**Space shooter Game**



Session: 2022 – 2026

**Submitted by:**

Syed Muhammad Faisal Ilyas 2022-CS-63

**Supervised by:**

Maida Shahid

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

**Table of Contents**

Table of Contents

[Short Description and Story Writing of Game 3](#_Toc129995144)

[Game Characters Description 3](#_Toc129995145)

[Game Objects Description 4](#_Toc129995146)

[Rules & Interactions 5](#_Toc129995147)

[Goal of the Game 5](#_Toc129995148)

Wireframes of game ……………………………………………………………………………..5

Data Structures (2D Arrays)……………………………………………………………………..7

Function Prototypes……………………………………………………………………………….8

Complete Code…………………………………………………………………………………..10

# Short Description and Story Writing of Game

Space Shooter is an action-packed game set in a futuristic space environment. In this game, players control a spaceship and must navigate through a dangerous space filled with enemy ships. The goal is to survive as long as possible while shooting down enemy ships and collecting power-ups.The player will also fire bullets onto the enemies and if the enemies will get the bullets they will be destroyed.The enemies will leave behind powers and fires on getting bullets from the player . If the shooter collects a power, their power will increase and they will become stronger. However, if the shooter collects a fire, their power will decrease until they die. Additionally, if an enemy ship from above collides with the shooter, the game will be over. Players must navigate the dangerous space with skill and strategy to survive and emerge victorious.

# Game Characters Description

**Player**

There is one human player in the Game.

**Rockey:**

In Space Shooter, Rockey is the main character that controls a spaceship and must navigate through a dangerous space filled with enemy ships. The player's goal is to survive as long as possible while shooting down enemy ships and collecting power-ups. The player's spaceship is equipped with weapons and other abilities that can be used to defeat the enemy ships and stay alive.

**Enemies**

There are 5 enemies in the game.

**The Ravager:**

It is a big and powerful, heavily armored enemy with multiple weapons systems that attacks relentlessly.

**The Swarm:**

It is a small, nimble enemy that flies and overwhelms the player with their numbers.

**The Hunter:**

A stealthy enemy that waits to ambush the player with its powerful plasma weapons.

**The Colossus:**

A massive enemy with a massive amount of armor and firepower, making it a formidable opponent.

**The Tempest:**

A fast and agile enemy that use its speed and mobility to evade the player's attacks and launch devastating counterattacks

# Game Objects Description

Following are the Objects in the Game

**Power-ups:**

A Power-up, also known as an Energizer, is an object used in the space shooter game. When the player will destroy an enemy it will release power-ups sometimes and if the player will get this power it’s power will be increased by one.

**Fires:**

The fires are the weapons released by the enemies when they will be destroyed.If the player will get this fire its power will be decreased by one.

**Bullets:**

The bullets are the main weapon of the player.It will hit the enemies and destroy them.

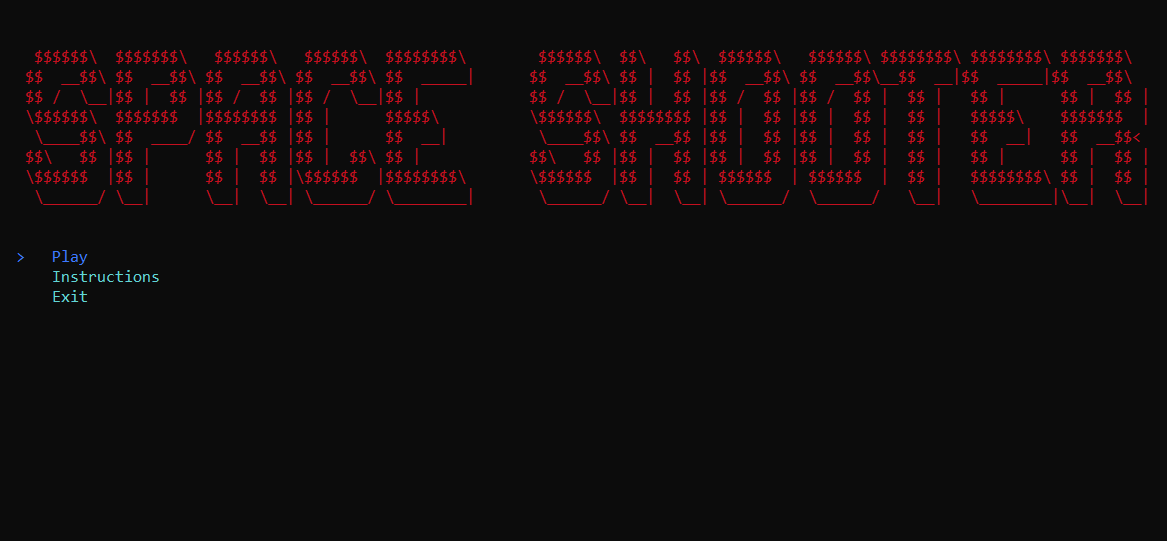
# Rules & Interactions

The player Rocky will release bullets constantly and can move to protect himself from the enemies.If Rockey’s bullets will touch any enemy,that enemy will be destroyed and may leave a power up or a fire.If the rocky will get power-up, his power will be increased by one and if he will get a fire , his power will be decreased by one.The enemies will move towards the player.If the player will get hit by the enemy,the player will die and the game will be over and in the other case his power will become zero he will die and the game will be over.

# Goal of the Game

The goal of the game is to get more and more power-ups and to survive and protect yourself from the enemies.

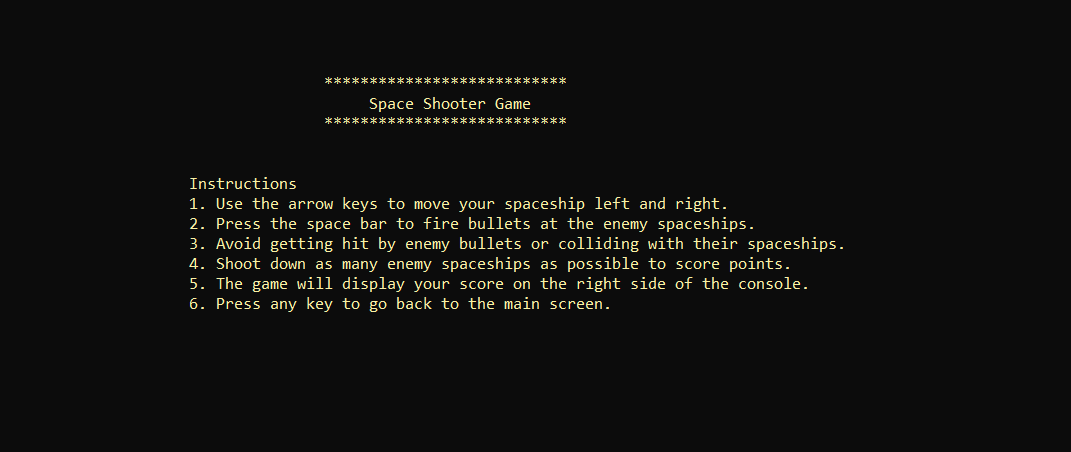
**Wireframes of the Game**

****

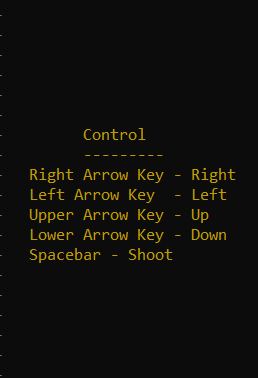
**Figure 1. Main Menu**

****

**Figure 2. Game Story**

****

**Figure 3. Instructions**



**Figure 4. Controls**

**Data Structures (2D Arrays)**

char player\_space\_ship[7][13] = {{' ', ' ', ' ', ' ', ' ', ' ', '\_', ' ', ' ', ' ', ' ', ' ', ' '},

{' ', ' ', ' ', ' ', ' ', '/', ' ', '\\', ' ', ' ', ' ', ' ', ' '},

{' ', ' ', ' ', ' ', '/', ' ', '\_', ' ', '\\', ' ', ' ', ' ', ' '},

{' ', ' ', ' ', '/', ' ', ' ', '\_', ' ', ' ', '\\', ' ', ' ', ' '},

{' ', '\_', '/', ' ', ' ', ' ', '\_', ' ', ' ', ' ', '\\', '\_', ' '},

{'|', '\_', '\_', '\_', '\_', '\_', ' ', '\_', '\_', '\_', '\_', '\_', '|'},

{' ', ' ', ' ', ' ', ' ', '\\', '\_', '/', ' ', ' ', ' ', ' '}};

char enemy1\_space\_ship[5][7] = {{' ', ' ', '\*', '\*', '\*', ' ', ' '},

{'\*', '\*', '\*', '\*', '\*', '\*', '\*'},

{'\*', '\*', '\*', '\*', '\*', '\*', '\*'},

{' ', '\*', '\*', '\*', '\*', '\*', ' '},

{' ', ' ', '\*', '\*', '\*', ' ', ' '}};

char enemy2\_space\_ship[5][5] = {{' ', ' ', '\_', ' ', ' '},

{' ', '|', 'o', '|', ' '},

{' ', '(', '|', ')', ' '},

{' ', '/', '|', '\\', ' '},

{'/', ' ', '|', ' ', '\\'}};

char enemy3\_space\_ship[4][4] = {{'\_', '^', '^', '\_'},

{'\*', '\*', '\*', '\*'},

{'\*', '\*', '\*', '\*'},

{'.', '\*', '\*', '.'}};

char enemy4\_space\_ship[6][7] = {{' ', ' ', '\_', '\_', ' ', ' ', ' '},

{' ', '/', ' ', ' ', '\\', ' ', ' '},

{'-', '|', ' ', ' ', '|', '-', ' '},

{'/', ' ', '%', ' ', '%', ' ', '\\'},

{' ', ':', ' ', ' ', ' ', ':', ' '},

{' ', ' ', '^', '-', '^', ' ', ' '}};

char enemy5\_space\_ship[4][4] = {{character, character, character, character},

{character, character, character, character},

{character, character, character, character},

{' ', '\\', '/', ' '}};

**Function Prototypes**

* void Play();
* void Instructions();
* void loading();
* void history();
* void drawborder();
* void printmenu();
* void logo();
* void header();
* void gotoxy(int x, int y);
* char getcharAtxy(short int x, short int y);
* void hideCursor();
* void stopBlinking();
* void printspaceship();
* void erasespaceship();
* void moveplayerleft();
* void moveplayerright();
* void moveplayerup();
* void moveplayerdown();
* void printenemy1();
* void eraseenemy1();
* void printenemy2();
* void eraseenemy2();
* void printenemy3();
* void eraseenemy3();
* void printenemy4();
* void eraseenemy4();
* void printenemy5();
* void eraseenemy5();
* void moveenemy1();
* void moveenemy2();
* void moveenemy3();
* void moveenemy4();
* void moveenemy5();
* void generatebullet();
* void movebullet();
* void printbullet(int x, int y);
* void erasebullet(int x, int y);
* void makebulletInactive(int index);
* void deletebullet(int index);
* void bulletcollisionwidenemy();
* void gameover();
* void controlmenu();
* void storescore(int final\_score);
* void readscore();
* void addscore();
* void updatescore();
* bool checkBestScore(int score);
* void loselife();
* void updatelives();
* void setColor(int colorCode);

**Complete Code**

#include <iostream>

#include <windows.h>

#include <conio.h>

#include <fstream>

#include <stdlib.h>

#include <thread>

#include <chrono>

#define SCREEN\_HEIGHT 40

#define SCREEN\_WIDTH 180

#define WIN\_WIDTH 150

using namespace std;

//---- Prototypes---

void playBeep();

// void loadpic();

void Play();

void Instructions();

void loading();

void history();

void drawborder();

void printmenu();

void logo();

void header();

void controlmenu();

// ----BuiltIn Funtions---

void gotoxy(int x, int y);

char getcharAtxy(short int x, short int y);

void hideCursor();

void stopBlinking();

// ---Playerspaceship---

void printspaceship();

void erasespaceship();

// ----Moving Players----

void moveplayerleft();

void moveplayerright();

void moveplayerup();

void moveplayerdown();

// ----Enemies Functions---

void printenemy1();

void eraseenemy1();

void printenemy2();

void eraseenemy2();

void printenemy3();

void eraseenemy3();

void printenemy4();

void eraseenemy4();

void printenemy5();

void eraseenemy5();

// -----Moving ENEMIES------

void moveenemy1();

void moveenemy2();

void moveenemy3();

void moveenemy4();

void moveenemy5();

// ----bullets----

void generatebullet();

void movebullet();

void printbullet(int x, int y);

void erasebullet(int x, int y);

void makebulletInactive(int index);

void deletebullet(int index);

// ---collision---

void bulletcollisionwidenemy();

// ----gameover---

void gameover();

// ----score-----

void addscore();

void updatescore();

bool checkBestScore(int score);

void loselife();

void updatelives();

// ---File Handling---

void readBorderFromFile();

void writeBorderToFile();

void storescore(int final\_score);

void readscore();

// ---color---

void setColor(int colorCode);

// ----Counters nd coordinates---

char character = 219;

char bulletchar = 30;

int PX = 5;

int PY = 30;

int E1X = 20;

int E1Y = 26;

int E5X = 50;

int E5Y = 10;

int E2X = 108;

int E2Y = 2;

int E3X = 140;

int E3Y = 20;

int E4X = 76;

int E4Y = 16;

int score = 0;

int score\_count = 0;

int lives = 5;

int enemy1counter = 0;

int bulletcount = 0;

bool gamerunning = true;

string enemydirection = "down";

// ---Arrays---

int bulletx[100];

int bullety[100];

bool isBulletActice[100];

int Score[100];

// ----Players and Enemies Characters

char player\_space\_ship[7][13] = {{' ', ' ', ' ', ' ', ' ', ' ', '\_', ' ', ' ', ' ', ' ', ' ', ' '},

{' ', ' ', ' ', ' ', ' ', '/', ' ', '\\', ' ', ' ', ' ', ' ', ' '},

{' ', ' ', ' ', ' ', '/', ' ', '\_', ' ', '\\', ' ', ' ', ' ', ' '},

{' ', ' ', ' ', '/', ' ', ' ', '\_', ' ', ' ', '\\', ' ', ' ', ' '},

{' ', '\_', '/', ' ', ' ', ' ', '\_', ' ', ' ', ' ', '\\', '\_', ' '},

{'|', '\_', '\_', '\_', '\_', '\_', ' ', '\_', '\_', '\_', '\_', '\_', '|'},

{' ', ' ', ' ', ' ', ' ', '\\', '\_', '/', ' ', ' ', ' ', ' '}};

char enemy1\_space\_ship[5][7] = {{' ', ' ', '\*', '\*', '\*', ' ', ' '},

{'\*', '\*', '\*', '\*', '\*', '\*', '\*'},

{'\*', '\*', '\*', '\*', '\*', '\*', '\*'},

{' ', '\*', '\*', '\*', '\*', '\*', ' '},

{' ', ' ', '\*', '\*', '\*', ' ', ' '}};

char enemy2\_space\_ship[5][5] = {{' ', ' ', '\_', ' ', ' '},

{' ', '|', 'o', '|', ' '},

{' ', '(', '|', ')', ' '},

{' ', '/', '|', '\\', ' '},

{'/', ' ', '|', ' ', '\\'}};

char enemy3\_space\_ship[4][4] = {{'\_', '^', '^', '\_'},

{'\*', '\*', '\*', '\*'},

{'\*', '\*', '\*', '\*'},

{'.', '\*', '\*', '.'}};

char enemy4\_space\_ship[6][7] = {{' ', ' ', '\_', '\_', ' ', ' ', ' '},

{' ', '/', ' ', ' ', '\\', ' ', ' '},

{'-', '|', ' ', ' ', '|', '-', ' '},

{'/', ' ', '%', ' ', '%', ' ', '\\'},

{' ', ':', ' ', ' ', ' ', ':', ' '},

{' ', ' ', '^', '-', '^', ' ', ' '}};

char enemy5\_space\_ship[4][4] = {{character, character, character, character},

{character, character, character, character},

{character, character, character, character},

{' ', '\\', '/', ' '}};

main()

{

// loadpic();

hideCursor();

system("cls");

header();

// Sleep(700);

getch();

int currentSelection = 1;

int key;

setColor(15);

while (gamerunning)

{

system("cls");

logo();

if (currentSelection == 1)

{

setColor(9); // Blue color

cout << " >";

}

else

{

setColor(11);

cout << " ";

}

cout << " Play" << endl;

if (currentSelection == 2)

{

setColor(9); // Blue color

cout << " >";

}

else

{

setColor(11);

cout << " ";

}

cout << " Instructions" << endl;

if (currentSelection == 3)

{

setColor(9); // Blue color

cout << " >";

}

else

{

setColor(11);

cout << " ";

}

cout << " Exit" << endl;

// handle user input

key = getch();

if (key == 72)

{ // up arrow key

if (currentSelection > 1)

{

currentSelection--;

}

else if (currentSelection == 1)

{

currentSelection = 3;

}

}

else if (key == 80)

{ // down arrow key

if (currentSelection < 3)

{

currentSelection++;

}

else if (currentSelection == 3)

{

currentSelection = 1;

}

}

else if (key == 13)

{ // enter key

if (currentSelection == 1)

{

Play();

}

else if (currentSelection == 2)

{

system("cls");

Instructions();

getch(); // wait for user input before going back to main screen

}

else if (currentSelection == 3)

{

exit(0);

}

}

}

}

// ----Play---

void Play()

{

readscore();

system("cls");

history();

system("Color 0D");

system("cls");

// printmaze();

drawborder();

controlmenu();

updatescore();

updatelives();

printspaceship();

printenemy1();

printenemy2();

printenemy3();

printenemy4();

printenemy5();

while (gamerunning)

{

enemy1counter++;

moveenemy1();

moveenemy2();

moveenemy3();

moveenemy4();

moveenemy5();

if (GetAsyncKeyState(VK\_LEFT))

{

moveplayerleft();

}

if (GetAsyncKeyState(VK\_RIGHT))

{

moveplayerright();

}

if (GetAsyncKeyState(VK\_UP))

{

moveplayerup();

}

if (GetAsyncKeyState(VK\_DOWN))

{

moveplayerdown();

}

if (GetAsyncKeyState(VK\_SPACE))

{

generatebullet();

// PlaySound(TEXT("C:/PF/Game Development/shipLaser.wav"),NULL,SND\_FILENAME);

Beep(1000, 50);

// Beep(2000,50);

// Beep(3000,50);

}

if (GetAsyncKeyState(VK\_ESCAPE))

{

gamerunning = false;

}

if (gamerunning == false)

{

system("cls");

gameover();

cin.get();

Score[score\_count] = score;

storescore(Score[score\_count]);

score\_count++;

}

movebullet(); // to move bullets

bulletcollisionwidenemy();

Sleep(2); // to slow the game speed if you want to move the enemies fast the pass the small value to small function like 1 , 2 ,3 and same for slow pass the larger value like 50 or 70

}

}

//----- Moving Players Functions----

void moveplayerleft()

{

enemy1counter++;

if (enemy1counter % 3 == 0)

{

moveenemy1();

}

if (enemy1counter % 3 == 0)

{

moveenemy2();

}

if (enemy1counter % 3 == 0)

{

moveenemy3();

}

if (enemy1counter % 3 == 0)

{

moveenemy4();

}

if (enemy1counter % 3 == 0)

{

moveenemy5();

}

char next = getcharAtxy(PX - 1, PY);

if (next == ' ')

{

erasespaceship();

PX--;

printspaceship();

}

}

void moveplayerright()

{

enemy1counter++;

if (enemy1counter % 3 == 0)

{

moveenemy1();

}

if (enemy1counter % 3 == 0)

{

moveenemy2();

}

if (enemy1counter % 3 == 0)

{

moveenemy3();

}

if (enemy1counter % 3 == 0)

{

moveenemy4();

}

if (enemy1counter % 3 == 0)

{

moveenemy5();

}

char next = getcharAtxy(PX + 13, PY);

if (next == ' ')

{

erasespaceship();

PX++;

printspaceship();

}

}

void moveplayerup()

{

enemy1counter++;

if (enemy1counter % 3 == 0)

{

moveenemy1();

}

if (enemy1counter % 3 == 0)

{

moveenemy2();

}

if (enemy1counter % 3 == 0)

{

moveenemy3();

}

if (enemy1counter % 3 == 0)

{

moveenemy4();

}

if (enemy1counter % 3 == 0)

{

moveenemy5();

}

char next = getcharAtxy(PX, PY - 1);

if (next == ' ')

{

erasespaceship();

PY--;

printspaceship();

}

}

void moveplayerdown()

{

enemy1counter++;

if (enemy1counter % 3 == 0)

{

moveenemy1();

}

if (enemy1counter % 3 == 0)

{

moveenemy2();

}

if (enemy1counter % 3 == 0)

{

moveenemy3();

}

if (enemy1counter % 3 == 0)

{

moveenemy4();

}

if (enemy1counter % 3 == 0)

{

moveenemy5();

}

char next = getcharAtxy(PX, PY + 7);

if (next == ' ')

{

erasespaceship();

PY++;

printspaceship();

}

}

// -----Bullets----

void generatebullet()

{

bulletx[bulletcount] = PX + 6;

bullety[bulletcount] = PY - 1;

isBulletActice[bulletcount] = true;

gotoxy(bulletx[bulletcount], bullety[bulletcount]);

cout << bulletchar;

bulletcount++;

}

void movebullet()

{

for (int x = 0; x < bulletcount; x++)

{

if (isBulletActice[x] = true)

{

char next = getcharAtxy(bulletx[x], bullety[x] - 1); // get the next possition of bullet if valid the move the bullets else remove it

if (next == ' ') // to check the conditions if you want to add the collision of bullet with enemy then simply add the enemy char with or (||) operater in if statement of enemy remove function in if body

{

erasebullet(bulletx[x], bullety[x]);

bullety[x] = bullety[x] - 1;

printbullet(bulletx[x], bullety[x]);

}

else // remove the bullets

{

erasebullet(bulletx[x], bullety[x]);

makebulletInactive(x);

deletebullet(x);

}

}

}

}

void printbullet(int x, int y)

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 4);

gotoxy(x, y);

cout << bulletchar;

}

void erasebullet(int x, int y)

{

gotoxy(x, y);

cout << ' ';

}

void makebulletInactive(int index)

{

isBulletActice[index] = false;

}

void deletebullet(int index)

{

for (int i = index; i < bulletcount; i++)

{

bulletx[i] = bulletx[i + 1];

bullety[i] = bullety[i + 1];

}

bulletcount--;

}

// -----Moving Enemies-----

void moveenemy1()

{

if (enemydirection == "down")

{

char next = getcharAtxy(E1X, E1Y + 5);

if (next == ' ')

{

eraseenemy1();

E1Y++;

printenemy1();

}

if (next == '+')

{

eraseenemy1();

E1Y = 1;

printenemy1();

}

for (int i = 0; i < 6; i++)

{

next = getcharAtxy(E1X + i, E1Y + 5);

if (next == '\_' || next == '\\' || next == '/' || next == '|')

{

// gotoxy(E1X + i, E1Y + 5);

// erasespaceship();

loselife();

updatelives();

}

}

}

}

void moveenemy2()

{

if (enemydirection == "down")

{

char next = getcharAtxy(E2X, E2Y + 5);

if (next == ' ')

{

eraseenemy2();

E2Y++;

printenemy2();

}

if (next == '+')

{

eraseenemy2();

E2Y = 4;

printenemy2();

}

for (int i = 0; i < 6; i++)

{

next = getcharAtxy(E2X + i, E2Y + 5);

if (next == '\_' || next == '\\' || next == '/' || next == '|')

{

loselife();

updatelives();

}

}

}

}

void moveenemy3()

{

if (enemydirection == "down")

{

char next = getcharAtxy(E3X, E3Y + 4);

if (next == ' ')

{

eraseenemy3();

E3Y++;

printenemy3();

}

if (next == '+')

{

eraseenemy3();

E3Y = 1;

printenemy3();

}

for (int i = 0; i < 6; i++)

{

next = getcharAtxy(E3X + i, E3Y + 5);

if (next == '\_' || next == '\\' || next == '/' || next == '|')

{

loselife();

updatelives();

}

}

}

}

void moveenemy4()

{

if (enemydirection == "down")

{

char next = getcharAtxy(E4X, E4Y + 6);

if (next == ' ')

{

eraseenemy4();

E4Y++;

printenemy4();

}

if (next == '+')

{

eraseenemy4();

E4Y = 1;

printenemy4();

}

for (int i = 0; i < 6; i++)

{

next = getcharAtxy(E4X + i, E4Y + 5);

if (next == '\_' || next == '\\' || next == '/' || next == '|')

{

loselife();

updatelives();

}

}

}

}

void moveenemy5()

{

if (enemydirection == "down")

{

char next = getcharAtxy(E5X, E5Y + 4);

if (next == ' ')

{

eraseenemy5();

E5Y++;

printenemy5();

}

if (next == '+')

{

eraseenemy5();

E5Y = 1;

printenemy5();

}

for (int i = 0; i < 6; i++)

{

next = getcharAtxy(E5X + i, E5Y + 5);

if (next == '\_' || next == '\\' || next == '/' || next == '|')

{

loselife();

updatelives();

}

}

}

}

// ----Printing and Erasing function---

void printspaceship()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 1);

for (int rows = 0; rows < 7; rows++)

{

gotoxy(PX, PY + rows);

for (int col = 0; col < 13; col++)

{

cout << player\_space\_ship[rows][col];

}

cout << endl;

}

}

void erasespaceship()

{

for (int rows = 0; rows < 7; rows++)

{

gotoxy(PX, PY + rows);

for (int col = 0; col < 13; col++)

{

cout << " ";

}

cout << endl;

}

}

void printenemy1()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 13);

for (int rows = 0; rows < 5; rows++)

{

gotoxy(E1X, E1Y + rows);

for (int col = 0; col < 7; col++)

{

cout << enemy1\_space\_ship[rows][col];

}

cout << endl;

}

}

void eraseenemy1()

{

for (int rows = 0; rows < 5; rows++)

{

gotoxy(E1X, E1Y + rows);

for (int col = 0; col < 7; col++)

{

cout << " ";

}

cout << endl;

}

}

void printenemy2()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 14);

for (int rows = 0; rows < 5; rows++)

{

gotoxy(E2X, E2Y + rows);

for (int col = 0; col < 5; col++)

{

cout << enemy2\_space\_ship[rows][col];

}

cout << endl;

}

}

void eraseenemy2()

{

for (int rows = 0; rows < 5; rows++)

{

gotoxy(E2X, E2Y + rows);

for (int col = 0; col < 5; col++)

{

cout << " ";

}

cout << endl;

}

}

void printenemy3()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 12);

for (int rows = 0; rows < 4; rows++)

{

gotoxy(E3X, E3Y + rows);

for (int col = 0; col < 4; col++)

{

cout << enemy3\_space\_ship[rows][col];

}

cout << endl;

}

}

void eraseenemy3()

{

for (int rows = 0; rows < 4; rows++)

{

gotoxy(E3X, E3Y + rows);

for (int col = 0; col < 4; col++)

{

cout << " ";

}

cout << endl;

}

}

void printenemy4()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 3);

for (int rows = 0; rows < 6; rows++)

{

gotoxy(E4X, E4Y + rows);

for (int col = 0; col < 7; col++)

{

cout << enemy4\_space\_ship[rows][col];

}

cout << endl;

}

}

void eraseenemy4()

{

for (int rows = 0; rows < 6; rows++)

{

gotoxy(E4X, E4Y + rows);

for (int col = 0; col < 7; col++)

{

cout << " ";

}

cout << endl;

}

}

void printenemy5()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 2);

for (int rows = 0; rows < 4; rows++)

{

gotoxy(E5X, E5Y + rows);

for (int col = 0; col < 4; col++)

{

cout << enemy5\_space\_ship[rows][col];

}

cout << endl;

}

}

void eraseenemy5()

{

for (int rows = 0; rows < 4; rows++)

{

gotoxy(E5X, E5Y + rows);

for (int col = 0; col < 4; col++)

{

cout << " ";

}

cout << endl;

}

}

// ----Built-In Functions----

void gotoxy(int x, int y)

{

COORD coord;

coord.X = x;

coord.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

char getcharAtxy(short int x, short int y)

{

CHAR\_INFO ci;

COORD xy = {0, 0};

SMALL\_RECT rect = {x, y, x, y};

COORD coordBufSize;

coordBufSize.X = 1;

coordBufSize.Y = 1;

return ReadConsoleOutput(GetStdHandle(STD\_OUTPUT\_HANDLE), &ci, coordBufSize, xy, &rect) ? ci.Char.AsciiChar : ' ';

}

void hideCursor()

{

HANDLE console;

console = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_CURSOR\_INFO cursor;

cursor.dwSize = 1;

cursor.bVisible = false;

SetConsoleCursorInfo(console, &cursor);

}

void stopBlinking()

{

HANDLE consoleHandle = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_CURSOR\_INFO cursorInfo;

GetConsoleCursorInfo(consoleHandle, &cursorInfo);

cursorInfo.bVisible = false; // set the cursor visibility to false

SetConsoleCursorInfo(consoleHandle, &cursorInfo);

}

// ----Drawing border-----

void drawborder()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 8);

for (int i = 0; i < SCREEN\_WIDTH; i++)

{

cout << "+";

}

for (int i = 0; i < SCREEN\_WIDTH; i++)

{

gotoxy(i, SCREEN\_HEIGHT);

cout << "+";

}

for (int i = 0; i < SCREEN\_HEIGHT; i++)

{

gotoxy(0, i);

cout << "+";

gotoxy(SCREEN\_WIDTH, i);

cout << "+";

}

for (int i = 0; i < SCREEN\_HEIGHT; i++)

{

gotoxy(WIN\_WIDTH, i);

cout << "+";

}

}

// ----Printing Functions---

void printmenu()

{

system("Color 04");

cout << "1. Play Game" << endl;

cout << "2. Instructions" << endl;

cout << "3. Exit" << endl;

cout << "Enter option:";

}

void header()

{

system("color 17");

cout << " " << endl;

cout << " " << endl;

cout << " " << endl;

cout << " " << endl;

cout << " " << endl;

cout << " " << endl;

cout << " " << endl;

cout << " SSSSSSSSSSSSSSS " << endl;

cout << " SS:::::::::::::::S " << endl;

cout << " S:::::SSSSSS::::::S " << endl;

cout << " S:::::S SSSSSSS " << endl;

cout << " S:::::S ppppp ppppppppp aaaaaaaaaaaaa cccccccccccccccc eeeeeeeeeeee " << endl;

cout << " S:::::S p::::ppp:::::::::p a::::::::::::a cc:::::::::::::::c ee::::::::::::ee " << endl;

cout << " S::::SSSS p:::::::::::::::::p aaaaaaaaa:::::a c:::::::::::::::::c e::::::eeeee:::::ee " << endl;

cout << " SS::::::SSSSS pp::::::ppppp::::::p a::::ac:::::::cccccc:::::ce::::::e e:::::e " << endl;

cout << " SSS::::::::SS p:::::p p:::::p aaaaaaa:::::ac::::::c ccccccce:::::::eeeee::::::e " << endl;

cout << " SSSSSS::::S p:::::p p:::::p aa::::::::::::ac:::::c e:::::::::::::::::e " << endl;

cout << " S:::::Sp:::::p p:::::p a::::aaaa::::::ac:::::c e::::::eeeeeeeeeee " << endl;

cout << " S:::::Sp:::::p p::::::pa::::a a:::::ac::::::c ccccccce:::::::e " << endl;

cout << " SSSSSSS S:::::Sp:::::ppppp:::::::pa::::a a:::::ac:::::::cccccc:::::ce::::::::e " << endl;

cout << " S::::::SSSSSS:::::Sp::::::::::::::::p a:::::aaaa::::::a c:::::::::::::::::c e::::::::eeeeeeee " << endl;

cout << " S:::::::::::::::SS p::::::::::::::pp a::::::::::aa:::a cc:::::::::::::::c ee:::::::::::::e " << endl;

cout << " SSSSSSSSSSSSSSS p::::::pppppppp aaaaaaaaaa aaaa cccccccccccccccc eeeeeeeeeeeeee " << endl;

cout << " p:::::p " << endl;

cout << " p:::::p " << endl;

cout << " SSSSSSSSSSSSSSS hhhhhhh p:::::::p tttt " << endl;

cout << " SS:::::::::::::::Sh:::::h p:::::::p ttt:::t " << endl;

cout << " S:::::SSSSSS::::::Sh:::::h p:::::::p t:::::t " << endl;

cout << " S:::::S SSSSSSSh:::::h ppppppppp t:::::t " << endl;

cout << " S:::::S h::::h hhhhh ooooooooooo ooooooooooo ttttttt:::::ttttttt eeeeeeeeeeee rrrrr rrrrrrrrr " << endl;

cout << " S:::::S h::::hh:::::hhh oo:::::::::::oo oo:::::::::::oo t:::::::::::::::::t ee::::::::::::ee r::::rrr:::::::::r " << endl;

cout << " S::::SSSS h::::::::::::::hh o:::::::::::::::oo:::::::::::::::ot:::::::::::::::::t e::::::eeeee:::::eer:::::::::::::::::r " << endl;

cout << " SS::::::SSSSS h:::::::hhh::::::h o:::::ooooo:::::oo:::::ooooo:::::otttttt:::::::tttttt e::::::e e:::::err::::::rrrrr::::::r" << endl;

cout << " SSS::::::::SS h::::::h h::::::ho::::o o::::oo::::o o::::o t:::::t e:::::::eeeee::::::e r:::::r r:::::r" << endl;

cout << " SSSSSS::::S h:::::h h:::::ho::::o o::::oo::::o o::::o t:::::t e:::::::::::::::::e r:::::r rrrrrrr" << endl;

cout << " S:::::S h:::::h h:::::ho::::o o::::oo::::o o::::o t:::::t e::::::eeeeeeeeeee r:::::r " << endl;

cout << " S:::::S h:::::h h:::::ho::::o o::::oo::::o o::::o t:::::t tttttte:::::::e r:::::r " << endl;

cout << " SSSSSSS S:::::S h:::::h h:::::ho:::::ooooo:::::oo:::::ooooo:::::o t::::::tttt:::::te::::::::e r:::::r " << endl;

cout << " S::::::SSSSSS:::::S h:::::h h:::::ho:::::::::::::::oo:::::::::::::::o tt::::::::::::::t e::::::::eeeeeeee r:::::r " << endl;

cout << " S:::::::::::::::SS h:::::h h:::::h oo:::::::::::oo oo:::::::::::oo tt:::::::::::tt ee:::::::::::::e r:::::r " << endl;

cout << " SSSSSSSSSSSSSSS hhhhhhh hhhhhhh ooooooooooo ooooooooooo ttttttttttt eeeeeeeeeeeeee rrrrrrr " << endl;

}

void logo()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 4);

cout << endl

<< endl

<< endl;

cout << " $$$$$$\\ $$$$$$$\\ $$$$$$\\ $$$$$$\\ $$$$$$$$\\ $$$$$$\\ $$\\ $$\\ $$$$$$\\ $$$$$$\\ $$$$$$$$\\ $$$$$$$$\\ $$$$$$$\\ " << endl;

cout << " $$ \_\_$$\\ $$ \_\_$$\\ $$ \_\_$$\\ $$ \_\_$$\\ $$ \_\_\_\_\_| $$ \_\_$$\\ $$ | $$ |$$ \_\_$$\\ $$ \_\_$$\\\_\_$$ \_\_|$$ \_\_\_\_\_|$$ \_\_$$\\ " << endl;

cout << " $$ / \\\_\_|$$ | $$ |$$ / $$ |$$ / \\\_\_|$$ | $$ / \\\_\_|$$ | $$ |$$ / $$ |$$ / $$ | $$ | $$ | $$ | $$ |" << endl;

cout << " \\$$$$$$\\ $$$$$$$ |$$$$$$$$ |$$ | $$$$$\\ \\$$$$$$\\ $$$$$$$$ |$$ | $$ |$$ | $$ | $$ | $$$$$\\ $$$$$$$ |" << endl;

cout << " \\\_\_\_\_$$\\ $$ \_\_\_\_/ $$ \_\_$$ |$$ | $$ \_\_| \\\_\_\_\_$$\\ $$ \_\_$$ |$$ | $$ |$$ | $$ | $$ | $$ \_\_| $$ \_\_$$< " << endl;

cout << " $$\\ $$ |$$ | $$ | $$ |$$ | $$\\ $$ | $$\\ $$ |$$ | $$ |$$ | $$ |$$ | $$ | $$ | $$ | $$ | $$ |" << endl;

cout << " \\$$$$$$ |$$ | $$ | $$ |\\$$$$$$ |$$$$$$$$\\ \\$$$$$$ |$$ | $$ | $$$$$$ | $$$$$$ | $$ | $$$$$$$$\\ $$ | $$ |" << endl;

cout << " \\\_\_\_\_\_\_/ \\\_\_| \\\_\_| \\\_\_| \\\_\_\_\_\_\_/ \\\_\_\_\_\_\_\_\_| \\\_\_\_\_\_\_/ \\\_\_| \\\_\_| \\\_\_\_\_\_\_/ \\\_\_\_\_\_\_/ \\\_\_| \\\_\_\_\_\_\_\_\_|\\\_\_| \\\_\_|" << endl

<< endl

<< endl;

}

void gameover()

{

system("cls");

setColor(10);

cout << "\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\t\t\t \* Space Shooter Game \*" << endl;

cout << "\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl

<< endl

<< endl;

setColor(4);

cout << "\t\t\t-----------------------------------------------------" << endl;

cout << "\t\t\t-----------------------Game Over---------------------" << endl;

cout << "\t\t\t-----------------------------------------------------" << endl

<< endl

<< endl;

if (checkBestScore(score))

{

setColor(2);

cout << "\t\t\t You reached the best score : " << score << endl

<< endl;

}

else

{

setColor(11);

cout << "\t\t\t Your score is : " << score << endl

<< endl;

}

setColor(15);

cout << "\t\t\t Press any key to continue." << endl;

}

void controlmenu()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 6);

gotoxy(WIN\_WIDTH + 8, 12);

cout << "Space Shooter";

gotoxy(WIN\_WIDTH + 9, 14);

cout << "-----------";

gotoxy(WIN\_WIDTH + 9, 16);

cout << "-----------";

gotoxy(WIN\_WIDTH + 9, 18);

cout << "-----------";

gotoxy(WIN\_WIDTH + 10, 27);

cout << "Control";

gotoxy(WIN\_WIDTH + 10, 28);

cout << "---------";

gotoxy(WIN\_WIDTH + 4, 29);

cout << "Right Arrow Key - Right";

gotoxy(WIN\_WIDTH + 4, 30);

cout << "Left Arrow Key - Left";

gotoxy(WIN\_WIDTH + 4, 31);

cout << "Upper Arrow Key - Up";

gotoxy(WIN\_WIDTH + 4, 32);

cout << "Lower Arrow Key - Down";

gotoxy(WIN\_WIDTH + 4, 33);

cout << "Spacebar - Shoot";

}

void Instructions()

{

int x = 40;

int y = 10;

system("Color 0E");

cout << endl

<< endl;

gotoxy(x + 15, y - 5);

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

gotoxy(x + 15, y - 4);

cout << " Space Shooter Game" << endl;

gotoxy(x + 15, y - 3);

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\n\n";

gotoxy(x, y);

cout << "Instructions" << endl;

cout << "\n\n";

gotoxy(x, y + 1);

cout << "1. Use the arrow keys to move your spaceship left and right." << endl;

gotoxy(x, y + 2);

cout << "2. Press the space bar to fire bullets at the enemy spaceships." << endl;

gotoxy(x, y + 3);

cout << "3. Avoid getting hit by enemy bullets or colliding with their spaceships." << endl;

gotoxy(x, y + 4);

cout << "4. Shoot down as many enemy spaceships as possible to score points." << endl;

gotoxy(x, y + 5);

cout << "5. The game will display your score on the right side of the console." << endl;

gotoxy(x, y + 6);

cout << "6. Press any key to go back to the main screen." << endl;

}

void loading()

{

// this is just a loading bar

gotoxy(87, 26);

cout << "LOADING....";

int x, y;

// HORIZONTAL

//--UP

x = 87;

y = 27;

gotoxy(x, y);

for (int z = 0; z <= 30; z++)

{

cout << static\_cast<char>(205);

}

//--Down

y = 29;

gotoxy(x, y);

for (int z = 0; z <= 30; z++)

{

cout << static\_cast<char>(205);

}

// VERTICAL

//--Right

x = 117;

gotoxy(x, y);

for (int y = 28; y < 29; y++)

{

gotoxy(x, y);

cout << static\_cast<char>(186);

}

//--Left

x = 87;

gotoxy(x, y);

for (int y = 28; y < 29; y++)

{

gotoxy(x, y);

cout << static\_cast<char>(186);

}

//-------------------------CORNERS-----------------------------------------

// UP -LEFT

gotoxy(87, 27);

cout << static\_cast<char>(201);

// UP - RIGHT

gotoxy(117, 27);

cout << static\_cast<char>(187);

// DOWN -LEFT

gotoxy(87, 29);

cout << static\_cast<char>(200);

// DOWN RIGHT.

gotoxy(117, 29);

cout << static\_cast<char>(188);

// LOADING

int cargar;

for (cargar = 88; cargar < 117; cargar++)

{

gotoxy(cargar, 28);

cout << static\_cast<char>(178);

Sleep(50);

}

}

void history()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 6);

// a little story of the game

int x, y;

x = 48;

y = 3;

gotoxy(x, y);

cout << "WELCOME TO SPACE SHOOTER GAME";

x = 12;

y = 5;

gotoxy(x, y);

cout << "The galaxy is under Attack, an alien race called 'The Yincas' has invaded our solar";

gotoxy(x, y + 1);

cout << " system and it is taking the earth's resources. If we don't do something the human";

gotoxy(x, y + 2);

cout << "race will disappear.";

gotoxy(x, y + 4);

cout << "YOUR MISION:";

gotoxy(x, y + 5);

cout << "Destroy some yinca ships to show them we are not afraid";

gotoxy(x, y + 7);

cout << "VEHICLE:";

gotoxy(x, y + 8);

cout << "You've got the StarFighter spaceship, use the left,right,up and down keys to move it";

gotoxy(x, y + 10);

cout << "TECHNOLOGY:";

gotoxy(x, y + 11);

cout << "We have a laser gun, to shoot it use the spaceBar";

x = 33;

y = 20;

gotoxy(x, y);

cout << "OK SOLDIER, HIT ENTER TO JUMP INTO THE ACTION!!!";

loading(); // LOADING BAR

getch(); // WE HIT ANY KEY TO CLEAN THE SCREEN

system("cls");

}

// ---Collision---

void bulletcollisionwidenemy()

{

for (int x = 0; x < bulletcount; x++)

{

if (isBulletActice[x] == true)

{

if (bullety[x] == E1Y + 5 && (bulletx[x] == E1X || bulletx[x] == E1X + 1 || bulletx[x] == E1X + 2 || bulletx[x] == E1X + 3 || bulletx[x] == E1X + 4 || bulletx[x] == E1X + 5 || bulletx[x] == E1X + 6))

{

eraseenemy1();

addscore();

updatescore();

E1X = rand() % WIN\_WIDTH;

E1Y = 2;

// char next= getcharAtxy(E1X,E1Y);

// if( next != ' ' )

// {

// E1X = rand()%WIN\_WIDTH;

// E1Y = rand()%WIN\_WIDTH;

// }

moveenemy1();

}

if (bullety[x] == E2Y + 5 && (bulletx[x] == E2X || bulletx[x] == E2X + 1 || bulletx[x] == E2X + 2 || bulletx[x] == E2X + 3 || bulletx[x] == E2X + 4))

{

eraseenemy2();

addscore();

updatescore();

// E2X = 40;

E2X = rand() % WIN\_WIDTH;

E2Y = 10;

moveenemy2();

}

if (bullety[x] == E3Y + 4 && (bulletx[x] == E3X || bulletx[x] == E3X + 1 || bulletx[x] == E3X + 2 || bulletx[x] == E3X + 3))

{

eraseenemy3();

addscore();

updatescore();

// E3X = 108;

E3X = rand() % WIN\_WIDTH;

E3Y = 4;

moveenemy3();

}

if (bullety[x] == E4Y + 6 && (bulletx[x] == E4X || bulletx[x] == E4X + 1 || bulletx[x] == E4X + 2 || bulletx[x] == E4X + 3 || bulletx[x] == E4X + 4 || bulletx[x] == E4X + 5 || bulletx[x] == E4X + 6))

{

eraseenemy4();

addscore();

updatescore();

// E4X = 60;

E4X = rand() % WIN\_WIDTH;

E4Y = 9;

moveenemy4();

}

if (bullety[x] == E5Y + 4 && (bulletx[x] == E5X || bulletx[x] == E5X + 1 || bulletx[x] == E5X + 2 || bulletx[x] == E5X + 3))

{

eraseenemy5();

addscore();

// E5X = 90;

E5X = rand() % WIN\_WIDTH;

E5Y = 5;

updatescore();

moveenemy5();

}

}

}

}

// ---Score Functions---

void updatescore()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 15);

gotoxy(WIN\_WIDTH + 9, 15);

cout << " Score: " << score << endl;

}

void addscore()

{

score = score + 5;

}

bool checkBestScore(int score)

{

for (int i = 0; i < score\_count; i++)

{

if (score < Score[i])

{

return false;

}

}

return true;

}

// ----Lives----

void updatelives()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 15);

gotoxy(WIN\_WIDTH + 9, 17);

cout << " Lives: " << lives << endl;

}

void loselife()

{

erasespaceship();

PX = 2;

gotoxy(PX, PY);

printspaceship();

lives--;

if (lives == 0)

{

gamerunning = false;

}

}

// ----File Handling----

void storescore(int final\_score)

{

fstream file;

file.open("scores.txt", ios::app);

file << final\_score << endl;

file.close();

}

void readscore()

{

int best\_score = 0;

fstream file;

file.open("scores.txt", ios::in);

while (file >> best\_score)

{

Score[score\_count] = best\_score;

score\_count++;

}

file.close();

}

void writeBorderToFile()

{

fstream file;

file.open("border.txt", ios::out); // create file object and open file

for (int i = 0; i < SCREEN\_WIDTH; i++)

{

file << "+";

}

for (int i = 0; i < SCREEN\_WIDTH; i++)

{

file << "+";

}

for (int i = 0; i < SCREEN\_HEIGHT; i++)

{

file << "+";

file << "+";

}

for (int i = 0; i < SCREEN\_HEIGHT; i++)

{

file << "+";

}

file.close(); // close file

}

void readBorderFromFile()

{

fstream file("border.txt", ios::in); // create file object and open file

if (file.is\_open())

{ // check if file is successfully opened

string line;

while (getline(file, line))

{ // read file line by line

cout << line << endl; // print each line to console

}

file.close(); // close file

}

}

// For Setting Colors

void setColor(int colorCode)

{

HANDLE consoleHandle = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(consoleHandle, colorCode);

}

void playBeep()

{

Beep(440, 500); // play a 440 Hz tone for 500 milliseconds

}