BATTLE-ZONE

TEAM:

Himanshu Payal

MridulSayana

Prerak Garg

GROUP-02

SLOT- 11

TA- ArunBabu

Project for CS 101

ACKNOWLEDGEMENT:

In this project, the concepts taught by our professors Mr. Deepak Phatak and Mr.Supratik Chakraborty were very useful.

The help and advice provided by our T.A. Mr. ArunBabu also helped our team to solve many problems.

For completion and implementation of our project, the work of Mr. Abhiram G. Ranade on graphics in C++ was also helpful.

TABLE OF CONTENTS

1.INTRODUCTION

The game Battle-Zone is based on a very popular game 'Pocket Tanks'. In our project we have tried to create a similar version of the game of Pocket Tanks using C++ as our programming language.

2. BACKGROUND

The game Battle-Zone is based on the popular game of Pocket Tanks.

For using graphics in this C++ project we have made use of various libraries like simplecpp and we also used the work of Mr. Abhiram G. Ranade on graphics in C++.

HEADER FILES USED:

- iostream.h
- math.h
- simplecpp.h
- stdlib.h
- time.h

3. OBJECTIVE

To create a game as described in our user manual. A brief description of the game is provided below

4. DESCRIPTION

Battle-zone is a simple artillery game in which there are two tanks containing a variety of weapons, which the tanks fire at each other turn by turn to score points which form the basis of winning the match.

5. GAME MODES

It is a game which can be played in two player mode where each player fires weapons at theopponent from his/her tank. Based on how accurate the shot is, and the choice of weapon, player will be awarded points for each shot. After a fixed number of rounds (user defined, between 1 and 10), the player with more points will be declared as the winner.
In our next version of the game we will include various terrains too like hills and valleys.

6. WORKING TOWARDS OUR GOAL

The work was divided among the team members.
Mridul designed a method to get input from the users and display it on the canvas immediately and continuously. He also made the changes in our original code to ensure that user interacts via the canvas only and not the terminal window. Himanshu designed algorithms of several functions and wrote the code for them (e.g.algorithm for the functioning of the game, function to show the trajectory of the fired weapon, function to award points depending on the accuracy of the shot, etc.). Prerak also wrote some functions like pt_awrdr2 (for player 2) and trajectory2 (for player 2). He also helped in writing the function range(for calculating range of the fired shot).
All team members were involved in deciding the rules and working of the game.
We defined the following functions too in our code and made their judicious use for the working of our code:
1-void trajectory1(float angle1,float power1,float

windspeed,int x1,int x2,int k,int &pt1)
2-void trajectory2(float angle2,float power2,float windspeed,int x2,int x1,int k,int &pt2)
{to show the moving weapon on the canvas under the influence of gravity and wind}
3-void pt_awrdr1(int &pt1,int x1,int x2,float R,float angle1,float power1,float windspeed,int k)
4-void pt_awrdr2(int &pt2,int x1,int x2,float R,float angle2,float power2,float windspeed,int j)
{to award points to the player depending upon the accuracy of the shot}
5-float range(int start,float windspeed,float angle,float power,int turn)
{to find the range of fired shot}
6-void explosion(int x,int y,float r)
{to show explosion where the bomb hits the ground}
The relevant structure of these functions can be found in our SRS document as well as the project code.

7. PROBLEMS FACED:

1. Taking continuous input from the user for ‘Angle’ and ‘Power’ without having him/her to press enter key repeatedly:

The problem was solved by Mridul by using combination of header file named <cstdio>, while loop and getch command.

2. Showing trajectory of the fired shot on the screen:

Himanshu suggested a turtle method which can show approximate trajectory but the team is working on it to make it more precise. We searched a lot for header files and libraries in simple c++ which can show a moving object on a predefined trajectory. But we couldn’t find any such libraries compatible with the versions of code::blocks we have in our labs and laptops.

The problem was finally solved. Himanshu wrote the code for trajectory function using the video lectures that were put on our course website.

3.Aligning the turret of the tank along the angle given bythe user.

4. Resettingvarious elements of the code at proper positions in order to ensure proper functioning of the game.

5. Debugging the final code after combining all

functions, testing with each and every mode and weapons.

8. BIBLIOGRAPHY

Textbook used: An Introduction to programming using C++ by Mr. Abhiram G. Ranade.

Code::blocks manual with simplecpp was critical for understanding the use and working of simplecpp.

SELF EVALUATION MARKS:
MRIDUL-5
HIMANSHU-5
PRERAK-3