TANK WARLORDS

Session: 2023 – 2027

Submitted by:

Muhammad Ahmad 2023-CS-69

Supervised by:

Dr. Awais Hassan

Course:

CSC-102 Programming Fundamentals

Department of Computer Science

University of Engineering and Technology

Lahore Pakistan

Important Instructions

Here you can find the major parts of your Game documentation

- **Short Description and Story Writing of your Game**

  My game is about tanks in which player can fire upon enemies moving in different manner. Its story is that a tank steps into a war field in which faces the high-flying jets of enemies. The player has to destroy all the enemies in order to win the game.

  In the chaotic aftermath of a global collapse, the world is dominated by rival factions vying for supremacy. Emerging from the chaos are the Tank Warlords, a coalition of elite commanders armed with advanced combat tanks. As a new recruit, players must climb the ranks, command a diverse fleet of customizable tanks, and lead strategic battles to establish dominance. The game combines intense multiplayer tank warfare with a gripping single-player storyline, unveiling the mysteries of a world on the brink. Only the most skilled commanders will prevail in the post-apocalyptic battleground of "Tank Warlords." Are you ready to lead your tank army to glory and become a legend among the Warlords?

- **Game Characters Description**

  Player:

  - The player is a tank which can fire upon the enemies .It can move in all the four directions. When its fire hits the enemy, the score increases. When it gets a bonus pill which appears randomly in the maze, its health increases.
  - **Enemies:**

    The enemies are fighter jets having a firing system which fires upon the tank when the tank comes in their range When their bullet hits the tank, the player's health decreases. When all the enemies are destroyed, the player wins the game. But if they damage the tank badly and the player's health becomes zero, the game is over.
  -

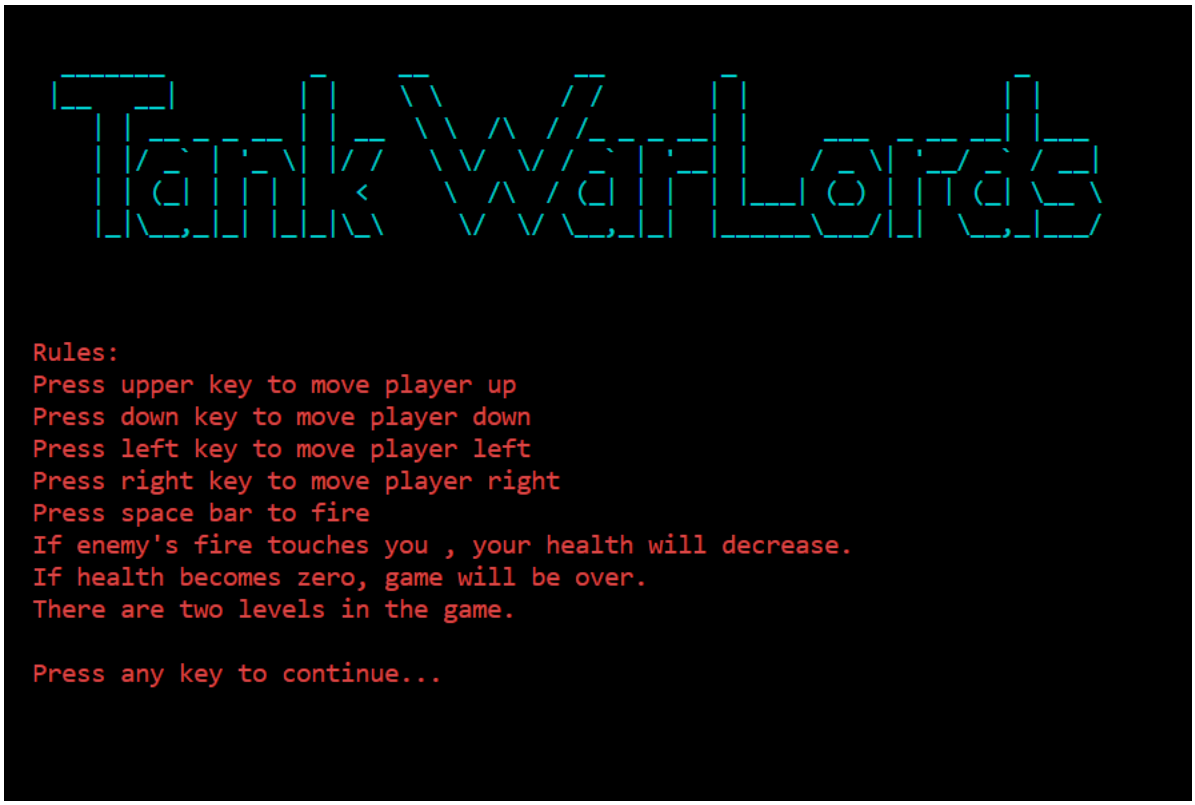
- **Game Objects Description**
    - **Maze:**

  The maze is a type of war field for the player and the enemies . In the maze, the player's coordinates ,score, health and the enemy's health is being displayed on the top right corner.
    - **Header**

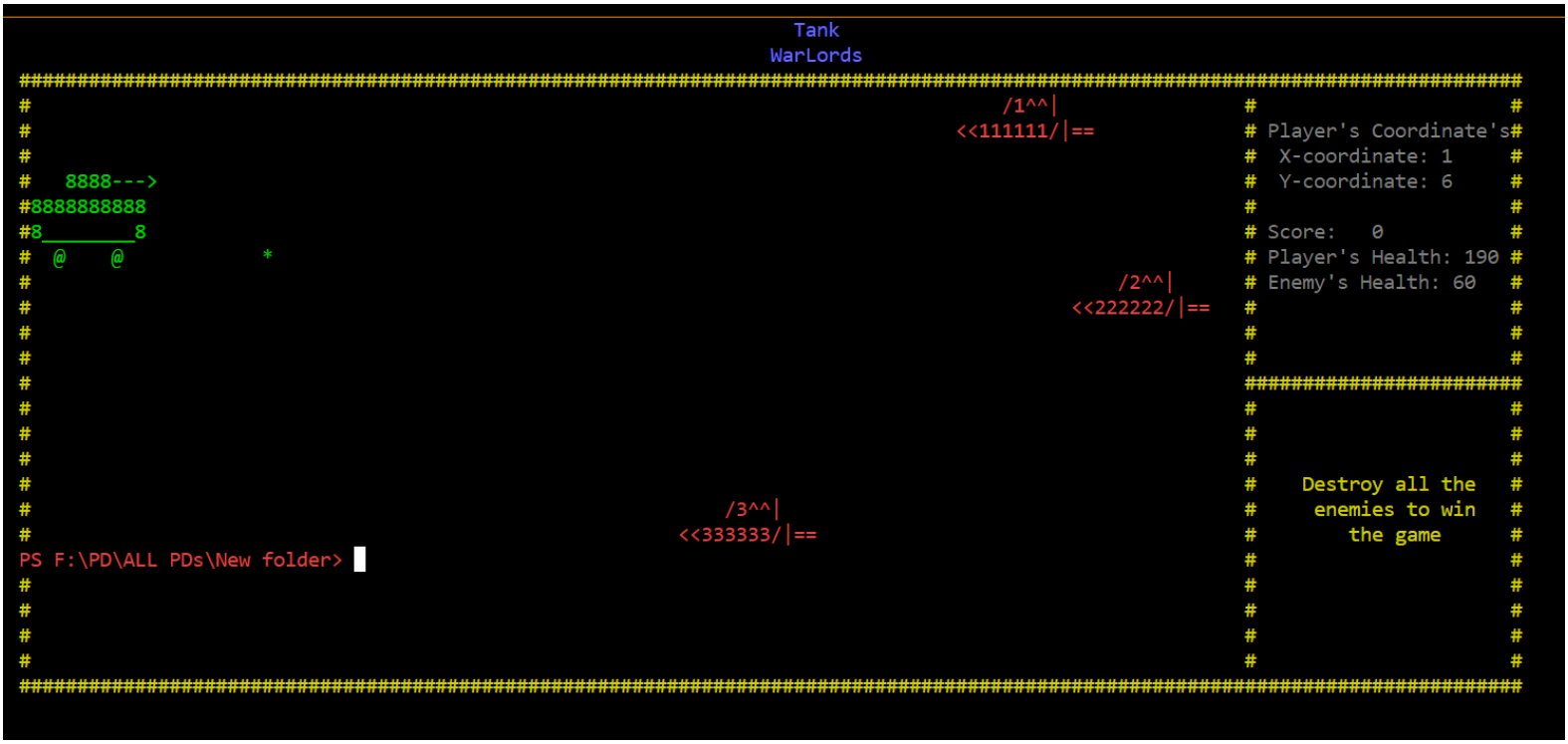
  Header of the game is the name of the game, also there are two more headers. Those two headers are about the result of the game.

- **Rules & Interactions**
    - **Rules:**
        - Press upper key to move player up .
        - Press down key to move player down.
        - Press left key to move player left.
        - Press right key to move player right.
        - Press space bar to fire.
        - If enemy's fire touches you, your health will decrease.
        - If health becomes zero, game will be over.
        - You can win the game if you destroy all the enemies.
- **Goal of the Game**
    - **Achieving a High Score:** This game challenge players to achieve the highest possible score within a given time limit or number of attempts.
    - **Survival:** Survival games often require players to endure in a hostile environment, managing resources and avoiding threats to stay alive for as long as possible.

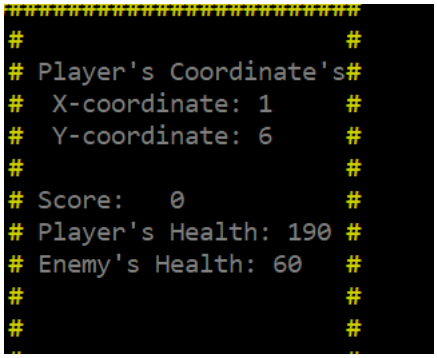
- **Wireframes of the Game (Screenshots of the Menus of the Game)**



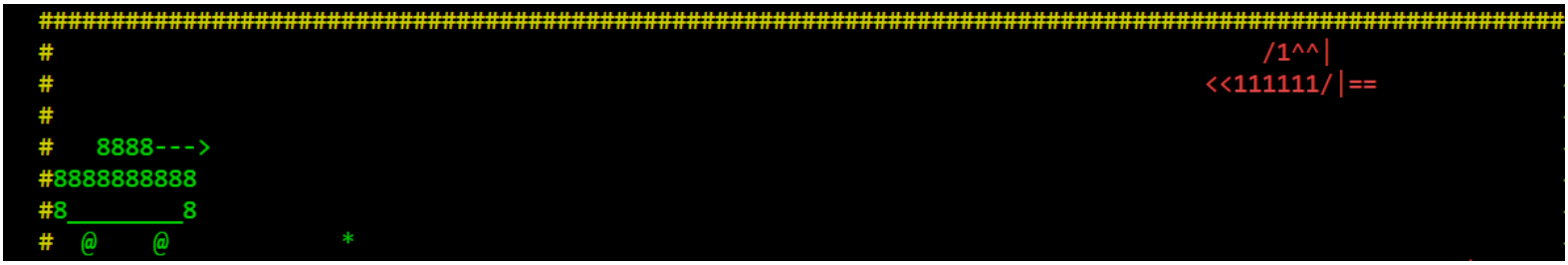
Rules and regulations



Maze or WarField

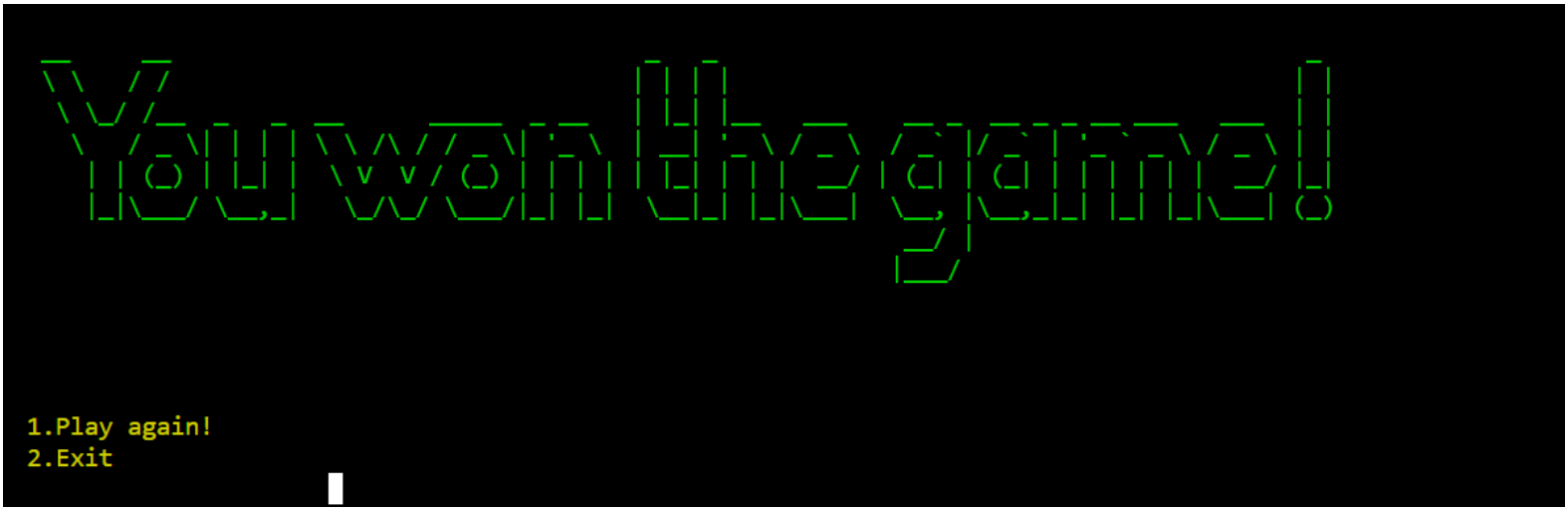


Details shown in the maze



Tank, Bonus Pill and Enemy

Gameover screen



Game winning Screen

- **Data Structures (2D Arrays)**
  - The is not made by 2D arrays.
- **Function Prototypes**
  - void title();
  - void gamewon();
  - string setcolor(unsigned short color);
  - void cursor_show();
  - void printmaze();
  - void gotoxy(int x, int y);
  - void printenemy1(int, int);
  - void eraseenemy1();
  - void gameover();
  - void moveenemy1(string);
  - string changedirection1(string);
  - string changedirection2(string);
  - string changedirection3(string);
  - void printenemy2(int, int);
  - void eraseenemy2();
  - void moveenemy2(string);
  - void printenemy3(int, int);
  - void eraseenemy3();
  - void moveenemy3(string);

- void placeBonusPill();
- char getCharAtxy(short int x, short int y);
- void printplayer();
- void eraseplayer();
- void moveplayerup();
- void moveplayerdown();
- void moveplayerleft();
- void moveplayerright();
- void movefire();
- void producefire();
- void cursor_hide();
- void moveenemyfire();
- void produceenemyfire();
- void printrules();

- **Complete Code**
-

```
#include <iostream>
#include <windows.h>
#include <conio.h>
#include<ctime>
#include<cstdlib>
#include<string>
using namespace std;
void title();
void gamewon();
string setcolor(unsigned short color);
void moveenemyfire();
void cursor_show();
void printmaze();
void gotoxy(int x, int y);
void printenemy1(int, int);
void eraseenemy1();
void gameover();
void moveenemy1(string);
string changedirection1(string);
string changedirection2(string);
string changedirection3(string);
void printenemy2(int, int);
void eraseenemy2();
void moveenemy2(string);
void printenemy3(int, int);
void eraseenemy3();
void moveenemy3(string);
```

```
void placeBonusPill();
char getCharAtxy(short int x, short int y);
void printplayer();
void eraseplayer();
void moveplayerup();
void moveplayerdown();
void moveplayerleft();
void moveplayerright();
void movefire();
void producefire();
void cursor_hide();
//void moveenemyfire();
void produceenemyfire();

void printrules();
int count=0;
int px = 1, py = 6;
int ex1 = 91, ey1 = 3, ex2 = 91, ey2 = 5, ex3 = 67, ey3 = 24, score = 0;
int bonusY ,bonusX;
float EH=60,E1h=20,E2h=20,E3h=20;
int count1=0,count2=0,count3=0;
int PHealth=200;
main()
{
    system("cls");
    title();
    printrules();
    system("cls");
    printmaze();
    printenemy1(91, 3);
    printenemy2(91, 5);
    printenemy3(67, 24);
    string direction1 = "left";
    string direction2 = "down";
    string direction3 = "diagonalleft";
    printplayer();
    placeBonusPill();

    string result;
    cursor_hide();
    cursor_show();
    while (true)
    {

        if (GetAsyncKeyState(VK_LEFT))
        {
            moveplayerleft();
        }
        if (GetAsyncKeyState(VK_RIGHT))
        {
            if(px+11<47)
            {moveplayerright();}
```

```
}
if (GetAsyncKeyState(VK_UP))
{
   moveplayerup();
}
if (GetAsyncKeyState(VK_DOWN))
{
   moveplayerdown();
}
if (GetAsyncKeyState(VK_SPACE))
{
   producefire();
  //produceenemyfire();
}

direction1 = changedirection1(direction1);
direction2 = changedirection2(direction2);
direction3 = changedirection3(direction3);
moveenemy1(direction1);
//eraseenemy1();

moveenemy2(direction2);
moveenemy3(direction3);
// {

//    gotoxy(0,29);
//    cout << "pill resoaned "<<x;
//    score+=5;

//    x=x+1;
// }
moveenemyfire();
produceenemyfire();
movefire();
gotoxy(108,4);
cout<<"Player's Coordinate's";
gotoxy(109,5);
cout<<"X-coordinate: ";
gotoxy(123,5);
cout<<px;
gotoxy(109,6);
cout<<"Y-coordinate: ";
gotoxy(123,6);
cout<<py;
gotoxy(108,8);
cout<<"Score: ";
gotoxy(117,8);
cout<<score;
gotoxy(108,9);
cout<<"Player's Health: ";
gotoxy(125,9);
cout<<"   ";
```

```
        gotoxy(125,9);
        cout<<PHealth;
        gotoxy(108,10);
        cout<<"Enemy's Health: ";
        gotoxy(124,10);
        cout<<EH;


       // testcases();

        if(EH<=0){
            system("cls");
            gamewon();
            string ans;
            setcolor(6);
            cout<<"1.Play again!"<<endl;
            cout<<"2.Exit"<<endl;
            getline(cin>>ws,ans);
            if(ans=="1"){
                system("cls");
                continue;
            }
            else{
                system("cls");
                break;
            }


        }
        else if(PHealth<=0){
            system("cls");
            gameover();
                string ans;
                setcolor(6);
            cout<<"1.Play again!"<<endl;
            cout<<"2.Exit"<<endl;
            getline(cin>>ws,ans);
            if(ans=="1"){
                system("cls");
                continue;
            }
            else{
                system("cls");
                break;
            }

        }
        Sleep(100);
    }
    gotoxy(2,2);
    getch();
}
```

```
void printmaze()
{   setcolor(1);
    cout<<"                                        Tank                                        "<<endl;
    cout<<"                                       WarLords                                      "<<endl;
    setcolor(6);
    cout
<<"##########################################################################################################
#########################################" << endl;
    cout <<"#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                ##########################" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                    Destroy all the   #" << endl;
   cout << "#                                                                     enemies to win   #" << endl;
   cout << "#                                                                       the game      #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
   cout << "#                                                                                #" << endl;
    cout
<<"##########################################################################################################
#########################################" << endl;

}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
void printenemy1(int ex1, int ey1)
{
    setcolor(4);
    gotoxy(ex1, ey1);
    cout << "    /1^^|    " << endl;
    gotoxy(ex1, ey1 + 1);
    cout << "<<111111/|==" << endl;
```

```
   /* gotoxy(ex1, ey1 + 2);
    cout << "/_____________|===  " << endl;*/
}
void eraseenemy1()
{
    gotoxy(ex1, ey1);
    cout << "             " << endl;
    gotoxy(ex1, ey1 + 1);
    cout << "             " << endl;
  /*  gotoxy(ex1, ey1 + 2);
    cout << "             " << endl;*/
}
void moveenemy1(string direction1)
{
    if(count1<5){
    if(getCharAtxy(ex1-1,ey1)=='o' || getCharAtxy(ex1-2,ey1)=='o' || getCharAtxy(ex1-1,ey1+1)=='o' || getCharAtxy(ex1-
2,ey1+1)=='o'){
        count1++;
    }}
    eraseenemy1();
    if(count1<5){

    if (direction1 == "left")
    {
        ex1 = ex1 - 2;
    }
    if (direction1 == "right")
    {
        ex1 = ex1 + 2;
    }
    printenemy1(ex1, ey1);}

}

string changedirection1(string direction1)
{
    setcolor(4);
    if (direction1 == "left" && ex1 <= 50)
    {
        direction1 = "right";
    }
    if (direction1 == "right" && ex1 >= 91)
    {
        direction1 = "left";
    }
    return direction1;
}
string changedirection2(string direction2)
{


    if (direction2 == "up" && ey2 == 5)
```

```
    {
        direction2 = "down";
    }
    if (direction2 == "down" && ey2 == 24)
    {
        direction2 = "up";
    }
    return direction2;
}
void printenemy2(int ex2, int ey2)
{
    setcolor(4);
   gotoxy(ex2, ey2);
    cout << "    /2^^|    " << endl;
    gotoxy(ex2, ey2 + 1);
    cout << "<<222222/|==" << endl;
   /* gotoxy(ex2, ey2 + 2);
    cout << "/_____________|===  " << endl;*/
}
void eraseenemy2()
{
    gotoxy(ex2, ey2);
    cout << "            " << endl;
    gotoxy(ex2, ey2 + 1);
    cout << "            " << endl;
   /* gotoxy(ex2, ey2 + 2);
    cout << "            " << endl;*/
}
void moveenemy2(string direction2)
{
    if(count2<5){
    if(getCharAtxy(ex1-1,ey1)=='o' || getCharAtxy(ex1-2,ey1)=='o' || getCharAtxy(ex1-1,ey1+1)=='o' || getCharAtxy(ex1-
2,ey1+1)=='o'){
        count2++;
    }}

    eraseenemy2();
    if(count2<5){
    if (direction2 == "up")
    {
        ey2 = ey2 - 1;
    }
    if (direction2 == "down")
    {
        ey2 = ey2 + 1;
    }

    printenemy2(ex2, ey2);}
}

void printenemy3(int ex3, int ey3)
{
```

```
    setcolor(4);
    gotoxy(ex3, ey3);
    cout << "    /3^^|    " << endl;
    gotoxy(ex3, ey3 + 1);
    cout << "<<333333/|==" << endl;
   /* gotoxy(ex3, ey3 + 2);
    cout << "/_____________|===  " << endl;*/
}
void eraseenemy3()
{
    gotoxy(ex3, ey3);
    cout << "             " << endl;
    gotoxy(ex3, ey3 + 1);
    cout << "             " << endl;
   /* gotoxy(ex3, ey3 + 2);
    cout << "                    " << endl;*/
}
void moveenemy3(string direction3)
{
    if(count3<5){
    if(getCharAtxy(ex1-1,ey1)=='o' || getCharAtxy(ex1-2,ey1)=='o' || getCharAtxy(ex1-1,ey1+1)=='o' || getCharAtxy(ex1-
2,ey1+1)=='o'){
        count3++;
    }}
    eraseenemy3();
    if(count3<5){
    if (direction3 == "diagonalleft")
    {
        ex3 = ex3 - 2;
        ey3 = ey3 - 1;
    }
    if (direction3 == "diagonalright")
    {
        ex3 = ex3 + 2;
        ey3 = ey3 + 1;
    }

    printenemy3(ex3, ey3);}
}

string changedirection3(string direction3)
{

    if (direction3 == "diagonalright" && ex3 == 67 && ey3 == 24)
    {
        direction3 = "diagonalleft";
    }
    if (direction3 == "diagonalleft" && ex3 == 47 && ey3 == 14)
    {
        direction3 = "diagonalright";
    }
    return direction3;
```

```
}

void printplayer()
{
    setcolor(2);
    gotoxy(px, py);
    cout << "   8888--->" << endl;
    gotoxy(px, py + 1);
    cout << "8888888888" << endl;
    gotoxy(px, py + 2);
    cout << "8_________8" << endl;
    gotoxy(px, py + 3);
    cout << "  @     @  " << endl;

}
void eraseplayer()
{
    gotoxy(px, py);
    cout << "           " << endl;
    gotoxy(px, py + 1);
    cout << "           " << endl;
    gotoxy(px, py + 2);
    cout << "           " << endl;
    gotoxy(px, py + 3);
    cout << "           " << endl;
}
void moveplayerleft()
{   char left1=getCharAtxy(px - 1, py) ;
    char left2=getCharAtxy(px - 1, py+1) ;
    char left3=getCharAtxy(px - 1, py+2) ;
    char left4=getCharAtxy(px - 1, py+3) ;

  if (left1 == ' ' && left2 == ' ' && left3 == ' ' && left4 == ' ' )
                    {

                        eraseplayer();
                        px = px - 1;
                        printplayer();
                    }
    if (left1 == '*' || left2 == '*' || left3 == '*' || left4 == '*' ){
         eraseplayer();
        px = px - 1;
        printplayer();
        PHealth +=5;
        placeBonusPill();
    }
                }
void moveplayerright()
{
    char next1 = getCharAtxy(px + 11, py);
    char next2 = getCharAtxy(px + 11, py + 1);
    char next3 = getCharAtxy(px + 11, py + 2);
```

```
    char next4 = getCharAtxy(px + 11, py + 3);
    //char next5 = getCharAtxy(px + 9, py + 4);
   // char next6 = getCharAtxy(px + 9, py + 5);
    if (next1 == ' ' && next2 == ' ' && next3 == ' ' && next4 == ' '/*&& next5 == ' ' && next6 == ' '*/ )
    {
        eraseplayer();
        px = px + 1;
        printplayer();
          }
    if (next1 == '*' || next2 == '*' || next3 == '*' || next4 == '*' /*|| next5 == '*' || next6 == '*'*/)
    {
        eraseplayer();
        px = px + 1;
        printplayer();
       PHealth += 5;
        placeBonusPill();
    }
}

void moveplayerup()
{   char up1=getCharAtxy(px, py - 1);
char up2=getCharAtxy(px+1, py - 1);
char up3=getCharAtxy(px+2, py - 1);
char up4=getCharAtxy(px+3, py - 1);
char up5=getCharAtxy(px+4, py - 1);
char up6=getCharAtxy(px+5, py - 1);
char up7=getCharAtxy(px+6, py - 1);
char up8=getCharAtxy(px+7, py - 1);
char up9=getCharAtxy(px+8, py - 1);
char up10=getCharAtxy(px+9, py - 1);
char up11=getCharAtxy(px+10, py - 1);
char up12=getCharAtxy(px+11, py - 1);
/*char up13=getCharAtxy(px+12, py - 1);
char up14=getCharAtxy(px+13, py - 1);
char up15=getCharAtxy(px+14, py - 1);
char up16=getCharAtxy(px+15, py - 1);*/
  if(up1==' ' && up2==' '&& up3==' ' && up4==' ' && up5==' ' && up5==' ' && up6==' ' && up7==' ' && up8==' ' && up9==' '
/*&& up10==' ' &&  up11==' ' && up12==' ' &&  up13==' ' && up14==' ' && up15 ==' ' && up16==' '*/ ){
    eraseplayer();
    py = py - 1;
    printplayer();
 }
if(up1=='*'|| up2=='*'|| up3=='*'|| up4=='*'|| up5=='*'|| up5=='*'|| up6=='*'|| up7=='*'|| up8=='*'|| up9=='*'/*|| up10=='*' ||
up11=='*'|| up12=='*'|| up13=='*'|| up14=='*'|| up15 =='*' || up16=='*'*/ ){
    eraseplayer();
    py = py - 1;
    printplayer();
    PHealth +=5;
    placeBonusPill();
}

}
```

```
void moveplayerdown()
{
char up1=getCharAtxy(px, py + 4);
char up2=getCharAtxy(px+1, py + 4);
char up3=getCharAtxy(px+2, py + 4);
char up4=getCharAtxy(px+3, py + 4);
char up5=getCharAtxy(px+4, py + 4);
char up6=getCharAtxy(px+5, py + 4);
char up7=getCharAtxy(px+6, py + 4);
char up8=getCharAtxy(px+7, py + 4);
char up9=getCharAtxy(px+8, py + 4);
char up10=getCharAtxy(px+9, py + 4);
char up11=getCharAtxy(px+10, py + 6);
char up12=getCharAtxy(px+11, py + 6);
/*char up13=getCharAtxy(px+12, py + 6);
char up14=getCharAtxy(px+13, py + 6);
char up15=getCharAtxy(px+14, py + 6);
char up16=getCharAtxy(px+15, py + 6);*/
  if(up1==' ' && up2==' '&& up3==' ' && up4==' ' && up5==' ' && up5==' '  && up6==' ' && up7==' ' && up8==' ' && up9==' 
'/*&& up10==' ' && up11==' ' && up12==' ' &&  up13==' ' && up14==' ' && up15 ==' ' && up16==' '*/ ){
    eraseplayer();
    py = py +1;
    printplayer();
 }
if(up1=='*'|| up2=='*'|| up3=='*'|| up4=='*'|| up5=='*'|| up5=='*'|| up6=='*'|| up7=='*'|| up8=='*'|| up9=='*'/*|| up10=='*'|| 
up11=='*'|| up12=='*'|| up13=='*'|| up14=='*'|| up15 =='*' || up16=='*'*/){
    eraseplayer();
    py = py + 1;
    printplayer();
    PHealth +=5;
    placeBonusPill();
}

}

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 bonus()
{
    gotoxy(30, 6);
    cout << "*";
}
```

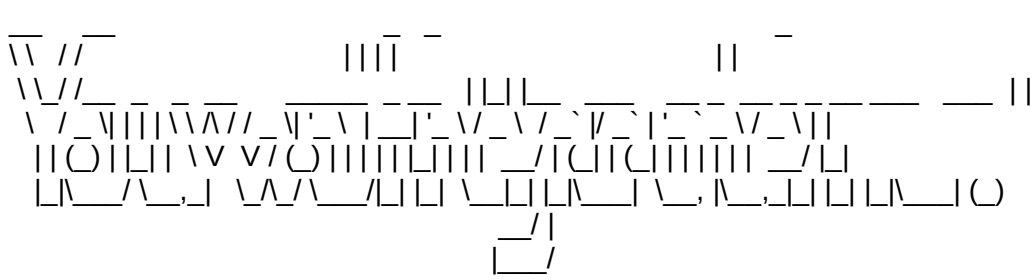
```
void placeBonusPill()
{
    srand(time(0));
    bonusX = rand() % 43; // Adjust based on the width of your console window
    srand(time(0));
    bonusY = rand() % 24; // Adjust based on the height of your console window
    while(!(getCharAtxy(bonusX, bonusY) ==' ' && bonusX>1 && bonusY>3))
{
    // Generate random position for the bonus pill
    srand(time(0));
    bonusX = rand() % 43; // Adjust based on the width of your console window
    srand(time(0));
    bonusY = rand() % 24; // Adjust based on the height of your console window
}
    // Set the cursor position to represent the bonus pill
    gotoxy(bonusX,bonusY);
    cout << '*';

}

void producefire(){
    gotoxy(px+11,py);
    cout<<"o";
}
void movefire(){
    int k=0;
    for (int i = 1; i < 110; i++)
    {
        for (int j = 0; j < 24; j++)
        {
            if(k>0){
                k=0;
                continue;
            }

         if (getCharAtxy(i,j)=='o')

         {
            if (getCharAtxy(i+1,j)==' ')
            {
            gotoxy(i,j);
            cout<<" ";
            gotoxy(i+1,j);
            cout<<"o";
            k=i+1;
            }
            else if(getCharAtxy(i+1,j)=='/' || getCharAtxy(i+1,j)=='<' || getCharAtxy(i+1,j)=='#' || getCharAtxy(i+1,j)=='*' ||
getCharAtxy(i+1,j)=='+'){
                if(getCharAtxy(i+1,j)=='/' || getCharAtxy(i+1,j)=='<'){
                  if(getCharAtxy(i+3,j)=='1' || getCharAtxy(i+2,j)=='1'){

                   count1++;
```

```
            score+=5;
            EH-=4;
            gotoxy(i,j);
          }
          else if(getCharAtxy(i+3,j)=='2' || getCharAtxy(i+2,j)=='2'){
            count2++;
            score+=5;
            EH-=4;
            gotoxy(i,j);
          }
          else if(getCharAtxy(i+3,j)=='3' || getCharAtxy(i+2,j)=='3'){
            count3++;
            score+=5;
            EH-=4;
            gotoxy(i,j);}
            cout<<" ";
        }
        else if(getCharAtxy(i+1,j)=='#'|| getCharAtxy(i+1,j)=='*' || getCharAtxy(i+1,j)=='+'){
        gotoxy(i,j);
        cout<<" ";
        }
      }

   }

} }}
void enemyfire(){

}
void printrules(){
   string select;
   setcolor(4);
   cout<<"Rules: "<<endl;
   cout<<"Press upper key to move player up"<<endl;
   cout<<"Press down key to move player down"<<endl;
   cout<<"Press left key to move player left"<<endl;
   cout<<"Press right key to move player right"<<endl;
   cout<<"Press space bar to fire"<<endl;
   cout<<"If enemy's fire touches you , your health will decrease."<<endl;
   cout<<"If health becomes zero, game will be over."<<endl;
   cout<<"There are two levels in the game."<<endl;
   cout<<endl;
   cout<<"Press any key to continue...";
   getch();

}
void produceenemyfire(){

   setcolor(8);
   if((py==ey1 || py+1==ey1 || py==ey1+1 || py+1==ey1+1) && count1<5){
      gotoxy(ex1-3,ey1+1);
      cout<<"+";
```

```
    }
      if((py+1==ey2 || py+2==ey2 || py+1==ey2+1 || py+2==ey2+1) && count2<5){
        gotoxy(ex2-3,ey2+1);
        cout<<"+";
       // getch();
    }
    if((py+1==ey3 || py+2==ey3 || py+1==ey3+1 || py+2==ey3+1 ) && count3<5){
        gotoxy(ex3-3,ey3+1);
        cout<<"+";
       // getch();
    }
}
void moveenemyfire(){
    for (int i = 110; i >= 0; i--)
    {
        for (int j = 0; j <26; j++)
        {


         if (getCharAtxy(i,j)=='+')

         {
            if (getCharAtxy(i-1,j)==' ')
            {
            gotoxy(i,j);
            cout<<" ";
            gotoxy(i-1,j);
            cout<<"+";
            }
            else if(getCharAtxy(i-1,j)=='8' || getCharAtxy(i-1,j)=='>' || getCharAtxy(i-1,j)=='@' || getCharAtxy(i-1,j)=='#' ||
getCharAtxy(i-1,j)=='|' || getCharAtxy(i-1,j)=='=' || getCharAtxy(i-1,j)=='*' || getCharAtxy(i-1,j)=='o'){
                if(getCharAtxy(i-1,j)=='8' || getCharAtxy(i-1,j)=='>' || getCharAtxy(i-1,j)=='@'){

                  PHealth -= 5;
                  gotoxy(i,j);
                  cout<<" ";
                  }

                else if(getCharAtxy(i-1,j)=='#' || getCharAtxy(i-1,j)=='|' || getCharAtxy(i-1,j)=='=' || getCharAtxy(i-1,j)=='*'  ||
getCharAtxy(i-1,j)=='o'){
                gotoxy(i,j);
                cout<<" ";
                }}


    }
        }
        }}
void title()
{
    gotoxy(14, 1);
    setcolor(3);
```

```
    cout << R"(


 ________        _   __        __        _                 _
|__    __|       | |   \ \       / /       | |               | |
   | | __ _ _ __ | | __  \ \  /\  / /_ _ _ __| | ___  _ __ __| |___
   | |/ _` | '_ \| |/ /   \ \/  \/ / _` | '__| |/ _ \| '__/ _` / __|
   | | (_| | | | |   <     \  /\  / (_| | |  | |___| (_) | | | (_| \__ \
   |_|\__,_|_| |_|_|\_\     \/  \/ \__,_|_|  |_______\___/|_|  \__,_|___/


    )"
        << endl;
}
void cursor_hide()
{
    /*
       For Removing Blinking Cursor on Screen
    */
    HANDLE hStdOut = NULL;
    CONSOLE_CURSOR_INFO curInfo;

    hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
    GetConsoleCursorInfo(hStdOut, &curInfo);
    curInfo.bVisible = FALSE;
    SetConsoleCursorInfo(hStdOut, &curInfo);
}
void cursor_show()
{
    /*
       For Removing Blinking Cursor on Screen
    */
    HANDLE hStdOut = NULL;
    CONSOLE_CURSOR_INFO curInfo;

    hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
    GetConsoleCursorInfo(hStdOut, &curInfo);
    curInfo.bVisible = TRUE;
    SetConsoleCursorInfo(hStdOut, &curInfo);
}

void gameover()
{
    setcolor(4);
    gotoxy(14, 1);
    cout << R"(


  _____
 / ____|
| |  __  __ _ _ __ ___   ___    _____   _____ _ __
| | |_ |/ _` | '_ ` _ \ / _ \  / _ \ \ / / _ \ '__|
| |__| | (_| | | | | | |  __/ | (_) \ V /  __/ |
 \_____|\__,_|_| |_| |_|\___|  \___/ \_/ \___|_|
```

```
    )"
        << endl;
}
void gamewon()
{
    setcolor(2);
    gotoxy(14, 1);
    cout << R"(
```



```
    )"
        << endl;
}
string setcolor(unsigned short color){
    HANDLE hcon=GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hcon,color);
    return "";
```

Student Reg. No:2023-CS-69 **Student Name:**

Muhammad Ahmad

	A-Extensive Evidence	**B-Convincing Evidence**	**C-Limited Evidence**	**D-No Evidence**
Documentation Formatting **Grade:**	All the documentation meets all the criteria.	Documentation is well formatted but some of the criteria is not fulfilled.	Documentation is required a lot of improvement.	Documentation is not Available
Documentation Formatting Criteria: In **Binder**, **Title** Page, **Header**-Footers, Font **Style**, Font **Size** all are all consistence and according to given **guidelines**. Project **Poster** is professionally design and well presented				
Documentation Contents **Grade:**	Documentation includes all of the criteria.	Documentation meet more than 80% of the criteria given.	Documentation meet more than 50% of the criteria.	When the documentation meet less than 50% of the criteria.
Documentation Contents Criteria: Title Page - **Table** of Contents - Project **Short Description and Story Writing of Game** - **Game Characters** Description - **Rules** & Interactions - **Goal** of the Game **- Screenshot** of the Game - **Data Structures** Used in the Game - **Functions** Prototype - **Full Code**				
Project Complexity **Grade:**	Project has at least 1 Player and 3 enemies. Proper use of gotoxy() function. Health system, Firing System and lives decreasing system.	Project complexity meet 80% criteria given in extensive evidence	Project complexity meet 50% criteria given in extensive evidence	Project complexity meet less than 50% criteria given in extensive evidence
Randomness **Grade:**	Objects are produced randomly in the game.	meet more than 80% of the criteria given.	meet more than 50% of the criteria given.	Objects are appearing in the same pattern
Code Style **Grade:**	All Code style criteria is followed	All code style criteria followed but some improvements required	lot of improvements required in coding style.	**Did not follow** code style,
Code Style Criteria: Consistent code style. Code is well indented. Variable and Function names are well defined. White Spaces are well used. Comments are added.				
Code Documentation Mapping **Grade:**	Code and documentation is synchronized.	Code and documentation does not synchronized at **some** places	Code and documentation does not synchronized at **many** places	Code and documentation **does not** synchronized.
Idea Novelty and Creativity **Grade:**	Idea is unique of the game	Idea is merged by combining other different games	Same idea as a previous game	Could not implement the existing game idea.
Data Structure (2D Arrays) **Optional**	Data structure is sufficient for the project requirements	Data Structure is sufficient but require improvement to meet project requirements.	Data structure is not sufficient and need a lot of improvement	Data Structure is not properly identified and declared.
File Handling **Optional**	Game maze is loaded and the updated maze is stored in the file	Game maze is loaded and partial data is stored in the file.	Game maze is just loaded but the updated game configuration is not stored in the maze.	Project do not contain file handling
Modularity **Grade:**	Meet all Modularity criteria	Meet all Modularity criteria but at some places it is missing	Do not sufficiently meet the modularity criteria.	No modularity or very minimum modularity.
Modularity criteria: Functions are defined for each major feature. Functions are independent (identify from parameter list and return types)- There is no global variable defined. Arrays and variables are passed as parameters to the functions. Functions exhibit single responsibility principle.				
Screen flickering **Grade:**	There is no Screen flickering.	Maze is not flickering but the characters are flickering at great speed	Flickering is done at lot of places	Screen is flickering at all places
Presentation and Demo **Grade:**	Presentation and Demo was 100% working	Presentation and Demo require some improvements	Presentation and Demo require a lot of improvements	Presentation was not ok and Demo was not working
Student Understanding with the Code. **Grade:**	Student has complete understanding how the code is working and knows the concept.	Student has good understand but some place he does not know the concepts	Student has a very little understand and lack the major concepts.	Student does not have any level of understanding of the code.

Checked by: Click or tap here to enter text.

Student Reg. No:2023-CS-69 **Student Name:** Muhammad Ahmad

	A-Extensive Evidence	**B-Convincing Evidence**	**C-Limited Evidence**	**D-No Evidence**
Documentation Formatting **Grade:**	All the documentation meets all the criteria.	Documentation is well formatted but some of the criteria is not fulfilled.	Documentation is required a lot of improvement.	Documentation is not Available
Documentation Formatting Criteria: In **Binder**, **Title** Page, **Header**-Footers, Font **Style**, Font **Size** all are all consistence and according to given **guidelines**. Project **Poster** is professionally design and well presented				
Documentation Contents **Grade:**	Documentation includes all of the criteria.	Documentation meet more than 80% of the criteria given.	Documentation meet more than 50% of the criteria.	When the documentation meet less than 50% of the criteria.
Documentation Contents Criteria: Title Page - **Table** of Contents - Project **Short Description and Story Writing of Game** - **Game Characters** Description - **Rules** & Interactions - **Goal** of the Game **- Screenshot** of the Game - **Data Structures** Used in the Game - **Functions** Prototype - **Full Code**				
Project Complexity **Grade:**	Project has at least 1 Player and 3 enemies. Proper use of gotoxy() function. Health system, Firing System and lives decreasing system.	Project complexity meet 80% criteria given in extensive evidence	Project complexity meet 50% criteria given in extensive evidence	Project complexity meet less than 50% criteria given in extensive evidence
Randomness **Grade:**	Objects are produced randomly in the game.	meet more than 80% of the criteria given.	meet more than 50% of the criteria given.	Objects are appearing in the same pattern
Code Style **Grade:**	All Code style criteria is followed	All code style criteria followed but some improvements required	lot of improvements required in coding style.	**Did not follow** code style,
Code Style Criteria: Consistent code style. Code is well indented. Variable and Function names are well defined. White Spaces are well used. Comments are added.				
Code Documentation Mapping **Grade:**	Code and documentation is synchronized.	Code and documentation does not synchronized at **some** places	Code and documentation does not synchronized at **many** places	Code and documentation **does not** synchronized.
Idea Novelty and Creativity **Grade:**	Idea is unique of the game	Idea is merged by combining other different games	Same idea as a previous game	Could not implement the existing game idea.
Data Structure (2D Arrays) **Optional**	Data structure is sufficient for the project requirements	Data Structure is sufficient but require improvement to meet project requirements.	Data structure is not sufficient and need a lot of improvement	Data Structure is not properly identified and declared.
File Handling **Optional**	Game maze is loaded and the updated maze is stored in the file	Game maze is loaded and partial data is stored in the file.	Game maze is just loaded but the updated game configuration is not stored in the maze.	Project do not contain file handling
Modularity **Grade:**	Meet all Modularity criteria	Meet all Modularity criteria but at some places it is missing	Do not sufficiently meet the modularity criteria.	No modularity or very minimum modularity.
Modularity criteria: Functions are defined for each major feature. Functions are independent (identify from parameter list and return types)- There is no global variable defined. Arrays and variables are passed as parameters to the functions. Functions exhibit single responsibility principle.				
Screen flickering **Grade:**	There is no Screen flickering.	Maze is not flickering but the characters are flickering at great speed	Flickering is done at lot of places	Screen is flickering at all places
Presentation and Demo **Grade:**	Presentation and Demo was 100% working	Presentation and Demo require some improvements	Presentation and Demo require a lot of improvements	Presentation was not ok and Demo was not working
Student Understanding with the Code. **Grade:**	Student has complete understanding how the code is working and knows the concept.	Student has good understand but some place he does not know the concepts	Student has a very little understand and lack the major concepts.	Student does not have any level of understanding of the code.

Checked by: Click or tap here to enter text.