Bro vs Aliens

Session: 2023 – 2027

Submitted by:

Abu Tayyab 2023-CS-54

Supervised by:

Dr. Muhammad Awais Hassan

Course:

CSC-102 Programming Fundamentals

Department of Computer Science

University of Engineering and Technology

Lahore Pakistan

Contents

Table of wireframes:

Bro vs Aliens

1) Background Story:

Bro, our cool space explorer, got lost in a super weird galaxy filled with unfriendly aliens. Now, he needs to shoot his way through these space bullies to find his way back home to Earth. Armed with a cool spaceship, Bro faces tough fights, dodges asteroids, and collects power-ups to beat the big boss alien, General Zorgon.

Along the journey, Bro finds special space gadgets that make his ship even cooler. It's like a space adventure where Bro needs to be a hero and defeat all the aliens to unlock the path back to Earth. Will Bro be able to do it? Let's help Bro make an epic comeback in his Space Rescue!

2) Description of game:

"Bro's Galactic Odyssey" thrusts you into space, maneuvering Bro with arrows and firing at alien foes with the Space key. Avoid deadly mines ('@') and collect power-ups ('o'). Close encounters with enemies decrease health, so strategize your moves. Gravity affects bullets, limiting their range to 20 points. Clear levels to confront General Zorgon, the ultimate antagonist. Successfully defeat him to unlock the spaceship and return home. An exciting blend of strategy and action awaits – embark on Bro's journey now! Press any key to commence the adventure

3) Game Character Description:

1) Player:

Meet Bro, an intrepid space adventurer stranded amidst alien adversaries. Equipped with courage and an arsenal of space weaponry, Bro relies on your strategic guidance. Utilize arrow keys to navigate the cosmic terrain and the Space key to unleash devastating firepower upon extraterrestrial foes. Bro's mission is clear: annihilate enemies, avoid treacherous mines, and collect power-ups to enhance his chances of survival. Beware of close encounters with enemies, as they pose a threat to Bro's health.

2) Enemies:

a) Alpha:

Move in coordinated waves, changing direction abruptly. Their collective gravitational pull makes them challenging to predict.

b) Beta:

Will move fastly and try to take his health away. It moves horizontally.

c) Gamma:

This enemy also moves horizontaly and this enemy is not very hard to beat.

d) Dalta:

This is a special type of enemy which is very hard to aim and kill its movement is Diagonal.

e) General Zorgon:

This is the last Enemy that you have to kill in order to go back home but it is very hard to kill General because its movement is vertical.

4)Game Objects Description:

a. bomb:

Bomb is very important in this game when player touch the bomb its health decreases.

b. Score pill:

Score pill is very useful because when player catches it the score increases.

c. Walls:

The field is divided into different sections and each level has fixed walls.

5)Rules and Regulations:

1. Player:

Bro can move in any direction he wants and can shoots fires horizontally. If bro touch the score pill then score will increase and it touches the bomb its health decreases bro has 3 totals lives and each life has 100 health.

2. Enemies:

There are total 5 enemies and in which Alpha, beta, gamma are the enemies with horizontal movement and Dalta can move diagonally and the General can move vertically

When even enemy touches with the player 1 life of player is lost.

Every enemy will destroy after taking different bullets.

6)Objective of the Game:

The objective of the game is to kill and the enemy so that bro can go to earth his home.

7)Wire Frames:

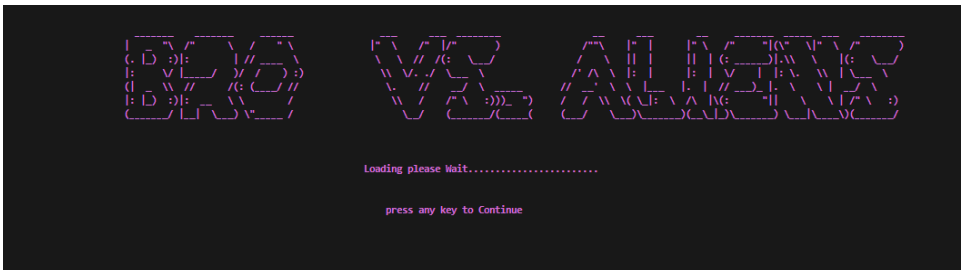

Figure 1 starting of game

Figure 2 Main Menu

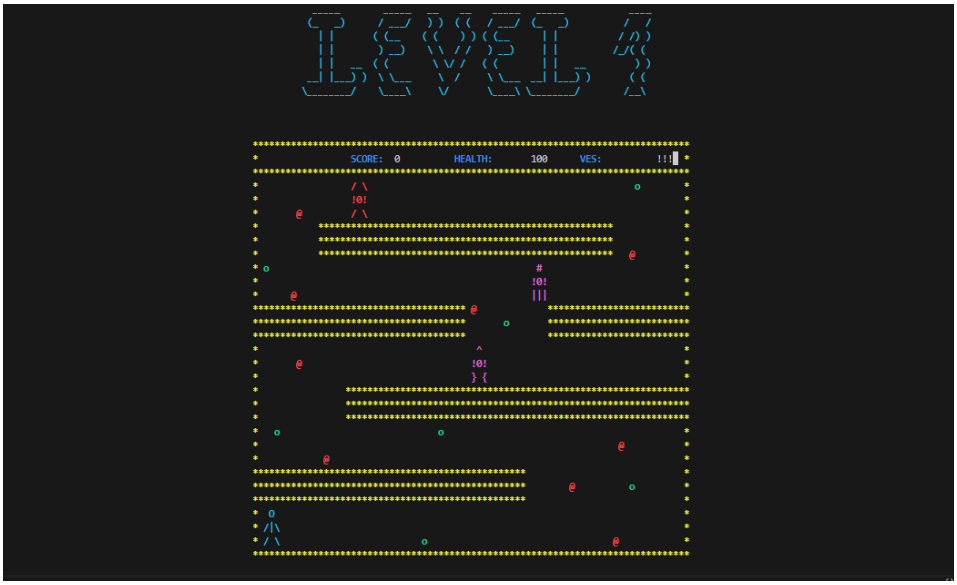

Figure 3Level 1 Maze

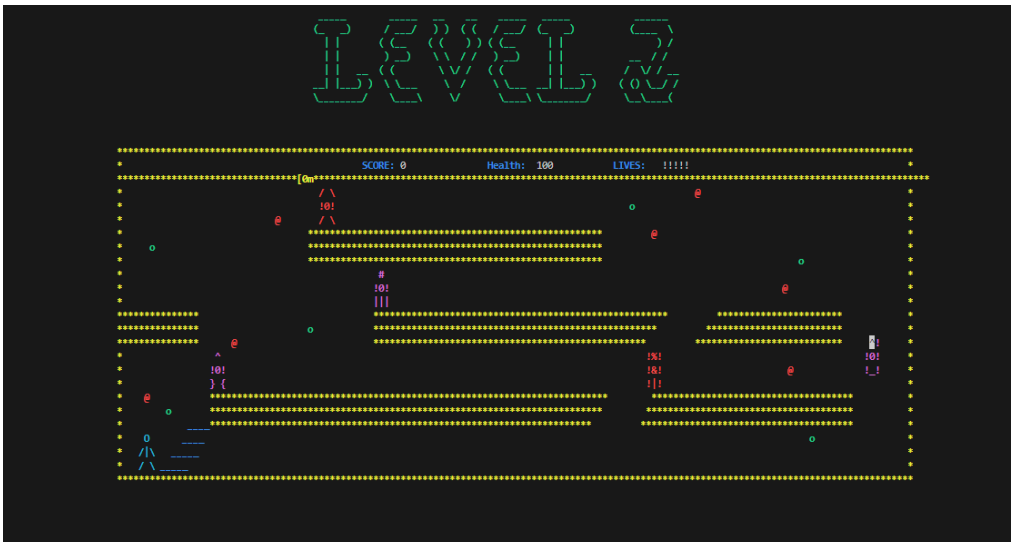

Figure 4Level 2 Maze

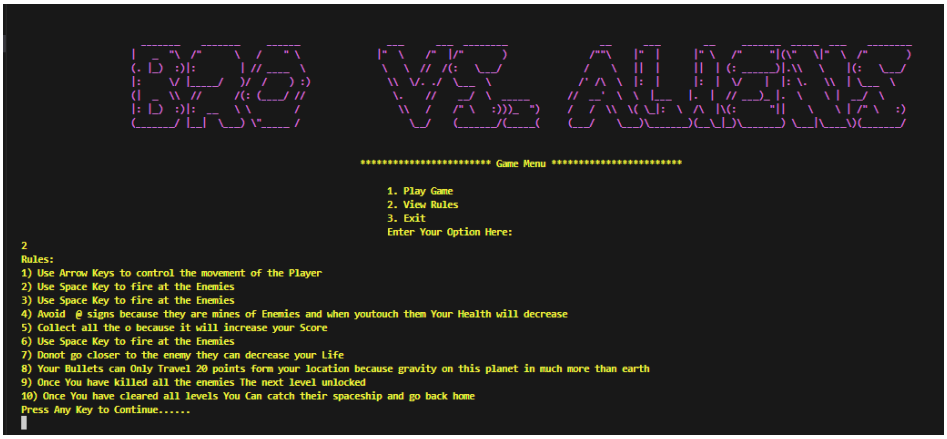

Figure 5Rules

8)Prototypes:

- char getCharAtxy(short int x, short int y);
-
- void print_Name();
- void print_BroVsAliean();

- void print_level1();
- void print_Leval2();
- void print_GameOver();
- void Print_Rules();
-
- string game_Menu();
- void fill_2DArray(char array[][130]);
- void print_2DArray(char array[][130]);
- void fill_2DArray_Level2(char array[][170]);
- void print_2DArray_Level2(char array[][170]);
- void Gotoxy(int x, int y);
- void hideCursor();
-
- void Print_player(int player_X, int player_Y);
- void remove_player(int player_X, int player_Y);
- void movePlayerRight(int &player_X, int &player_Y, int &score, int &health);
- void movePlayerLeft(int &player_X, int &player_Y, int &score, int &health);
- void movePlayerUp(int &player_X, int &player_Y, int &score, int &health);
- void movePlayerdown(int &player_X, int &player_Y, int &score, int &health);
- void movePlayerRight_Level2(int &player_X, int &player_Y, int &score, int &health);
- void movePlayerLeft_Level2(int &player_X, int &player_Y, int &score, int &health);
- void movePlayerUp_Level2(int &player_X, int &player_Y, int &score, int &health);
- void movePlayerdown_Level2(int &player_X, int &player_Y, int &score, int &health);
-
- void print_Enemy1(int enemy_OneX, int enemy_TwoY);
- void remove_Enemy1(int enemy_OneX, int enemy_TwoY);
- void move_enemy1(string direction_Enemy_1, int &enemy_OneX, int &enemy_OneY, int &health);
- string change_Direction_Enemy1(string direction_Enemy_1, int enemy_OneX, int enemy_TwoY);
- string change_Direction_Enemy1_level2(string direction_Enemy_1, int enemy_OneX, int enemy_TwoY);
-
- void print_Enemy2(int enemy_TwoX, int enemy_TwoY);
- void remove_Enemy2(int enemy_TwoX, int enemy_TwoY);
- void move_enemy2(string direction_Enemy_2, int &enemy_TwoX, int &enemy_TwoY);
- string change_Direction_Enemy2(string direction_Enemy_2, int enemy_TwoX, int enemy_TwoY);

- string change_Direction_Enemy2_level2(string direction_Enemy_2, int enemy_TwoX, int enemy_TwoY);
-
- void print_Enemy3(int enemy_ThreeX, int enemy_ThreeY);
- void remove_Enemy3(int enemy_ThreeX, int enemy_ThreeY);
- void move_enemy3(string direction_Enemy_3, int &enemy_ThreeX, int &enemy_ThreeY);
- string change_Direction_Enemy3(string direction_Enemy_3, int enemy_ThreeX, int enemy_ThreeY);
- string change_Direction_Enemy3_level2(string direction_Enemy_3, int enemy_ThreeX, int enemy_ThreeY);
-
- void print_Enemy4(int enemy_FourX, int enemy_FourY);
- void remove_Enemy4(int enemy_FourX, int enemy_FourY);
- string change_Direction_Enemy4_level2(string direction_Enemy_4, int enemy_FourX, int enemy_FourY);
- void move_enemy4(string direction_Enemy_4, int &enemy_FourX, int &enemy_FourY);
-
- void print_Enemy5(int enemy_FiveX, int enemy_FiveY);
- void remove_Enemy5(int enemy_FiveX, int enemy_FiveY);
- string change_Direction_Enemy5_level2(string direction_Enemy_5, int enemy_FiveX, int enemy_FiveY);
- void move_enemy5(string direction_Enemy_5, int &enemy_FiveX, int &enemy_FiveY);
-
- void Score_Player(int player_X, int player_Y, int &score);
-
- void Fire_Bullet(int player_X, int player_Y, int Bullet_X[], int Bullet_Y[], bool bullet_check[]);
- void remove_Bullet(int player_X, int player_Y, int &Bullet_X, int &Bullet_Y);
- void move_Bullet_Right(int player_X, int player_Y, int Bullet_X[], int Bullet_Y[], bool bullet_check[]);
- bool available_bullet(bool &bullet_check, int player_X, int player_Y, int &Bullet_X, int &Bullet_Y);
- void checkBulletEnemyCollision(int Bullet_X, int Bullet_Y, int &score, bool bullet_check[], int bulletIndex, int &enemy1Counter, int &enemy2Counter, int &enemy3Counter, int &enemy4Counter, int &enemy5Counter, int enemy_OneX, int enemy_TwoX, int enemy_ThreeX, int enemy_FourX, int enemy_FiveX, int enemy_OneY, int enemy_TwoY, int enemy_ThreeY, int enemy_FourY, int enemy_FiveY);

- void player_Health(int player_X, int player_Y, int &health, int &lives);
- void print_Lives(int &lives);

9)Variables:

- const int maxBullets = 3;
- bool moveEnemy1 = true;
- bool moveEnemy2 = true;
- bool moveEnemy3 = true;
- bool moveEnemy4 = true;
- bool moveEnemy5 = true;
- int Bullet_X[maxBullets];
- int Bullet_Y[maxBullets];
- int player_X = 27, player_Y = 31;
- int enemy_OneX = 60, enemy_OneY = 13;
- int enemy_TwoX = 70, enemy_TwoY = 19;
- int enemy_ThreeX = 40, enemy_ThreeY = 25;
- int enemy_FourX = 140, enemy_FourY = 15;
- int enemy_FiveX = 160, enemy_FiveY = 15;
-
- int score;
- int health;
- int lives;
-
- bool game;
- bool bullet_check[maxBullets];
-
- score = 0;
- health = 100;
- lives = 5;
- game = true;
-
- string direction_Enemy_1;
- string direction_Enemy_2;
- string direction_Enemy_3;
- string direction_Enemy_4;
- string direction_Enemy_5;
-
- direction_Enemy_1 = "right";

- direction_Enemy_2 = "right";
- direction_Enemy_3 = "right";
- direction_Enemy_4 = "diagonaldown";
- direction_Enemy_5 = "down";
-
- int enemy1Counter = 0;
- int enemy2Counter = 0;
- int enemy3Counter = 0;
- int enemy4Counter = 0;
- int enemy5Counter = 0;

10)Data structures:

2D array is used print the maze and 1D array is used to move bullets along local variales to control the game.

11) Complete code:

```
#include <iostream>

#include <windows.h>

#include <conio.h>

#include <string>


using namespace std;


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


void print_Name();

void print_BroVsAliean();

void print_level1();

void print_Leval2();

void print_GameOver();

void Print_Rules();
```

```
string game_Menu();
void fill_2DArray(char array[][130]);
void print_2DArray(char array[][130]);
void fill_2DArray_Level2(char array[][170]);
void print_2DArray_Level2(char array[][170]);
void Gotoxy(int x, int y);
void hideCursor();

void Print_player(int player_X, int player_Y);
void remove_player(int player_X, int player_Y);
void movePlayerRight(int &player_X, int &player_Y, int &score, int &health);
void movePlayerLeft(int &player_X, int &player_Y, int &score, int &health);
void movePlayerUp(int &player_X, int &player_Y, int &score, int &health);
void movePlayerdown(int &player_X, int &player_Y, int &score, int &health);
void movePlayerRight_Level2(int &player_X, int &player_Y, int &score, int &health);
void movePlayerLeft_Level2(int &player_X, int &player_Y, int &score, int &health);
void movePlayerUp_Level2(int &player_X, int &player_Y, int &score, int &health);
void movePlayerdown_Level2(int &player_X, int &player_Y, int &score, int &health);

void print_Enemy1(int enemy_OneX, int enemy_TwoY);
void remove_Enemy1(int enemy_OneX, int enemy_TwoY);
void move_enemy1(string direction_Enemy_1, int &enemy_OneX, int &enemy_OneY, int
&health);
string change_Direction_Enemy1(string direction_Enemy_1, int enemy_OneX, int
enemy_TwoY);
string change_Direction_Enemy1_level2(string direction_Enemy_1, int enemy_OneX, int
enemy_TwoY);

void print_Enemy2(int enemy_TwoX, int enemy_TwoY);
```

```
void remove_Enemy2(int enemy_TwoX, int enemy_TwoY);

void move_enemy2(string direction_Enemy_2, int &enemy_TwoX, int &enemy_TwoY);

string change_Direction_Enemy2(string direction_Enemy_2, int enemy_TwoX, int
enemy_TwoY);

string change_Direction_Enemy2_level2(string direction_Enemy_2, int enemy_TwoX, int
enemy_TwoY);


void print_Enemy3(int enemy_ThreeX, int enemy_ThreeY);

void remove_Enemy3(int enemy_ThreeX, int enemy_ThreeY);

void move_enemy3(string direction_Enemy_3, int &enemy_ThreeX, int &enemy_ThreeY);

string change_Direction_Enemy3(string direction_Enemy_3, int enemy_ThreeX, int
enemy_ThreeY);

string change_Direction_Enemy3_level2(string direction_Enemy_3, int enemy_ThreeX, int
enemy_ThreeY);


void print_Enemy4(int enemy_FourX, int enemy_FourY);

void remove_Enemy4(int enemy_FourX, int enemy_FourY);

string change_Direction_Enemy4_level2(string direction_Enemy_4, int enemy_FourX, int
enemy_FourY);

void move_enemy4(string direction_Enemy_4, int &enemy_FourX, int &enemy_FourY);


void print_Enemy5(int enemy_FiveX, int enemy_FiveY);

void remove_Enemy5(int enemy_FiveX, int enemy_FiveY);

string change_Direction_Enemy5_level2(string direction_Enemy_5, int enemy_FiveX, int
enemy_FiveY);

void move_enemy5(string direction_Enemy_5, int &enemy_FiveX, int &enemy_FiveY);


void Score_Player(int player_X, int player_Y, int &score);


void Fire_Bullet(int player_X, int player_Y, int Bullet_X[], int Bullet_Y[], bool bullet_check[]);
```

```
void remove_Bullet(int player_X, int player_Y, int &Bullet_X, int &Bullet_Y);

void move_Bullet_Right(int player_X, int player_Y, int Bullet_X[], int Bullet_Y[], bool
bullet_check[]);

bool available_bullet(bool &bullet_check, int player_X, int player_Y, int &Bullet_X, int
&Bullet_Y);

void checkBulletEnemyCollision(int Bullet_X, int Bullet_Y, int &score, bool bullet_check[], int
bulletIndex, int &enemy1Counter, int &enemy2Counter, int &enemy3Counter, int
&enemy4Counter, int &enemy5Counter, int enemy_OneX, int enemy_TwoX, int
enemy_ThreeX, int enemy_FourX, int enemy_FiveX, int enemy_OneY, int enemy_TwoY, int
enemy_ThreeY, int enemy_FourY, int enemy_FiveY);

void player_Health(int player_X, int player_Y, int &health, int &lives);

void print_Lives(int &lives);


main()

{

   system("cls");

   print_Name();

   getch();

   const int maxBullets = 3;

   bool moveEnemy1 = true;

   bool moveEnemy2 = true;

   bool moveEnemy3 = true;

   bool moveEnemy4 = true;

   bool moveEnemy5 = true;

   string main_Option;

   while (true)

   {


      print_BroVsAliean();

      main_Option = game_Menu();
```

```
if (main_Option == "1")
{

    bool level1;
    bool level2;
    level1 = false;
    level2 = true;

    if (level1)
    {
        int Bullet_X[maxBullets];
        int Bullet_Y[maxBullets];
        int player_X = 51, player_Y = 37;
        int enemy_OneX = 61, enemy_OneY = 13;
        int enemy_TwoX = 60, enemy_TwoY = 19;
        int enemy_ThreeX = 69, enemy_ThreeY = 25;
        int enemy_FourX = 140, enemy_FourY = 5;
        int enemy_FiveX = 160, enemy_FiveY = 5;
        int score;
        int health;
        int lives;

        bool game;
        bool bullet_check[maxBullets];

        score = 0;
```

```
health = 100;
lives = 3;
game = true;

string direction_Enemy_1;
string direction_Enemy_2;
string direction_Enemy_3;

direction_Enemy_1 = "right";
direction_Enemy_2 = "right";
direction_Enemy_3 = "right";

int enemy1Counter = 0;
int enemy2Counter = 0;
int enemy3Counter = 0;
int enemy4Counter = 0;
int enemy5Counter = 0;

system("cls");
char array[31][130];
fill_2DArray(array);
print_level1();
print_2DArray(array);
Print_player(player_X, player_Y);
print_Enemy1(enemy_OneX, enemy_OneY);
print_Enemy2(enemy_TwoX, enemy_TwoY);
```

```
        for (int i = 0; i < maxBullets; ++i)
        {
            Bullet_X[i] = 0;
            Bullet_Y[i] = 0;
            bullet_check[i] = false;
        }

        while (game)
        {
            if (enemy1Counter < 3 && moveEnemy1)
            {
                move_enemy1(direction_Enemy_1, enemy_OneX, enemy_OneY,
enemy1Counter);
                direction_Enemy_1 = change_Direction_Enemy1(direction_Enemy_1,
enemy_OneX, enemy_OneY);
            }

            if (enemy2Counter < 3 && moveEnemy2)
            {
                move_enemy2(direction_Enemy_2, enemy_TwoX, enemy_TwoY);
                direction_Enemy_2 = change_Direction_Enemy2(direction_Enemy_2,
enemy_TwoX, enemy_TwoY);
            }

            if (enemy3Counter < 3 && moveEnemy3)
            {
                move_enemy3(direction_Enemy_3, enemy_ThreeX, enemy_ThreeY);
                direction_Enemy_3 = change_Direction_Enemy3(direction_Enemy_3,
enemy_ThreeX, enemy_ThreeY);
```

```
}

if (GetAsyncKeyState(VK_RIGHT))
{
   movePlayerRight(player_X, player_Y, score, health); // for right
}
else if (GetAsyncKeyState(VK_LEFT))
{
   movePlayerLeft(player_X, player_Y, score, health); // for left
}
else if (GetAsyncKeyState(VK_UP))
{
   movePlayerUp(player_X, player_Y, score, health); // for up
}

else if (GetAsyncKeyState(VK_DOWN))
{
   movePlayerdown(player_X, player_Y, score, health); // for down
}
if (GetAsyncKeyState(VK_SPACE))
{
   Fire_Bullet(player_X, player_Y, Bullet_X, Bullet_Y, bullet_check);
}

for (int i = 0; i < maxBullets; ++i)
{
   if (bullet_check[i])
```

```
            {

                checkBulletEnemyCollision(Bullet_X[i], Bullet_Y[i], score, bullet_check, i,
enemy1Counter, enemy2Counter, enemy3Counter, enemy4Counter, enemy5Counter,
enemy_OneX, enemy_TwoX, enemy_ThreeX, enemy_FourX, enemy_FiveX, enemy_OneY,
enemy_TwoY, enemy_ThreeY, enemy_FourY, enemy_FiveY);

                move_Bullet_Right(player_X, player_Y, Bullet_X, Bullet_Y, bullet_check);
            }
        }

        if (enemy1Counter >= 3)
        {
            moveEnemy1 = false;
            remove_Enemy1(enemy_OneX, enemy_OneY);
        }

        if (enemy2Counter >= 3)
        {
            moveEnemy2 = false;
            remove_Enemy2(enemy_TwoX, enemy_TwoY);
        }

        if (enemy3Counter >= 3)
        {
            moveEnemy3 = false;
            remove_Enemy3(enemy_ThreeX, enemy_ThreeY);
        }
```

```
Score_Player(player_X, player_Y, score);
player_Health(player_X, player_Y, health, lives);
print_Lives(lives);
if (lives <= 0)
{
   game = false;
   system("cls");
   print_GameOver();
   cout << "your Score is: " << score << endl;
   Sleep(1000);
   getch();
   break;
}
if (enemy1Counter > 3 && enemy2Counter > 3 && enemy3Counter > 3)
{
   level2 = true;
   cout << "completed level 1" << endl;
   getch();
   break;
}
else if ((!moveEnemy1) && (!moveEnemy2))
{
   level2 = true;
   cout << "completed level 1" << endl;
   getch();
   break;
}
```

```
        Sleep(5);
    }
}
if (level2)
{
    int Bullet_X[maxBullets];
    int Bullet_Y[maxBullets];
    int player_X = 27, player_Y = 31;
    int enemy_OneX = 60, enemy_OneY = 13;
    int enemy_TwoX = 70, enemy_TwoY = 19;
    int enemy_ThreeX = 40, enemy_ThreeY = 25;
    int enemy_FourX = 140, enemy_FourY = 15;
    int enemy_FiveX = 160, enemy_FiveY = 15;

    int score;
    int health;
    int lives;

    bool game;
    bool bullet_check[maxBullets];

    score = 0;
    health = 100;
    lives = 5;
    game = true;
```

```
string direction_Enemy_1;
string direction_Enemy_2;
string direction_Enemy_3;
string direction_Enemy_4;
string direction_Enemy_5;

direction_Enemy_1 = "right";
direction_Enemy_2 = "right";
direction_Enemy_3 = "right";
direction_Enemy_4 = "diagonaldown";
direction_Enemy_5 = "down";

int enemy1Counter = 0;
int enemy2Counter = 0;
int enemy3Counter = 0;
int enemy4Counter = 0;
int enemy5Counter = 0;

system("cls");
char array[25][170];
fill_2DArray_Level2(array);
print_Leval2();
print_2DArray_Level2(array);
Print_player(player_X, player_Y);
print_Enemy1(enemy_OneX, enemy_OneY);
print_Enemy2(enemy_TwoX, enemy_TwoY);
print_Enemy3(enemy_ThreeX, enemy_ThreeY);
```

```
        print_Enemy4(enemy_FourX, enemy_FourY);
        print_Enemy5(enemy_FiveX, enemy_FiveY);

        for (int i = 0; i < maxBullets; ++i)
        {
           Bullet_X[i] = 0;
           Bullet_Y[i] = 0;
           bullet_check[i] = false;
        }

        while (game)
        {
           if (enemy1Counter < 3 && moveEnemy1)
           {
              move_enemy1(direction_Enemy_1, enemy_OneX, enemy_OneY, health);
              direction_Enemy_1 = change_Direction_Enemy1_level2(direction_Enemy_1,
enemy_OneX, enemy_OneY);
           }

           if (enemy2Counter < 3 && moveEnemy2)
           {
              move_enemy2(direction_Enemy_2, enemy_TwoX, enemy_TwoY);
              direction_Enemy_2 = change_Direction_Enemy2_level2(direction_Enemy_2,
enemy_TwoX, enemy_TwoY);
           }

           if (enemy3Counter < 3 && moveEnemy3)
           {
```

```
            move_enemy3(direction_Enemy_3, enemy_ThreeX, enemy_ThreeY);
            direction_Enemy_3 = change_Direction_Enemy3_level2(direction_Enemy_3, enemy_ThreeX, enemy_ThreeY);
        }
        if (enemy4Counter < 3 && moveEnemy4)
        {
            move_enemy4(direction_Enemy_4, enemy_FourX, enemy_FourY);
            direction_Enemy_4 = change_Direction_Enemy4_level2(direction_Enemy_4, enemy_FourX, enemy_FourY);
        }
        if (enemy5Counter < 3 && moveEnemy5)
        {
            move_enemy5(direction_Enemy_5, enemy_FiveX, enemy_FiveY);
            direction_Enemy_5 = change_Direction_Enemy5_level2(direction_Enemy_5, enemy_FiveX, enemy_FiveY);
        }


        if (GetAsyncKeyState(VK_RIGHT))
        {
            movePlayerRight_Level2(player_X, player_Y, score, health); // for right
        }
        else if (GetAsyncKeyState(VK_LEFT))
        {
            movePlayerLeft_Level2(player_X, player_Y, score, health); // for left
        }
        else if (GetAsyncKeyState(VK_UP))
        {
            movePlayerUp_Level2(player_X, player_Y, score, health); // for up
        }
```

```
else if (GetAsyncKeyState(VK_DOWN))
{
   movePlayerdown_Level2(player_X, player_Y, score, health); // for down
}
if (GetAsyncKeyState(VK_SPACE))
{
   Fire_Bullet(player_X, player_Y, Bullet_X, Bullet_Y, bullet_check);
}

for (int i = 0; i < maxBullets; ++i)
{
   if (bullet_check[i])
   {
      checkBulletEnemyCollision(Bullet_X[i], Bullet_Y[i], score, bullet_check, i,
enemy1Counter, enemy2Counter, enemy3Counter, enemy4Counter, enemy5Counter,
enemy_OneX, enemy_TwoX, enemy_ThreeX, enemy_FourX, enemy_FiveX, enemy_OneY,
enemy_TwoY, enemy_ThreeY, enemy_FourY, enemy_FiveY);
      move_Bullet_Right(player_X, player_Y, Bullet_X, Bullet_Y, bullet_check);
   }
}

if (enemy1Counter >= 3)
{
   moveEnemy1 = false;
   remove_Enemy1(enemy_OneX, enemy_OneY);
}
if (enemy2Counter >= 3)
{
```

```
        moveEnemy2 = false;
        remove_Enemy2(enemy_TwoX, enemy_TwoY);
    }
    if (enemy3Counter >= 3)
    {
        moveEnemy3 = false;
        remove_Enemy3(enemy_ThreeX, enemy_ThreeY);
    }
    if (enemy4Counter >= 3)
    {
        moveEnemy4 = false;
        remove_Enemy4(enemy_FourX, enemy_FourY);
    }
    if (enemy5Counter >= 3)
    {
        moveEnemy5 = false;
        remove_Enemy5(enemy_FiveX, enemy_FiveY);
    }
    Score_Player(player_X, player_Y, score);
    player_Health(player_X, player_Y, health, lives);
    print_Lives(lives);
    if (lives <= 0)
    {
        game = false;
        system("cls");
        print_GameOver();
        cout << "your Score is: " << score << endl;
```

```
                Sleep(1000);
                getch();
                break;
            }
            Sleep(5);
        }
    }
    else if (main_Option == "2")
    {
        Print_Rules();
        continue;
    }
    else if (main_Option == "3")
    {
        break;
    }
    else
    {
        continue;
    }
  }
}
void print_Name()
{
  // ANSI escape codes for color
  Gotoxy(1, 5);
```

```
    // ANSI escape codes for color
    cout << "\033[1;35m"; // Set text color to bright magenta
    cout << "                 ________    ________    _______              ___      ___  _________
__      ___        __     ________  ______  ___    _________  \n";
    cout << "                |  _  \"\\  /\"      \\   /    \" \\            |\"  \\    /\"  |/\"       )            /\"\"\\
|\"  |      |\" \\   /\"     \"|(\\\"   \\\|\"  \\  /\"       ) \n";
    cout << "                (. |_)  :)|:       | // ____  \\            \\   \\  //  /(:   \\___/            /    \\   || |
||  | (: _______)|.\\\\   \\    |(:   \\___/  \n";
    cout << "                |:    \\/ |_____/   )/  /    ) :)            \\\\\  \\/. ./  \\___  \\             /' /\\  \\  |:  |
|:  |  \\/    |  |: \\.   \\\\  | \\___  \\    \n";
    cout << "                (|  _  \\\\  //      /(: (____/ //              \\.    //    __/  \\  _____       //  __'  \\
\\  |____   |.  |  // ___)_ |.  \\    \\ |  __/  \\   \n";
    cout << "                |: |_)  :)|:  __   \\ \\        /                \\\\\   /    /\" \\   :)))_  \")     /   /  \\\\\  \\(
\\_|:  \\  /\\  |\\(:      \"||    \\    \\ | /\" \\   :) \n";
    cout << "                (________/ |__|  \\___) \\\"_____ /                  \\__/    (________/(_____(
(___/    \\___)\\_________)(__\\_|_)\\_________) \\___|\\_____\\)(________/  \n";
    cout << "\n";
    cout << endl;
    cout << endl;
    cout << "                                                    Loading please Wait....................." <<
endl;
    cout << endl;
    cout << endl;
    Sleep(200);
    cout << "                                                      press any key to Continue" << endl;
    cout << "\033[0m"; // Reset text color
}
void print_BroVsAliean()
{
```

```
    Gotoxy(1, 5);


    // ANSI escape codes for color
    cout << "\033[1;35m"; // Set text color to bright magenta
    cout << "                 _______   _______   ______          ___     ___ _________
__     ___       __    ________ _____ ___   _________  \n";
    cout << "                |  _  \"\\  /\"      \\   /   \" \\           |\"  \\    /\"  |/\"       )            /\"\"\\\\
|\"  |      |\" \\   /\"     \"|(\\\"   \\\|\"  \\  /\"       ) \n";
    cout << "                (. |_)  :)|:       | // ____  \\           \\   \\  //  /(:   \\___/             /    \\   || |
||  | (: _______)|.\\\\   \\    |(:   \\___/  \n";
    cout << "                |:    \\/ |_____/   )/  /    ) :)           \\\\  \\/. ./  \\___  \\             /' /\\  \\  |:  |
|:  |  \\/    |  |: \\.   \\\\  | \\___  \\    \n";
    cout << "                (|  _  \\\\  //     /(: (____/ //             \\.    //    __/  \\  _____       //  __'  \\
\\  |___    |.  |  // ___)_ |.  \\    \\ |  __/  \\   \n";
    cout << "                |: |_)  :)|:  __   \\ \\       /               \\\\   /    /\" \\   :)))_  \")     /   /  \\\\  \\(
\\_|:  \\  /\\  |\\(:      \"||    \\    \\ | /\" \\   :) \n";
    cout << "                (________/ |__|  \\___) \\\"_____ /                 \\__/    (________/(_____(
(___/    \\___)\\_________)(__\\_|_)\\_________) \\____|\\_____\\)(________/  \n";
    cout << "\n";
    cout << endl;
}
void print_level1()
{
    cout << "\033[1;36m"; // Set text color to bright cyan
    cout << "                                          _____       _____   __    __    _____   _____
_____ \n";
    cout << "                                        (_   _)     / ___/   ) )  ( (   / ___/  (_   _)         /   /
\n";
    cout << "                                         | |      ( (__    ( (    ) ) ( (__      | |          / /) ) \n";
    cout << "                                         | |       ) __)    \\ \\  / /  ) __)     | |         /_/( (
\n";
```

```
    cout << "                                  | |  __  ( (       \\ \\/ /  ( (       | |  __       ) )
\n";
    cout << "                                 __| |___) )  \\ \\___     \\  /    \\ \\___  __| |___) )
( (  \n";
    cout << "                                \\_________/   \\____\\\    \\/      \\____\\\
\\_________/       /__\\ \n";
    cout << "\n";
    cout << "\033[0m"; // Reset text color
    cout << endl;
    cout << endl;
}
void print_Leval2()
{
    cout << "\033[1;32m"; // Set text color to bright green
    cout << "                                  _____       _____  __   __   _____  _____
_______   \n";
    cout << "                                 (_   _)     / ___/  ) )  ( (   / ___/ (_   _)
(____  \\  \n";
    cout << "                                   | |      ( (__   ( (   ) )  ( (__     | |             ) /
\n";
    cout << "                                   | |       ) __)   \\ \\  / /   ) __)    | |          __  / /
\n";
    cout << "                                   | |  __  ( (       \\ \\/ /  ( (       | |  __     /  \\/ /
__ \n";
    cout << "                                 __| |___) )  \\ \\___     \\  /    \\ \\___  __| |___) )
( () \\__/ / \n";
    cout << "                                \\_________/   \\____\\\    \\/      \\____\\\
\\_________/      \\__\\_____(  \n";
    cout << "\n";
    cout << "\033[0m"; // Reset text color
    cout << endl;
```

```
    cout << endl;
}

void fill_2DArray(char array[][130])
{
    const char *lines[] = {
        "
**********************************************************************************
***",
        "                              *          SCORE:          HEALTH:          LIVES:
*",
        "
**********************************************************************************
***",
        "                              *                                          o      *",
        "                              *                                                 *",
        "                              *     @                                            *",
        "                              *
**************************************************          *",
        "                              *
**************************************************          *",
        "                              *
**************************************************  @       *",
        "                              * o                                               *",
        "                              *                                                 *",
        "                              *     @                                           *",
        "                              ****************************************** @
**************************",
        "                              ******************************************      o
**************************",
```

```
    "                          ************************************
**************************",
    "                          *                                    *",
    "                          *    @                               *",
    "                          *                                    *",
    "                          *
************************************************************",
    "                          *
************************************************************",
    "                          *
************************************************************",
    "                          * o              o                   *",
    "                          *                            @       *",
    "                          *      @                             *",
    "
************************************************  *",
    "
************************************************    @      o      *",
    "
************************************************  *",
    "                          *                                    *",
    "                          *                                    *",
    "                          *              o             @       *",
    "
*************************************************************************
***",
  };

  for (int x = 0; x < 31; x++)
  {
    for (int y = 0; y < 130; y++)
```

```
        {
            array[x][y] = lines[x][y];
        }
    }
}
void print_2DArray(char array[][130])
{
    for (int x = 0; x < 31; x++)
    {
        for (int y = 0; y < 130; y++)
        {
            if (array[x][y] == '*')
            {
                cout << "\033[1;33m";
                cout << array[x][y];
                cout << "\033[0m";
            }
            else if (array[x][y] == '@')
            {
                cout << "\033[1;31m";
                cout << array[x][y];
                cout << "\033[0m";
            }
            else if (array[x][y] == 'o')
            {
                cout << "\033[1;32m";
                cout << array[x][y];
```

```
            cout << "\033[0m";
        }
        else
        {
            cout << "\033[1;34m";
            cout << array[x][y];
            cout << "\033[0m";
        }
    }
    cout << endl;
  }
}
string change_Direction_Enemy1(string direction_Enemy_1, int enemy_OneX, int
enemy_OneY)
{
  if (enemy_OneX == 61)
  {
    direction_Enemy_1 = "right";
  }

  if (enemy_OneX == 109)
  {
    direction_Enemy_1 = "left";
  }
  return direction_Enemy_1;
}
string change_Direction_Enemy2(string direction_Enemy_2, int enemy_TwoX, int
enemy_TwoY)
```

```
{
   if (enemy_TwoX == 60)
   {
      direction_Enemy_2 = "right";
   }

   if (enemy_TwoX == 125)
   {
      direction_Enemy_2 = "left";
   }
   return direction_Enemy_2;
}
string change_Direction_Enemy3(string direction_Enemy_3, int enemy_ThreeX, int
enemy_ThreeY)
{
   if (enemy_ThreeX == 69)
   {
      direction_Enemy_3 = "right";
   }

   if (enemy_ThreeX == 124)
   {
      direction_Enemy_3 = "left";
   }
   return direction_Enemy_3;
}
void fill_2DArray_Level2(char array[][170])
{
```

```
   const char *lines[] = {
      "
**********************************************************************************
**********************************************************************",
      "              *                          SCORE:          Health:          LIVES:
*",
      "
**********************************************************************************
**********************************************************************",
      "              *                                                                @
*",
      "              *                                                          o
*",
      "              *                @
*",
      "              *
**************************************************          @
*",
      "              *   o
**************************************************
o     *",
      "              *
**************************************************                    o
*",
      "              *
*",
      "              *
@             *",
      "              *
*",
      "              ***************
**********************************************************
**********************       *",
```

```
    "               **********              o
**************************************************
************************       *",

    "               **********     @
**************************************************
**************************       *",

    "               *
*",

    "               *
@             *",

    "               *
*",

    "               *   @
**********************************************************************
**********************************       *",

    "               *      o
**********************************************************************
**********************************       *",

    "               *
____**********************************************************************
**********************************       *",

    "               *         ____
o           *",

    "               *        _____
*",

    "               *      _____
*",

    "
**********************************************************************
**********************************************************************",
  };


  for (int x = 0; x < 25; x++)
  {
```

```
        for (int y = 0; y < 170; y++)
        {
            array[x][y] = lines[x][y];
        }
    }
}
void print_2DArray_Level2(char array[][170])
{
    for (int x = 0; x < 25; x++)
    {
        for (int y = 0; y < 170; y++)
        {
            // cout << array[x][y];
            if (array[x][y] == '*')
            {
                cout << "\033[1;33m";
                cout << array[x][y];
                cout << "\033[0m";
            }
            else if (array[x][y] == '@')
            {
                cout << "\033[1;31m";
                cout << array[x][y];
                cout << "\033[0m";
            }
            else if (array[x][y] == 'o')
            {
```

```
            cout << "\033[1;32m";
            cout << array[x][y];
            cout << "\033[0m";
        }
        else
        {
            cout << "\033[1;34m";
            cout << array[x][y];
            cout << "\033[0m";
        }
    }
    cout << endl;
  }
}
void Gotoxy(int x, int y) // Function defination
{
  COORD cordinates;
  cordinates.X = x;
  cordinates.Y = y;
  SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), cordinates);
}
char getCharAtxy(short int x, short int y) // Function defination
{

  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 Print_player(int player_X, int player_Y) // Function defination
{
    cout << "\033[1;36m";
    Gotoxy(player_X, player_Y);
    cout << " O " << endl;
    Gotoxy(player_X, player_Y + 1);
    cout << "/|\\" << endl;
    Gotoxy(player_X, player_Y + 2);
    cout << "/ \\" << endl;
    cout << "\033[0m";
}
void remove_player(int player_X, int player_Y) // Function defination
{
    Gotoxy(player_X, player_Y);
    cout << "   " << endl;
    Gotoxy(player_X, player_Y + 1);
    cout << "   " << endl;
    Gotoxy(player_X, player_Y + 2);
    cout << "   " << endl;
}
void movePlayerRight(int &player_X, int &player_Y, int &score, int &health) // Function
defination
```

```
{


   if ((getCharAtxy(player_X + 3, player_Y) == ' ' && getCharAtxy(player_X + 3, player_Y + 1) == ' ' && getCharAtxy(player_X + 3, player_Y + 2) == ' ') || (getCharAtxy(player_X + 3, player_Y) == 'o' || getCharAtxy(player_X + 3, player_Y + 1) == 'o' || getCharAtxy(player_X + 3, player_Y + 2) == 'o') || (getCharAtxy(player_X + 3, player_Y) == '!' || getCharAtxy(player_X + 3, player_Y + 1) == '!' || getCharAtxy(player_X + 3, player_Y + 2) == '!') || (getCharAtxy(player_X + 3, player_Y) == '@' || getCharAtxy(player_X + 3, player_Y + 1) == '@' || getCharAtxy(player_X + 3, player_Y + 2) == '@') || (getCharAtxy(player_X + 3, player_Y) == '/' || getCharAtxy(player_X + 3, player_Y + 1) == '/' || getCharAtxy(player_X + 3, player_Y + 2) == '/'))

   {

      if (getCharAtxy(player_X + 3, player_Y) == 'o' || getCharAtxy(player_X + 3, player_Y + 1) == 'o' || getCharAtxy(player_X + 3, player_Y + 2) == 'o')

      {

         score++;

      }

      if (getCharAtxy(player_X + 3, player_Y) == '@' || getCharAtxy(player_X + 3, player_Y + 1) == '@' || getCharAtxy(player_X + 3, player_Y + 2) == '@')

      {

         health = health - 20;

      }

      remove_player(player_X, player_Y);

      player_X = player_X + 1;

      if (getCharAtxy(player_X + 3, player_Y) == '!' || getCharAtxy(player_X + 3, player_Y + 1) == '!' || getCharAtxy(player_X + 3, player_Y + 2) == '!')

      {

         player_X = 51;

         player_Y = 37;

         health = health - 100;

      }
```

```
        Print_player(player_X, player_Y);
    }
}
void movePlayerRight_Level2(int &player_X, int &player_Y, int &score, int &health) // Function defination
{

    if ((getCharAtxy(player_X + 3, player_Y) == ' ' && getCharAtxy(player_X + 3, player_Y + 1) == ' ' && getCharAtxy(player_X + 3, player_Y + 2) == ' ') || (getCharAtxy(player_X + 3, player_Y) == 'o' || getCharAtxy(player_X + 3, player_Y + 1) == 'o' || getCharAtxy(player_X + 3, player_Y + 2) == 'o') || (getCharAtxy(player_X + 3, player_Y) == '!' || getCharAtxy(player_X + 3, player_Y + 1) == '!' || getCharAtxy(player_X + 3, player_Y + 2) == '!') || (getCharAtxy(player_X + 3, player_Y) == '@' || getCharAtxy(player_X + 3, player_Y + 1) == '@' || getCharAtxy(player_X + 3, player_Y + 2) == '@') || (getCharAtxy(player_X + 3, player_Y) == '/' || getCharAtxy(player_X + 3, player_Y + 1) == '/' || getCharAtxy(player_X + 3, player_Y + 2) == '/'))
    {
        if (getCharAtxy(player_X + 3, player_Y) == 'o' || getCharAtxy(player_X + 3, player_Y + 1) == 'o' || getCharAtxy(player_X + 3, player_Y + 2) == 'o')
        {
            score++;
        }
        if (getCharAtxy(player_X + 3, player_Y) == '@' || getCharAtxy(player_X + 3, player_Y + 1) == '@' || getCharAtxy(player_X + 3, player_Y + 2) == '@')
        {
            health = health - 20;
        }
        remove_player(player_X, player_Y);
        player_X = player_X + 1;
        if (getCharAtxy(player_X + 3, player_Y) == '!' || getCharAtxy(player_X + 3, player_Y + 1) == '!' || getCharAtxy(player_X + 3, player_Y + 2) == '!')
```

```
        {
            player_X = 27;
            player_Y = 31;
            health = health - 100;
        }

        Print_player(player_X, player_Y);
    }
}
void movePlayerLeft(int &player_X, int &player_Y, int &score, int &health) // Function defination
{
    if ((getCharAtxy(player_X - 1, player_Y) == ' ' && getCharAtxy(player_X - 1, player_Y + 1) == ' ' && getCharAtxy(player_X - 1, player_Y + 2) == ' ') || (getCharAtxy(player_X - 1, player_Y) == 'o' || getCharAtxy(player_X - 1, player_Y + 1) == 'o' || getCharAtxy(player_X - 1, player_Y + 2) == 'o') || (getCharAtxy(player_X - 1, player_Y) == '!' || getCharAtxy(player_X - 1, player_Y + 1) == '!' || getCharAtxy(player_X - 1, player_Y + 2) == '!') || (getCharAtxy(player_X - 1, player_Y) == '@' || getCharAtxy(player_X - 1, player_Y + 1) == '@' || getCharAtxy(player_X - 1, player_Y + 2) == '@') || (getCharAtxy(player_X - 1, player_Y) == '/' || getCharAtxy(player_X - 1, player_Y + 1) == '/' || getCharAtxy(player_X - 1, player_Y + 2) == '/'))
    {
        if (getCharAtxy(player_X - 1, player_Y) == 'o' || getCharAtxy(player_X - 1, player_Y + 1) == 'o' || getCharAtxy(player_X - 1, player_Y + 2) == 'o')
        {
            score++;
        }
        if (getCharAtxy(player_X - 1, player_Y) == '@' || getCharAtxy(player_X - 1, player_Y + 1) == '@' || getCharAtxy(player_X - 1, player_Y + 2) == '@')
        {
            health = health - 20;
```

```
        }
        remove_player(player_X, player_Y);
        player_X = player_X - 1;
        if (getCharAtxy(player_X - 1, player_Y) == '!' || getCharAtxy(player_X - 1, player_Y + 1)
== '!' || getCharAtxy(player_X - 1, player_Y + 2) == '!')
        {
            player_X = 51;
            player_Y = 37;
            health = health - 100;
        }
        Print_player(player_X, player_Y);
    }
}
void movePlayerLeft_Level2(int &player_X, int &player_Y, int &score, int &health) //
Function defination
{
    if ((getCharAtxy(player_X - 1, player_Y) == ' ' && getCharAtxy(player_X - 1, player_Y + 1)
== ' ' && getCharAtxy(player_X - 1, player_Y + 2) == ' ') || (getCharAtxy(player_X - 1,
player_Y) == 'o' || getCharAtxy(player_X - 1, player_Y + 1) == 'o' || getCharAtxy(player_X - 1,
player_Y + 2) == 'o') || (getCharAtxy(player_X - 1, player_Y) == '!' || getCharAtxy(player_X -
1, player_Y + 1) == '!' || getCharAtxy(player_X - 1, player_Y + 2) == '!') ||
(getCharAtxy(player_X - 1, player_Y) == '@' || getCharAtxy(player_X - 1, player_Y + 1) ==
'@' || getCharAtxy(player_X - 1, player_Y + 2) == '@') || (getCharAtxy(player_X - 1, player_Y)
== '/' || getCharAtxy(player_X - 1, player_Y + 1) == '/' || getCharAtxy(player_X - 1, player_Y +
2) == '/'))
    {
        if (getCharAtxy(player_X - 1, player_Y) == 'o' || getCharAtxy(player_X - 1, player_Y + 1)
== 'o' || getCharAtxy(player_X - 1, player_Y + 2) == 'o')
        {
            score++;
        }
```

```
    if (getCharAtxy(player_X - 1, player_Y) == '@' || getCharAtxy(player_X - 1, player_Y + 1) == '@' || getCharAtxy(player_X - 1, player_Y + 2) == '@')
    {
        health = health - 20;
    }
    remove_player(player_X, player_Y);
    player_X = player_X - 1;
    if (getCharAtxy(player_X - 1, player_Y) == '!' || getCharAtxy(player_X - 1, player_Y + 1) == '!' || getCharAtxy(player_X - 1, player_Y + 2) == '!')
    {
        player_X = 27;
        player_Y = 31;
        health = health - 100;
    }
    Print_player(player_X, player_Y);
  }
}
void movePlayerUp(int &player_X, int &player_Y, int &score, int &health) // Function defination
{
  if ((getCharAtxy(player_X, player_Y - 1) == ' ' && getCharAtxy(player_X + 1, player_Y - 1) == ' ' && getCharAtxy(player_X + 2, player_Y - 1) == ' ') || (getCharAtxy(player_X, player_Y - 1) == 'o' || getCharAtxy(player_X + 1, player_Y - 1) == 'o' || getCharAtxy(player_X + 2, player_Y - 1) == 'o') || (getCharAtxy(player_X, player_Y - 1) == '!' || getCharAtxy(player_X + 1, player_Y - 1) == '!' || getCharAtxy(player_X + 2, player_Y - 1) == '!') || (getCharAtxy(player_X, player_Y - 1) == '@' || getCharAtxy(player_X + 1, player_Y - 1) == '@' || getCharAtxy(player_X + 2, player_Y - 1) == '@'))
  {
    if (getCharAtxy(player_X, player_Y - 1) == 'o' || getCharAtxy(player_X + 1, player_Y - 1) == 'o' || getCharAtxy(player_X + 2, player_Y - 1) == 'o')
    {
```

```
        score++;
    }
    if (getCharAtxy(player_X, player_Y - 1) == '@' || getCharAtxy(player_X + 1, player_Y - 1) == '@' || getCharAtxy(player_X + 2, player_Y - 1) == '@')
    {
        health = health - 20;
    }
    remove_player(player_X, player_Y);
    player_Y = player_Y - 1;
    if (getCharAtxy(player_X, player_Y - 1) == '!' || getCharAtxy(player_X + 1, player_Y - 1) == '!' || getCharAtxy(player_X + 2, player_Y - 1) == '!')
    {
        player_X = 51;
        player_Y = 37;
        health = health - 100;
    }
    Print_player(player_X, player_Y);
  }
}
void movePlayerUp_Level2(int &player_X, int &player_Y, int &score, int &health) // Function defination
{
  if ((getCharAtxy(player_X, player_Y - 1) == ' ' && getCharAtxy(player_X + 1, player_Y - 1) == ' ' && getCharAtxy(player_X + 2, player_Y - 1) == ' ') || (getCharAtxy(player_X, player_Y - 1) == 'o' || getCharAtxy(player_X + 1, player_Y - 1) == 'o' || getCharAtxy(player_X + 2, player_Y - 1) == 'o') || (getCharAtxy(player_X, player_Y - 1) == '!' || getCharAtxy(player_X + 1, player_Y - 1) == '!' || getCharAtxy(player_X + 2, player_Y - 1) == '!') || (getCharAtxy(player_X, player_Y - 1) == '@' || getCharAtxy(player_X + 1, player_Y - 1) == '@' || getCharAtxy(player_X + 2, player_Y - 1) == '@'))
  {
```

```
    if (getCharAtxy(player_X, player_Y - 1) == 'o' || getCharAtxy(player_X + 1, player_Y - 1) == 'o' || getCharAtxy(player_X + 2, player_Y - 1) == 'o')
    {
        score++;
    }
    if (getCharAtxy(player_X, player_Y - 1) == '@' || getCharAtxy(player_X + 1, player_Y - 1) == '@' || getCharAtxy(player_X + 2, player_Y - 1) == '@')
    {
        health = health - 20;
    }
    remove_player(player_X, player_Y);
    player_Y = player_Y - 1;
    if (getCharAtxy(player_X, player_Y - 1) == '!' || getCharAtxy(player_X + 1, player_Y - 1) == '!' || getCharAtxy(player_X + 2, player_Y - 1) == '!')
    {
        player_X = 27;
        player_Y = 31;
        health = health - 100;
    }
    Print_player(player_X, player_Y);
  }
}
void movePlayerdown(int &player_X, int &player_Y, int &score, int &health) // Function defination
{

  if ((getCharAtxy(player_X, player_Y + 3) == ' ' && getCharAtxy(player_X + 1, player_Y + 3) == ' ' && getCharAtxy(player_X + 2, player_Y + 3) == ' ') || (getCharAtxy(player_X, player_Y + 3) == 'o' || getCharAtxy(player_X + 1, player_Y + 3) == 'o' || getCharAtxy(player_X + 2, player_Y + 3) == 'o') || (getCharAtxy(player_X, player_Y + 3) == '!' ||
```

```
getCharAtxy(player_X + 1, player_Y + 3) == '!' || getCharAtxy(player_X + 2, player_Y + 3) == '!') || (getCharAtxy(player_X, player_Y + 3) == '@' || getCharAtxy(player_X + 1, player_Y + 3) == '@' || getCharAtxy(player_X + 2, player_Y + 3) == '@'))
   {
      if (getCharAtxy(player_X, player_Y + 3) == 'o' || getCharAtxy(player_X + 1, player_Y + 3) == 'o' || getCharAtxy(player_X + 2, player_Y + 3) == 'o')
      {
         score++;
      }
      if (getCharAtxy(player_X, player_Y + 3) == '@' || getCharAtxy(player_X + 1, player_Y + 3) == '@' || getCharAtxy(player_X + 2, player_Y + 3) == '@')
      {
         health = health - 20;
      }
      remove_player(player_X, player_Y);
      player_Y = player_Y + 1;
      if (getCharAtxy(player_X, player_Y + 3) == '!' || getCharAtxy(player_X + 1, player_Y + 3) == '!' || getCharAtxy(player_X + 2, player_Y + 3) == '!')
      {
         player_X = 51;
         player_Y = 37;
         health = health - 100;
      }

      Print_player(player_X, player_Y);
   }
}
void movePlayerdown_Level2(int &player_X, int &player_Y, int &score, int &health) // Function defination
{
```

```
    if ((getCharAtxy(player_X, player_Y + 3) == ' ' && getCharAtxy(player_X + 1, player_Y + 3) == ' ' && getCharAtxy(player_X + 2, player_Y + 3) == ' ') || (getCharAtxy(player_X, player_Y + 3) == 'o' || getCharAtxy(player_X + 1, player_Y + 3) == 'o' || getCharAtxy(player_X + 2, player_Y + 3) == 'o') || (getCharAtxy(player_X, player_Y + 3) == '!' || getCharAtxy(player_X + 1, player_Y + 3) == '!' || getCharAtxy(player_X + 2, player_Y + 3) == '!') || (getCharAtxy(player_X, player_Y + 3) == '@' || getCharAtxy(player_X + 1, player_Y + 3) == '@' || getCharAtxy(player_X + 2, player_Y + 3) == '@'))
    {
        if (getCharAtxy(player_X, player_Y + 3) == 'o' || getCharAtxy(player_X + 1, player_Y + 3) == 'o' || getCharAtxy(player_X + 2, player_Y + 3) == 'o')
        {
            score++;
        }
        if (getCharAtxy(player_X, player_Y + 3) == '@' || getCharAtxy(player_X + 1, player_Y + 3) == '@' || getCharAtxy(player_X + 2, player_Y + 3) == '@')
        {
            health = health - 20;
        }
        remove_player(player_X, player_Y);
        player_Y = player_Y + 1;
        if (getCharAtxy(player_X, player_Y + 3) == '!' || getCharAtxy(player_X + 1, player_Y + 3) == '!' || getCharAtxy(player_X + 2, player_Y + 3) == '!')
        {
            player_X = 27;
            player_Y = 31;
            health = health - 100;
        }

        Print_player(player_X, player_Y);
    }
```

```
}
void print_Enemy1(int enemy_OneX, int enemy_OneY) // Function defination
{
   cout << "\033[1;31m";
   Gotoxy(enemy_OneX, enemy_OneY);
   cout << "/ \\" << endl;
   Gotoxy(enemy_OneX, enemy_OneY + 1);
   cout << "!0!" << endl;
   Gotoxy(enemy_OneX, enemy_OneY + 2);
   cout << "/ \\" << endl;
   cout << "\033[0m";
}
void remove_Enemy1(int enemy_OneX, int enemy_OneY) // Function defination
{
   Gotoxy(enemy_OneX, enemy_OneY);
   cout << "   " << endl;
   Gotoxy(enemy_OneX, enemy_OneY + 1);
   cout << "   " << endl;
   Gotoxy(enemy_OneX, enemy_OneY + 2);
   cout << "   " << endl;
}
string change_Direction_Enemy1_level2(string direction_Enemy_1, int enemy_OneX, int
enemy_OneY)
{
   if (enemy_OneX == 60)
   {
      direction_Enemy_1 = "right";
   }
```

```
    if (enemy_OneX == 105)
    {
        direction_Enemy_1 = "left";
    }
    return direction_Enemy_1;
}
void move_enemy1(string direction_Enemy_1, int &enemy_OneX, int &enemy_OneY, int
&health)
{
    remove_Enemy1(enemy_OneX, enemy_OneY);

    if (direction_Enemy_1 == "right")
    {
        enemy_OneX = enemy_OneX + 1;
    }
    if (direction_Enemy_1 == "left")
    {
        enemy_OneX = enemy_OneX - 1;
    }

    print_Enemy1(enemy_OneX, enemy_OneY);
}
void print_Enemy2(int enemy_TwoX, int enemy_TwoY) // Function defination
{
    cout << "\033[1;35m";
    Gotoxy(enemy_TwoX, enemy_TwoY);
    cout << " #" << endl;
```

```
    Gotoxy(enemy_TwoX, enemy_TwoY + 1);
    cout << "!0!" << endl;
    Gotoxy(enemy_TwoX, enemy_TwoY + 2);
    cout << "|||" << endl;
    cout << "\033[0m";
}
void remove_Enemy2(int enemy_TwoX, int enemy_TwoY) // Function defination
{
    Gotoxy(enemy_TwoX, enemy_TwoY);
    cout << "   " << endl;
    Gotoxy(enemy_TwoX, enemy_TwoY + 1);
    cout << "   " << endl;
    Gotoxy(enemy_TwoX, enemy_TwoY + 2);
    cout << "   " << endl;
}
string change_Direction_Enemy2_level2(string direction_Enemy_2, int enemy_TwoX, int
enemy_TwoY)
{
    if (enemy_TwoX == 70)
    {
        direction_Enemy_2 = "right";
    }

    if (enemy_TwoX == 120)
    {
        direction_Enemy_2 = "left";
    }
    return direction_Enemy_2;
```

```
}
void move_enemy2(string direction_Enemy_2, int &enemy_TwoX, int &enemy_TwoY) // Function defination
{
   remove_Enemy2(enemy_TwoX, enemy_TwoY);
   if (direction_Enemy_2 == "right")
   {
      enemy_TwoX = enemy_TwoX + 1;
   }
   if (direction_Enemy_2 == "left")
   {
      enemy_TwoX = enemy_TwoX - 1;
   }
   print_Enemy2(enemy_TwoX, enemy_TwoY);
}
void print_Enemy3(int enemy_ThreeX, int enemy_ThreeY) // Function defination
{
   cout << "\033[1;35m";
   Gotoxy(enemy_ThreeX, enemy_ThreeY);
   cout << " ^ " << endl;
   Gotoxy(enemy_ThreeX, enemy_ThreeY + 1);
   cout << "!0!" << endl;
   Gotoxy(enemy_ThreeX, enemy_ThreeY + 2);
   cout << "} {" << endl;
   cout << "\033[0m";
}
void remove_Enemy3(int enemy_ThreeX, int enemy_ThreeY) // Function defination
{
```

```
    Gotoxy(enemy_ThreeX, enemy_ThreeY);
    cout << "   " << endl;
    Gotoxy(enemy_ThreeX, enemy_ThreeY + 1);
    cout << "   " << endl;
    Gotoxy(enemy_ThreeX, enemy_ThreeY + 2);
    cout << "   " << endl;
}
string change_Direction_Enemy3_level2(string direction_Enemy_3, int enemy_ThreeX, int enemy_ThreeY)
{
    if (enemy_ThreeX == 40)
    {
        direction_Enemy_3 = "right";
    }

    if (enemy_ThreeX == 110)
    {
        direction_Enemy_3 = "left";
    }
    return direction_Enemy_3;
}
void move_enemy3(string direction_Enemy_3, int &enemy_ThreeX, int &enemy_ThreeY) // Function defination
{
    remove_Enemy3(enemy_ThreeX, enemy_ThreeY);
    if (direction_Enemy_3 == "right")
    {
        enemy_ThreeX = enemy_ThreeX + 1;
```

```
    }
    if (direction_Enemy_3 == "left")
    {
        enemy_ThreeX = enemy_ThreeX - 1;
    }
    print_Enemy3(enemy_ThreeX, enemy_ThreeY);
}
void print_Enemy4(int enemy_FourX, int enemy_FourY) // Function defination
{
    cout << "\033[1;31m"; // Set text color to bright red

    Gotoxy(enemy_FourX, enemy_FourY);
    cout << "!%!" << endl;
    Gotoxy(enemy_FourX, enemy_FourY + 1);
    cout << "!&!" << endl;
    Gotoxy(enemy_FourX, enemy_FourY + 2);
    cout << "!|!" << endl;
    cout << "\033[0m";

}
void remove_Enemy4(int enemy_FourX, int enemy_FourY) // Function defination
{
    Gotoxy(enemy_FourX, enemy_FourY);
    cout << "   " << endl;
    Gotoxy(enemy_FourX, enemy_FourY + 1);
    cout << "   " << endl;
    Gotoxy(enemy_FourX, enemy_FourY + 2);
```

```
    cout << "    " << endl;
}
string change_Direction_Enemy4_level2(string direction_Enemy_4, int enemy_FourX, int enemy_FourY)
{
    if (enemy_FourX == 140)
    {
        direction_Enemy_4 = "diagonaldown";
    }

    if (enemy_FourY == 30)
    {
        direction_Enemy_4 = "diagonalup";
    }
    return direction_Enemy_4;
}
void move_enemy4(string direction_Enemy_4, int &enemy_FourX, int &enemy_FourY) // Function defination
{
    remove_Enemy4(enemy_FourX, enemy_FourY);
    if (direction_Enemy_4 == "diagonaldown")
    {
        enemy_FourY = enemy_FourY + 1;

        enemy_FourX = enemy_FourX - 2;
    }
    if (direction_Enemy_4 == "diagonalup")
    {
```

```
        enemy_FourY = enemy_FourY - 1;
        enemy_FourX = enemy_FourX + 2;
    }
    print_Enemy4(enemy_FourX, enemy_FourY);
}
void print_Enemy5(int enemy_FiveX, int enemy_FiveY) // Function defination
{
    cout << "\033[1;35m";

    Gotoxy(enemy_FiveX, enemy_FiveY);
    cout << "!^!" << endl;
    Gotoxy(enemy_FiveX, enemy_FiveY + 1);
    cout << "!0!" << endl;
    Gotoxy(enemy_FiveX, enemy_FiveY + 2);
    cout << "!_!" << endl;
    cout << "\033[0m";

}
void remove_Enemy5(int enemy_FiveX, int enemy_FiveY) // Function defination
{
    Gotoxy(enemy_FiveX, enemy_FiveY);
    cout << "   " << endl;
    Gotoxy(enemy_FiveX, enemy_FiveY + 1);
    cout << "   " << endl;
    Gotoxy(enemy_FiveX, enemy_FiveY + 2);
    cout << "   " << endl;
}
```

```
string change_Direction_Enemy5_level2(string direction_Enemy_5, int enemy_FiveX, int enemy_FiveY)
{
   if (enemy_FiveY == 15)
   {
      direction_Enemy_5 = "down";
   }
   if (enemy_FiveY == 30)
   {
      direction_Enemy_5 = "up";
   }
   return direction_Enemy_5;
}
void move_enemy5(string direction_Enemy_5, int &enemy_FiveX, int &enemy_FiveY) // Function defination
{
   remove_Enemy5(enemy_FiveX, enemy_FiveY);
   if (direction_Enemy_5 == "down")
   {
      enemy_FiveY = enemy_FiveY + 1;
   }
   if (direction_Enemy_5 == "up")
   {
      enemy_FiveY = enemy_FiveY - 1;
   }
   print_Enemy5(enemy_FiveX, enemy_FiveY);
}
void Score_Player(int player_X, int player_Y, int &score)
```

```
{
   Gotoxy(75, 11);
   cout << "         ";
   Gotoxy(75, 11);
   cout << score;
}
void Fire_Bullet(int player_X, int player_Y, int Bullet_X[], int Bullet_Y[], bool bullet_check[])
{
   for (int i = 0; i < 3; ++i)
   {
      if (!bullet_check[i])
      {
         Bullet_X[i] = player_X + 3;
         Bullet_Y[i] = player_Y + 1;
         bullet_check[i] = true;
         break; // Break out of the loop after firing a bullet
      }
   }
}
void remove_Bullet(int player_X, int player_Y, int &Bullet_X, int &Bullet_Y)
{
   if (Bullet_X < 170)
   {
      Gotoxy(Bullet_X, Bullet_Y);
      cout << " ";
   }
}
```

```
void move_Bullet_Right(int player_X, int player_Y, int Bullet_X[], int Bullet_Y[], bool bullet_check[])
{
   for (int i = 0; i < 3; ++i)
   {
      if (bullet_check[i])
      {
         remove_Bullet(player_X, player_Y, Bullet_X[i], Bullet_Y[i]); // Remove the existing bullet

         // Check if the next position is a space character
         char nextChar = getCharAtxy(Bullet_X[i] + 1, Bullet_Y[i]);

         // Move the bullet only if the next position is a space
         if (nextChar == ' ' && Bullet_X[i] <= player_X + 20)
         {
            // Move the bullet right
            Bullet_X[i] = Bullet_X[i] + 1;

            // Render the bullet at the new position
            Gotoxy(Bullet_X[i], Bullet_Y[i]);
            cout << ">";
         }
         else
         {
            // If the next position is not a space, mark the bullet as inactive
            bullet_check[i] = false;
         }
```

```
        }
    }
}
bool available_bullet(bool &bullet_check, int player_X, int player_Y, int &Bullet_X, int &Bullet_Y)
{
    if (!(bullet_check))
    {
        Bullet_X = player_X + 3;
        Bullet_Y = player_Y + 1;
        bullet_check = true;
    }

    return bullet_check;
}
void player_Health(int player_X, int player_Y, int &health, int &lives)
{
    if (health <= 0)
    {
        health = 0;
        lives--;
        health = 100;
    }
    Gotoxy(100, 11);
    cout << "          ";
    Gotoxy(100, 11);
    cout << health << endl;
}
```

```
void checkBulletEnemyCollision(int Bullet_X, int Bullet_Y, int &score, bool bullet_check[], int bulletIndex, int &enemy1Counter, int &enemy2Counter, int &enemy3Counter, int &enemy4Counter, int &enemy5Counter, int enemy_OneX, int enemy_TwoX, int enemy_ThreeX, int enemy_FourX, int enemy_FiveX, int enemy_OneY, int enemy_TwoY, int enemy_ThreeY, int enemy_FourY, int enemy_FiveY)
{
   bool hitEnemyOne = (Bullet_X >= enemy_OneX && Bullet_X <= enemy_OneX + 2 && Bullet_Y >= enemy_OneY && Bullet_Y <= enemy_OneY + 2);
   bool hitEnemyTwo = (Bullet_X >= enemy_TwoX && Bullet_X <= enemy_TwoX + 2 && Bullet_Y >= enemy_TwoY && Bullet_Y <= enemy_TwoY + 2);
   bool hitEnemyThree = (Bullet_X >= enemy_ThreeX && Bullet_X <= enemy_ThreeX + 2 && Bullet_Y >= enemy_ThreeY && Bullet_Y <= enemy_ThreeY + 2);
   bool hitEnemyFour = (Bullet_X >= enemy_FourX && Bullet_X <= enemy_FourX + 2 && Bullet_Y >= enemy_FourY && Bullet_Y <= enemy_FourY + 2);
   bool hitEnemyFive = (Bullet_X >= enemy_FiveX && Bullet_X <= enemy_FiveX + 2 && Bullet_Y >= enemy_FiveY && Bullet_Y <= enemy_FiveY + 2);

   if (hitEnemyOne || hitEnemyTwo || hitEnemyThree || hitEnemyFour || hitEnemyFive)
   {
      score = score + 10;

      if (hitEnemyOne)
      {
         enemy1Counter++;
      }
      else if (hitEnemyTwo)
      {
         enemy2Counter++;
      }
      else if (hitEnemyThree)
      {
```

```
            enemy3Counter++;
        }
        else if (hitEnemyFour)
        {
            enemy4Counter++;
        }
        else if (hitEnemyFive)
        {
            enemy5Counter++;
        }

        bullet_check[bulletIndex] = false;
    }
}
void print_Lives(int &lives)
{
    Gotoxy(121, 11);
    cout << "      ";

    for (int i = 0; i < lives; i++)
    {
        int x = i + 2;
        Gotoxy(121 + x, 11);
        cout << "!";
    }
}
void hideCursor()
```

```
{
    CONSOLE_CURSOR_INFO cursorInfo;
    HANDLE outputHandle = GetStdHandle(STD_OUTPUT_HANDLE);

    GetConsoleCursorInfo(outputHandle, &cursorInfo);
    cursorInfo.bVisible = false;
    SetConsoleCursorInfo(outputHandle, &cursorInfo);
}
string game_Menu()
{
    string option;
    system("cls");
    print_BroVsAliean();
    cout << "\033[1;33m";

    cout << "                                        ********************** Game Menu
**********************" << endl;
    cout << endl;
    cout << "                                            1. Play Game " << endl;
    cout << "                                            2. View Rules " << endl;
    cout << "                                            3. Exit " << endl;
    cout << "                                            Enter Your Option Here:  ";
    getline(cin >> ws, option);

    cout << "\033[0m";

    return option;
}
```

```
void print_GameOver()
{
   cout << "\033[1;31m"; // Set text color to bright red
   cout << "                                                 _____     ____      __   __     _____
____     __   __   _____   _______              ___ \n";
   cout << "                                                  / ___ \\   (   )     \\ \\ \\ / /    / ___/      / __ \\
) )  ( (   / ___/  (   __ \\       __    / _( \n";
   cout << "                                                 / /   \\_)  / /\\ \\     () \\/ ()   ( (__       / /  \\ \\  (
(    ) ) ( (__     ) (__) )     (__)  / /   \n";
   cout << "                                                ( (  ____   ( (__) )    / _  _ \\    ) __)     ( ()  () )
\\ \\ / /   ) __)   (    __/          ( (    \n";
   cout << "                                                ( ( (__  )   )   (    / / \\/ \\ \\ \\  ( (        ( ()  () )   \\
\\/ /   ( (       ) \\ \\ _          ( (    \n";
   cout << "                                                 \\ \\__/ /   /  /\\ \\  \\ \\  /_/     \\_\\   \\ \\___      \\
\\__/ /     \\ /     \\ \\___   ( ( \\ \\_))     __    \\ \\__ \n";
   cout << "                                                  \\_____/   /__(  )__\\ \\ (/          \\)   \\_____\\
\\_____/       \\/       \\_____\\  )_) \\__/     (__)   \\__\n";
   cout << "\n";
   cout << "\033[0m";
}
void Print_Rules()
{
   cout << "\033[1;33m",
      cout << "Rules: " << endl;
   cout << "1) Use Arrow Keys to control the movement of the Player " << endl;
   cout << "2) Use Space Key to fire at the Enemies " << endl;
   cout << "3) Use Space Key to fire at the Enemies " << endl;
   cout << "4) Avoid  @ signs because they are mines of Enemies and when youtouch them
Your Health will decrease" << endl;
   cout << "5) Collect all the o because it will increase your Score" << endl;
```

```
    cout << "6) Use Space Key to fire at the Enemies " << endl;
    cout << "7) Donot go closer to the enemy they can decrease your Life" << endl;
    cout << "8) Your Bullets can Only Travel 20 points form your location because gravity on this planet in much more than earth" << endl;
    cout << "9) Once You have killed all the enemies The next level unlocked" << endl;
    cout << "10) Once You have cleared all levels You Can catch their spaceship and go back home" << endl;
    cout << "Press Any Key to Continue....." << endl;
    cout << "\033[0m",

        getch();
}
```