

BATTLE TANKS

Contents

Introduction

System Requirements

Game Basics

Working

Member variables

Header Files Used

Credits

Introduction

This game is a modified version of the game pocket tanks. The objective of the game is to destroy the opponent tank, i.e. Reduce the health of the opponent to zero. The game provides choices for weapons and tanks. The various tanks has various statistics of attack and defence. The game also facilitates the user to select a variety of weapons which have different strength, weakness and accuracy.

System Requirements

- GCC Compiler
- EzWindows
- 20MB of HardDisk space
- 2 Mb Virtual Memory
- Ubuntu 9.4(or above)

Game Basics

Game Basics are simple: select your tank type, buy desired

weapons and play. The basic objective is to hit the enemy tank as close as possible. Estimate power and angle to do that and destroy the enemy tank to win.

Working

The game is designed in c++ making use of the EzWindows library for graphics. The functions and variables are stored in different header files which will be included in the main function of the game. The game starts with the main gamescreen which gives options for different modes of the game and to select options for sound settings. The main outline of the game is as follows.

1. Project skeleton

- Home Screen
 - 1 Player
 - 2 Players
 - High Scores
 - Exit
- 1 Player
 - Select difficulty level
 - Select tank
 - Weapons Shop
 - Back to home screen
- 2 Players
 - Select tank
 - Weapons Shop
 - Back to home screen
- Main game screen
- Result

- Score = Remaining Health of Winner
- End game menu
 - Play Again
 - Back to Home Screen
 - Exit

2. Main gameplay contents

- Decide basic look of game screen
- Input: Select
 - Weapon
 - Power
 - Angle
- Programming for projectile trajectory
- Animation of projectile
- Calculating damage from particular weapon function
- Deducting damage from health
- End Game when health = 0

3.1 Player Mode

- Computer randomly chooses tank and weapons
- Randomising Computer accuracy according to difficulty level

4. 2 Player Mode

- Alternate turns

5. Tank types

- Offence (Attack 1.2x Armour 0.8x)
- Defence (Attack 0.8x Armour 1.2x)
- Normal (Attack 1.0x Armour 1.0x)

6. Terrain

A random terrain will be generated everytime the game starts . The terreaain is based in a continuous mathematical function.

7. Weapon Shop

- Different weapons availabe
- The player chooses 5 Weapons of his choice

Algorithms

Terrain Generation

- Array to store height with index = x-coordinate*10
- Height[0] is randomly generated
- Height[k] is randomly generated between (Height[k-0.01] – 0.1) and (Height[k-0.01] + 0.1)
- Space below Height array is filled with green rectangles

Trajectory Generation

- Input of Power and Angle taken from user Vertical and horizontal velocities determined from power and angle
- X and y coordinates at different time generated using basic Newton's laws of motion
- Trajectory of projectile generated accordingly

Damage Generation

- Distance between projectile and target tank taken as input
- Damage calculated by a function of input
- This function is particular to the weapon used
- Damage modified according to the tank type(i.e. offensive, defensive, normal)

- Damage deducted from health of tank

Member Functions

1. `void ClearScreen();`

Clears the screen by erasing all bitmap images.

2. `void DoublePlayerScreen();`

This function allows both players to select different types of tanks.

Global variables used : ScreenID, Height, Width, PhotoPosition;

3. `void GameMenuScreen();`

This is the main menu of the game where user can select single player mode, double player mode, instructions, high scores or exit.

Global variables used : ScreenID, Height, Width;

4. `int MouseClickEvent(const Position &MousePosition);`

Whenever mouse is clicked this function is called. It handles all mouse click events.

Global variables used : MouseClick, NumberOfPlayers, Width, Height, Tank1Attack, Difficulty, TerrainHeight, Angle, Power,;

5. `void SinglePlayerScreen();`

Player selects type of tank and difficulty level after choosing the Single Player mode.

Global variables used : Width, Height, ScreenID;

6. `void TerrainGenerator();`

Generates random terrain.

Global variables used :TerrainHeight, Height, Width;

7. void WeaponShop(int,int);

Generates weapon shop where players can select weapons.

Global variables used :Count, Height, Width;

8. void WeaponSelection();

This function lets players to select weapons and stores the selected weapons in array.

Global variables used :Width, Height, ScreenID, Weapon1Chance[], Weapon2Chance[], Player1Weapons[], Player2Weapons[], Count;

9. void GamePlayScreen();

This loads the screen where game is played.

Global variables used :Tank1Health, Tank2Health;

10. void SelectWeapon();

User can select which weapon to fire.

Global variables used :PlayerNumber,Width, Player1Weapons[], Player2Weapons[], SelectedWeapon, Height;

11. void CpuSelectWeapon();

This function randomly selects weapons for CPU.

This happens in Single Player mode where the player plays against CPU

Global variables used :CpuChance[], SelectedWeapon, CpuWeapons[];

12. void CpuInput();

This function randomly selects Angle and Power for

generating artificial trajectory for CPU in Single Player mode.

Global variables used :Angle, Power, TerrainHeight[], Width, Difficulty;

13. void BmpMove(Bitmap &Image,const Position &Start,const Position &End);

This is used to move the buttons on game menu screen.

14. void wait (float milliseconds);

This function is used to wait for specified time before executing next command.

15. void InstructionsScreen();

This shows the instructions screen.

Global variables used :Height, Width, ScreenID;

16. void GameLoad ();

Loads the terrain, tanks, weapons etc.

Global variables used :Width, TerrainHeight[], Height, Wind, PlayerNumber, Player1Weapons[], Player2Weapons[],Player1Chance[],Player1Chance[],CpuChance[],CpuWeapons[];

17. void FireWeapon();

This function is called when weapon is fired. It calls other functions based on the selected weapon.

Global variables used :Tank1Damage, Tank2Damage, Player1Weapons[], Player2Weapons[], SelectedWeapon, PlayerNumber, Player1Chance[], Player2Chance[],

18. void SingleShot();

This function simulates trajectory and calculates damage for weapon Single Shot.

Global variables used :Angle, Power, PlayerNumber, Width, Height, TerrainHeight[], Wind, Tank1Damage, Tank2Damage;

19. void ThreeShots();

This function simulates trajectory and calculates damage for weapon Three Shots.

Global variables used :Angle, Power, PlayerNumber, Width, TerrainHeight[], Wind, Tank1Damage, Tank2Damage;

20. void FiveShots();

This function simulates trajectory and calculates damage for weapon Five Shots.

Global variables used :Angle, Power, PlayerNumber, Width, TerrainHeight[], Wind, Tank1Damage, Tank2Damage;

21. void SpaceLaser();

This function simulates trajectory and calculates damage for weapon Space Laser.

Global variables used : Angle, Power, PlayerNumber, Width, TerrainHeight[], Wind, Tank1Damage, Tank2Damage;

22. void EarthMover();

This function simulates trajectory and calculates damage for weapon EarthMover.

Global variables used :Angle, Power, PlayerNumber, Width, Height, TerrainHeight[], Wind, Tank1Damage,

Tank2Damage;

23. void Cruiser();

This function simulates trajectory and calculates damage for weapon Cruiser.

Global variables used :Angle, Power, PlayerNumber, Width, Height, TerrainHeight[], Wind, Tank1Damage, Tank2Damage;

24. void SuperBullet();

This function simulates trajectory and calculates damage for weapon SuperBullet.

Global variables used :Difficulty,Angle, PlayerNumber, Width, TerrainHeight[], Wind, Tank1Damage, Tank2Damage;

25. void PotShots();

This function simulates trajectory and calculates damage for weapon PotShots.

Global variables used : :Angle, Power, PlayerNumber, Width, TerrainHeight[], Wind, Tank1Damage, Tank2Damage;

26. void MagicHeal();

This function increases health of player by 200units.

Global variables used :PlayerNumber, Tank1Damage, Tank2Damage;

27. void Damage();

Deducts damage done to the tanks.

Global variables used :Tank1Damage, Tank2Damage, Tank1Health, Tank2Health, Tank1Attack, Tank2Attack,

NumberOfPlayers, PlayerNumber ;

28. void HighScoresScreen() ;

This function loads High Scores screen.

Global variables used :ScreenID, HighScore[],Width, Height;

29. void CpuWeaponSelection() ;

This functions selects the weapon to be fired by CPU.

Global variables used :CpuWeapons[];

30. void GameOverScreen() ;

This function will be called when the game comes to an end, i.e. When the health of a player reaches zero.

Global variables used :NumberOfPlayers, Tank1Health, Tank2Health, HighScore[], Width, Height;

31.int SaveScore() ;

This function takes care of the input to the files. Every time the game is over this function is called which writes the score to a file called “high.txt”

Header Files Used

cstdio

cmath

cstring

cstdlib

sstream

cassert

iostream

Credits

CS101 Course Professor : D.B.Phatak

LabBatch TA : Pranay Bhatia

Senior TA: Harsh Pareek

Batch Members

ARUN VINAYAK U

DOPPALAPUDI CHANDRAKANTH

MULA REVANT BABU

ANIL KUMAR

JAYANTH

BHISHEK ANAND

GUNTI SIDDHARTHA

ASHISH LALAWAT

BAGADIA SAMEEP JAYESH

DEVENDRA SINGH CHAPLOT

DIKSHA

P V KRISHNA KOUNDINYA

SUGALI ASHOK NAIK

Special Thanks : Saswat Padhi

The game Pocket Tanks for the idea, tank image, terrain image

cplusplus.com, cppreference.com

Various websites for free images