Project 2

<Spot a Tank>

CSC\_5\_40717

Oscar Mayorga

Dare: 02/2/15

***INTRODUCTION:***

Title: Spot a Tank

The game explains what it’s happening, in this case it output that the camp it’s under attack.

It gives you 15 opportunities or shots to spot as many tanks as you can.

The Game consist of targeting tanks outside your camp which is under attack. You will have to input the angle to where you want to shoot and also the power with which you want to hit the tank. The only information given about where the tank is located is the distance of where they are coming towards your camp.

The Number of targeted it tank would add up and after your 15 shots are use it would tell your score.

***SUMMARY***

SPOT A TANK is a game where you can show how much your brain can think and show some math skills. The math skills would be able to be proven when you enter the angle to where you want to shoot and the power you want the projectile to have for the trajectory.

The game not only tell you that you missed but it also tell you that the shot was too far or close.

***Description:***

This project includes most of the concepts we have learn in class from the chapters we have covered. This project have strong possibilities to be extended for next project. The project can be improve by adding more levels using less space and different loops would also be a way to improve the game for next project.

It took me about 6-7 days or from 18-24 hours to make it, if more time could have been use to build it, it would have given better results. I tried to add as many knowledge as possible. The game it’s very simple but it can be improved to be better and wroth to play and have fun.

|  |  |
| --- | --- |
| <Iostream> |  |
| <ctime> |  |
| <Iomanip> |  |
| <cstdlib> |  |
| <cmath> |  |
| <fstream> |  |
| <string> |  |
| void | highscores(int, int); input(int &, int &);hitDisp(bool, int, int); |
| bool | hitMiss(float, int, char); bool hitMiss(float dist, int t, char diff) |
| int |  |
| float |  |
| char |  |
| unsigned |  |
| srand | * srand(seed); |
| for |  |
| if |  |
| while |  |
| else if |  |
| else |  |
| do |  |
| Easyfile. & hardfilr. |  |
| play | (int&, int, int&, int, int&, char&); |
| string | name; |
| fstream | easyFile, hardFile; |

***No pseudocode,***

//seed random number generator

srand(seed);

do{ tnkPos = plcTnk(); //place tank

//Prompt for difficulty diff = getDiff();

//Reset scores and ammo score = 0;

ammo = 15;

for (int attmpt = 0; attmpt<15; attmpt++){ //Loop for tank destruction

play (tnkPos, diff, score, ammo, att, b); ammo--; //call main game play function

if (b == 1){ //Bonus shot rewarding system increment shots remaining message decrement for loop

ammo++;

attempt--; }

}

while(choice=='y'||choice=='Y'); //Ask user for replay

int getDiff(){ //function for selecting difficulty

while(chc!='e'&&chc!='E' && chc!='h'&&chc!='H'){ //Input

//Initialize difficulty by setting explosive blast radius

if(chc == 'e'||chc == 'E'){ //Difficulty chosen EASY

else if(chc == 'h'||chc == 'H'){ // Difficulty chosen HARD

}

//Function to place tank

int plcTnk(){ //Place the Tank int

int x;

x = 1 + rand() % 999; // Random distance of tank placement

}

//Function for user firing input

void input(int &a,int &p){ //Void the input

do{ cout<<"Input firing angle (0-90)"<<endl;

cin>>a; } //Prompt the user for inputs

while(a>90||a<0); //The number have to be between 0 and 90

do{ cout<<"Input projectile power in meters per second (1-100)"<<endl;

cin>>p; }// Input projectile power in meters per second

}

//Function calculating where projectile hits

int ballDst(int a, int p){ //The trajectory of the projectile

float d;

d=((p\*p)\*sin(2\*(a\*(PI/180))))/G; //Projectile distance according to inputs(Angle and power

return d; }

bool hitMiss(float dist, int t, char diff){ / /Function testing if projectile hit

if (dist > t-diff && dist < t+diff){ //test if projectiles blast radius hit tank

else //Projectile missed

void hitDisp(bool hit, int d, int t){ //Function for displaying hit information

if(hit==true){ // Target hit!

else if(hit==false){ // Target missed...

//Display if too close or too far

if(d<t-10){ // Shot too close

else if(d>t+10){ // Shot too far

int play(int &tnkPos, int diff, int &score, int ammo, int &att, char &b){//Projectile outputs

input (ang, power); //Prompt user for firing instructions

dist = ballDst(ang, power); //calculate projectile distance

hit = hitMiss(dist, tnkPos, diff);// your projectile hit

hitDisp(hit, dist, tnkPos); // //Display if hit or miss

if(hit==true){ //if hit, move tank, increment score

if(att==1){ // Hole in one bonus! Extra shot rewarded

if(ammo>1){ //reset number of attempts for next tank to 0

void highscores(int score, int diff){ // High score

string name; //Keep track of scores

fstream easyFile, hardFile;//Copy to a File

highscore scores[11];//11 high sores

if(diff==70){ // //High scores for easy

easyFile.open("easyscores.txt",ios::in); // //read in high scores from file

for(int i=0;i<10;i++){ //bring in values from file

easyFile>>scores[i].name;

easyFile>>scores[i].score; }

easyFile.close();//Close file

//test users new score

if(score>scores[9].score){ // New High Score

//set new score to 11th spot in array

scores[10].name = name;

scores[10].score = score;

for(int i = 10; i>0;i--){ //sort scores

if (scores[i].score > scores[i-1].score){ //set temp value

scores[11].score = scores[i].score;

scores[i].score = scores[i-1].score;

scores[i-1].score = scores[11].score;

//sort names

scores[11].name = scores[i].name;

scores[i].name = scores[i-1].name;

scores[i-1].name = scores[11].name;

easyFile.close();//Close file

easyFile.open("easyscores.txt", ios::out); //Open file for writing

easyFile.clear();//Clear file for writing

for(int i = 0; i<10;i++){ //write scores to file

easyFile.close();//close that file

for(int i=0; i<10;i++)//output high scores

else if(diff==30){ //high scores for hard

hardFile.open("hardscores.txt",ios::in); //read in high scores from file

for(int i=0;i<10;i++){//bring in values from file

hardFile>>scores[i].name;

hardFile>>scores[i].score;

if(score>scores[9].score){ //test users new score

//set new score to 11th spot in array

scores[10].name = name;

scores[10].score = score;

for(int i = 10; i>0;i--){//sort scores

if (scores[i].score > scores[i-1].score){//set temp value

scores[11].score = scores[i].score;

scores[i].score = scores[i-1].score;

scores[i-1].score = scores[11].score;

//sort names

scores[11].name = scores[i].name;

scores[i].name = scores[i-1].name;

scores[i-1].name = scores[11].name;

hardFile.close();//Close file

hardFile.open("hardscores.txt", ios::out); //Open file for writing

hardFile.clear();//Clear file for writing

for(int i = 0; i<10;i++){ //write scores to file

hardFile.close(); //close that file

for(int i=0; i<10;i++){//output high scores

***GAME CODE***

//Libraries

#include <iostream>

#include <ctime>

#include <iomanip>

#include <cstdlib>

#include <cmath>

#include <fstream>

#include <string>

using namespace std;

//Global constants

const float PI = 3.14;

const float G = 9.8;

//Function prototypes

int play(int&, int, int&, int, int&, char&);

int getDiff();

int plcTnk();

void highscores(int, int);

bool hitMiss(float, int, char);

void input(int &, int &);

int ballDst(int, int);

void hitDisp(bool, int, int);

//initialize structure for high scores

struct highscore{

string name;

int score;

};

//Execution begins here

int main(){

//Declare variables

int attempt, tnkPos, score, diff, ammo, att=0;

float dist;

char choice, b;

char chc;

int x;

float d;

//Display title and story

cout<<" Fatal Trajectory "<<endl;

cout<<"You're under attack!!"<<endl;

cout<<"Outside your base, several tanks appear"<<endl;

cout<<"The tanks will come within 1000 meters."<<endl;

cout<<"Input the angle to aim your cannon and the speed to fire the projectile."<<endl;

cout<<"You have 15 shots to destroy as many as you can."<<endl;

cout<<endl;

//Get the system time

unsigned seed = time(0);

//seed random number generator

srand(seed);

do{

tnkPos = plcTnk(); //place tank

//Prompt for difficulty

diff = getDiff();

cout<<endl<<endl;

//Reset scores and ammo

score = 0;

ammo = 15;

//Loop for tank destruction

for (int attmpt = 0; attmpt<15; attmpt++){

//reset bonus to 0

b=0;

//call main game play function

play (tnkPos, diff, score, ammo, att, b);

ammo--;

//Bonus shot rewarding system

//increment shots remaining message

//decrement for loop

if (b == 1){

ammo++;

attempt--;

}

}

cout<<"score = "<<score<<endl;

cout<<"The number of tank you hit = "<<score/50<<endl;

//call function to show high scores

//Ask user for replay

cout<<"You want to Play again?"<<endl;

cout<<"Type Y to play again"<<endl;

cin>>choice;

}while(choice=='y'||choice=='Y');

return 0;

}

//function for selecting difficulty

int getDiff(){

char chc;

int diff;

//Prompt for difficulty

cout<<"How Difficult you want the game to be?"<<endl;

cout<<"Type 'e' for easy, type 'h' for hard"<<endl;

cin>>chc;

//Input

while(chc!='e'&&chc!='E' && chc!='h'&&chc!='H'){

cout<<"Invalid choice, type E or H"<<endl;

cin>>chc;

cout<<endl;

}

//Initialize difficulty by setting explosive blast radius

if(chc == 'e'||chc == 'E'){

cout<<"Easy mode activated!"<<endl;

cout<<"Cluster bombs enabled"<<endl;

cout<<"Blast radius: 70m"<<endl;

diff = 70;

}

else if(chc == 'h'||chc == 'H'){

cout<<"Hard mode!"<<endl;

cout<<"projectile blast radius: 30m"<<endl;

diff = 30;

}

return diff;

}

//Function to place tank

int plcTnk(){

int x;

x = 1 + rand() % 999;

return x;

}

//Function for user firing input

void input(int &a,int &p){

//Prompt the user for inputs

do{

cout<<"Input firing angle (0-90)"<<endl;

cin>>a;

}while(a>90||a<0);

do{

cout<<"Input projectile power in meters per second (1-100)"<<endl;

cin>>p;

}while(p>100||p<0);

}

//Function calculating where projectile hits

int ballDst(int a, int p){

float d;

d = ((p\*p)\*sin(2\*(a\*(PI/180))))/G;

return d;

}

//Function testing if projectile hit

bool hitMiss(float dist, int t, char diff){

//test if projectiles blast radius hit tank

if (dist > t-diff && dist < t+diff){

return true;

}

else

return false;

}

//Function for displaying hit information

void hitDisp(bool hit, int d, int t){

if(hit==true){

cout<<"Target hit!"<<endl;

cout<<"Tank was at "<<t<<"m"<<endl;

}

else if(hit==false){

cout<<"Target missed..."<<endl;

//Display if too close or too far

if(d<t-10){

cout<<"Shot too close"<<endl;

}

else if(d>t+10){

cout<<"Shot too far"<<endl;

}

}

cout<<endl<<endl;

}

int play(int &tnkPos, int diff, int &score, int ammo, int &att, char &b){

int ang, power;

float dist;

bool hit;

//show how many shots left

cout<<"Tank is approx. "<<tnkPos<<" meters away."<<endl;

cout<<"You have "<<ammo<<" shots left."<<endl;

//Prompt user for firing instructions

input (ang, power);

//calculate projectile distance

dist = ballDst(ang, power);

cout<<endl<<endl;

//test if hit or miss

hit = hitMiss(dist, tnkPos, diff);

cout<<"your projectile hit "<<dist<<" meters"<<endl<<endl;

//Display if hit or miss

hitDisp(hit, dist, tnkPos);

//increment bonus tester

att++;

//if hit, move tank, increment score

if(hit==true){

tnkPos = plcTnk();

score+=50;

//if tank was destroyed on first try, reward with extra shot

if(att==1){

cout<<"Hole in one bonus!"<<endl;

cout<<"Extra shot rewarded"<<endl<<endl;

b = 1;

}

//reset number of attempts for next tank to 0

if(ammo>1){

cout<<"Another tank spotted!"<<endl<<endl;

}

att=0;

}

return score;

}

void highscores(int score, int diff){

string name;

fstream easyFile, hardFile;

highscore scores[11];

//High scores for easy

if(diff==70){

//read in high scores from file

easyFile.open("easyscores.txt",ios::in);

//bring in values from file

for(int i=0;i<10;i++){

easyFile>>scores[i].name;

easyFile>>scores[i].score;

}

//Close file

easyFile.close();

//test users new score

if(score>scores[9].score){

cout<<"New High Score!"<<endl;

cout<<"Input name:"<<endl;

cin>>name;

//set new score to 11th spot in array

scores[10].name = name;

scores[10].score = score;

//sort scores

for(int i = 10; i>0;i--){

if (scores[i].score > scores[i-1].score){

//set scores value

scores[11].score = scores[i].score;

scores[i].score = scores[i-1].score;

scores[i-1].score = scores[11].score;

//sort names

scores[11].name = scores[i].name;

scores[i].name = scores[i-1].name;

scores[i-1].name = scores[11].name;

}

}

}

//Close file

easyFile.close();

//Open file for writing

easyFile.open("easyscores.txt", ios::out);

//Clear file for writing

easyFile.clear();

//write scores to file

for(int i = 0; i<10;i++){

easyFile<<scores[i].name<<setw(15)<<scores[i].score<<endl;

}

//close that file

easyFile.close();

//output high scores

cout<<"Easy High Scores!\n"<<endl;

for(int i=0; i<10;i++){

cout<<scores[i].name<<setw(10)<<scores[i].score<<endl;

}

}

//high scores for hard

else if(diff==30){

//read in high scores from file

hardFile.open("hardscores.txt",ios::in);

//bring in values from file

for(int i=0;i<10;i++){

hardFile>>scores[i].name;

hardFile>>scores[i].score;

}

//test users new score

if(score>scores[9].score){

cout<<"New High Score!"<<endl;

cout<<"Input name:"<<endl;

cin>>name;

//set new score to 11th spot in array

scores[10].name = name;

scores[10].score = score;

//sort scores

for(int i = 10; i>0;i--){

if (scores[i].score > scores[i-1].score){

//set temp value

scores[11].score = scores[i].score;

scores[i].score = scores[i-1].score;

scores[i-1].score = scores[11].score;

//sort names

scores[11].name = scores[i].name;

scores[i].name = scores[i-1].name;

scores[i-1].name = scores[11].name;

}

}

}

//Close file

hardFile.close();

//Open file for writing

hardFile.open("hardscores.txt", ios::out);

//Clear file for writing

hardFile.clear();

//write scores to file

for(int i = 0; i<10;i++){

hardFile<<scores[i].name<<setw(15)<<scores[i].score<<endl;

}

//close that file

hardFile.close();

//output high scores

cout<<"Hard High Scores!"<<endl;

cout<<endl;

for(int i=0; i<10;i++){

cout<<scores[i].name<<setw(10)<<scores[i].score<<endl;

}

}

}