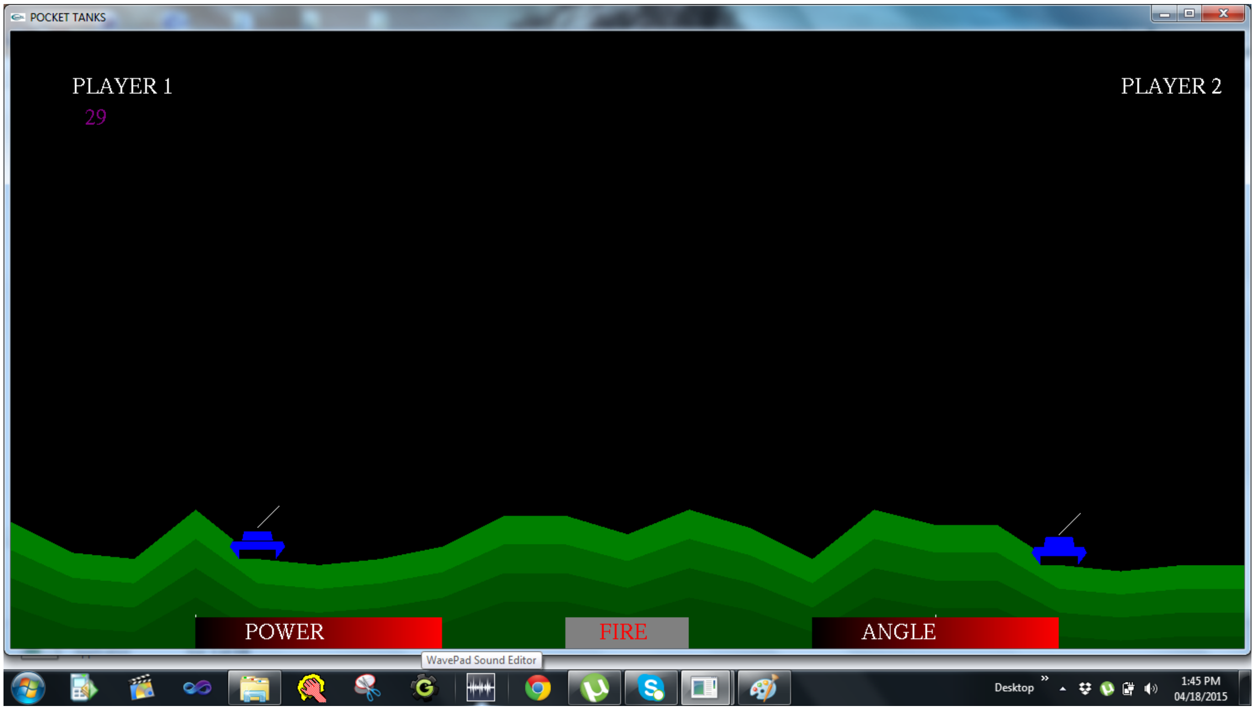


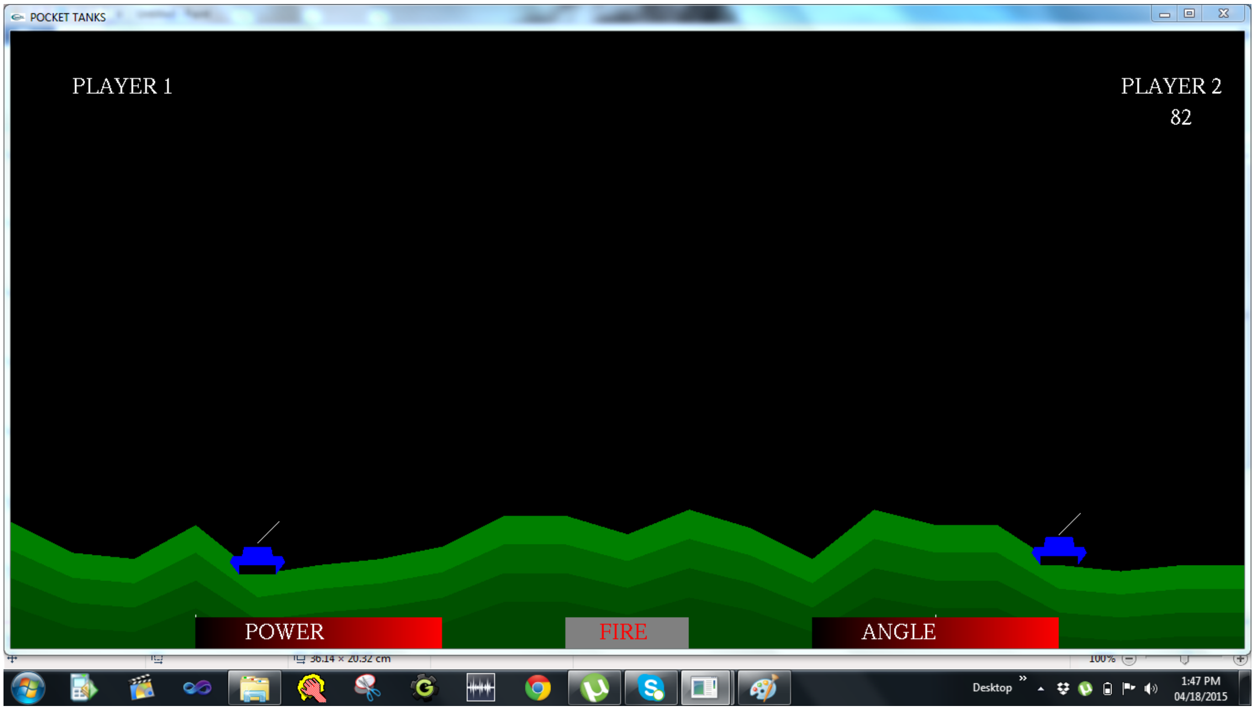
The screen also shows the the toggle for the power and the angle and power. Also it has the fire icon. On clicking on the icon after setting the power and the angle a projectile is fired

.

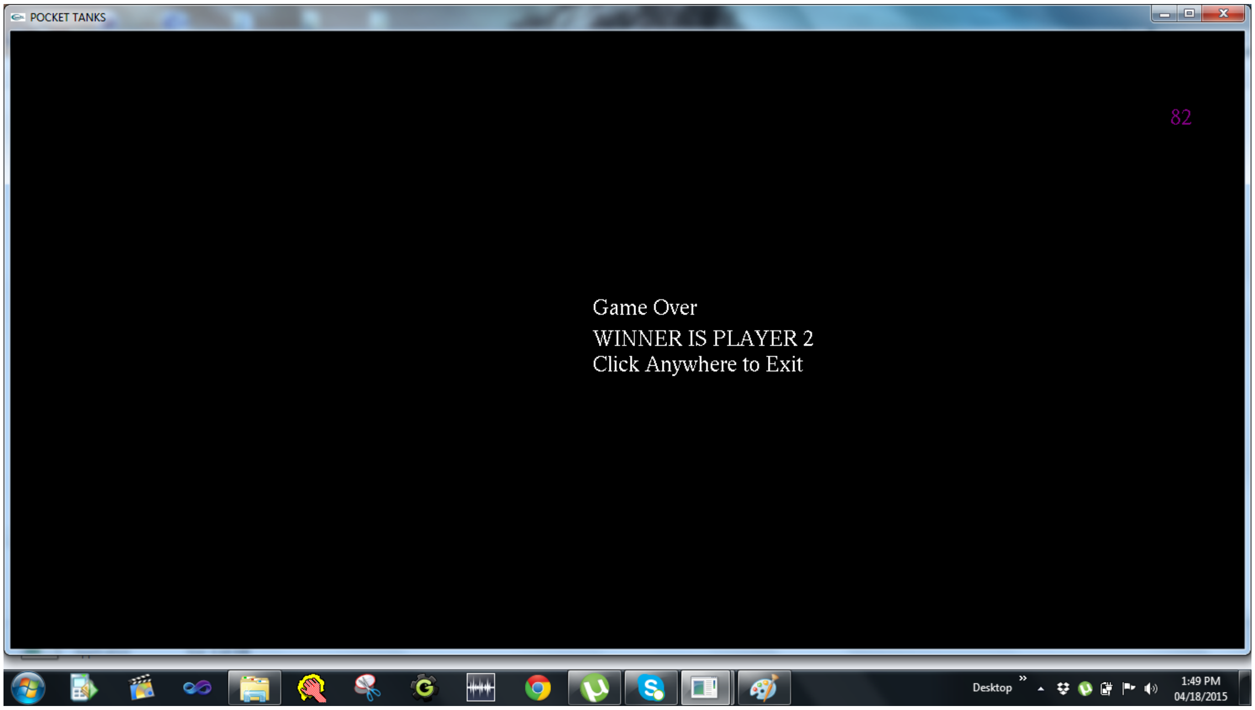


Depending on where the projectile lands on the terrain, the score increases and it is inversely proportional to the distance between the center of the tank and the point of contact of the shot, The turn changes after each shot is fired.





At the end of three turns each, the player with the highest points wins and the endscreen is displayed, The screen is such that if the user clicks anywhere, the game will exit.



Game Over
WINNER IS PLAYER 2
Click Anywhere to Exit

# 6. Discussion of System

1. **Creating a random terrain:** We divided the OpenGL window into 20 parts and created a trapezium in each part while randomizing their height. Thus a new terrain is developed every time the game is played making it more interesting.
2. **Ending the projectile when it touches the terrain:** We were constantly calculating the x and y coordinate of the projectile. Based on the x coordinate we calculated the slope of the trapezium (used to make the terrain) it was over. Using this, as soon as the y coordinate of the projectile became lesser than the one calculated using the slope, we terminated the projectile.
3. **Score calculation:** The score awarded each time the projectile hit was inversely proportional to the distance between the center of the tank and the end point of the projectile. We set a blast radius in which points were awarded and the score was calculated using the formula 100*(r-d)/r.
4. **Display:** We displayed the dynamic scores and the toggles for power and angle using the OpenGL Text Libraries.

# 7. Future work:

1. As of now we only have a two player game which can further be increased to involve more people and improve the strategic aspect of the game. We can also introduce a single player mode which will be played against an AI (Artificial Intelligence).
2. The creation of new types of weapons and new types of maps partial to some weapons can be made. Eg. A map with water which could make attacks related to fire useless.
3. We can connect the game to servers through LAN or the internet to allow players everywhere to play against each other.

.

# 8. Conclusion:

Hence we have successfully created a simple entertaining game, Pocket Tanks, which had been well tested on Windows 7.