Tanker Titans

Session: 2023 – 2027

Submitted by:

Umme Aymen 2023-CS-112

Supervised by:

Dr. Awais Hassan

&

Mam Maida Mirza

Department of Computer Science

University of Engineering and Technology

Lahore Pakistan

Contents

Short Description

Once there was a car, which set on a journey. When half of the distance was covered to his destination, the car was trapped by the army tanks. Afraid of being squashed by them, the car didn't know that its owner has installed bullets in it. Now there is only one way to survive. DESTROY THEM ALL!!!!!!!

Game Characters Description

Player

There is a one-player name CAR.

CAR:

CAR is the main character of the story, a car which is durable and armed. Throughout the game, it uses its skills to overcome obstacles and escape from dangers. Despite the challenges he faces, and never gives up and uses his courage and intelligence to find his way back home.

Enemies

There are 4 enemies in the game of similar type.

TANKS:

Tanks are relentless enemy in the game, who can only move horizontally on the game. But they can fire projectiles at the player. Its movement is paired with its deadly accuracy. The player will need to be quick to avoid Tank shots.

Game Objects Description

Following are the Objects in the Game

Player bullets:

There will be player bullets represented by '@'. When a player's bullet will hit the enemy, the score will increase and the enemy's health will decrease according to the applied conditions.

Enemy bullets:

There will be enemy bullets represented by 'o'. When the enemy bullets interact with the player its health will be reduced.

Goal of the Game

The goal of the game is to overcome and beat each enemy one by one and make all the enemies dead and make a way back to destination.

Wireframes

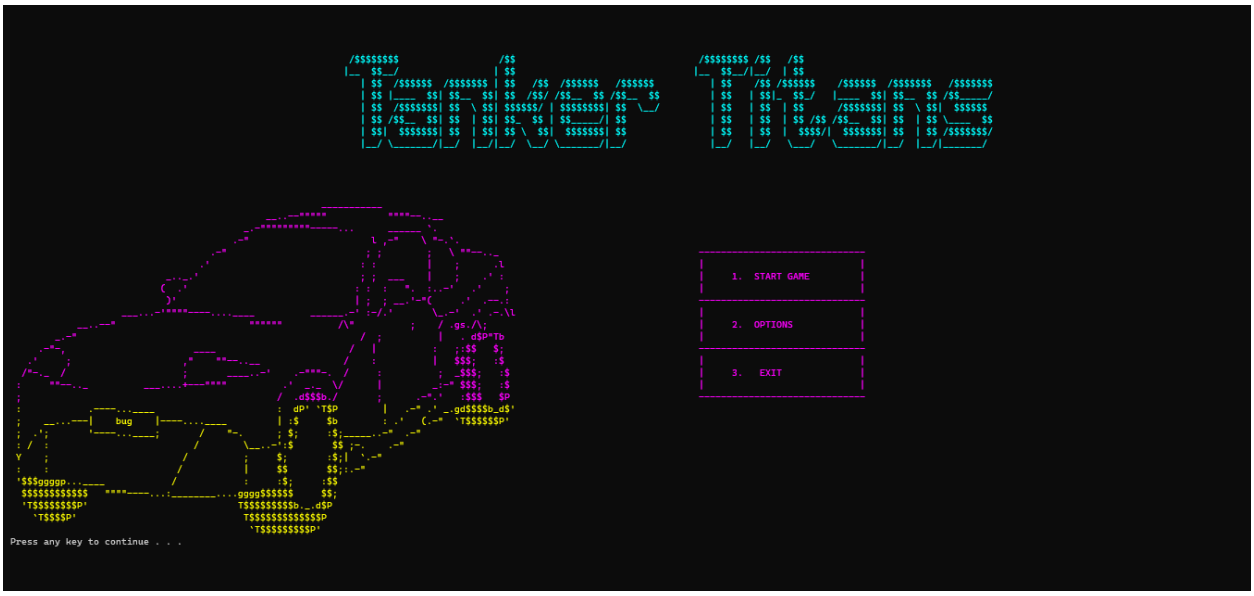

Figure 1: Game menu

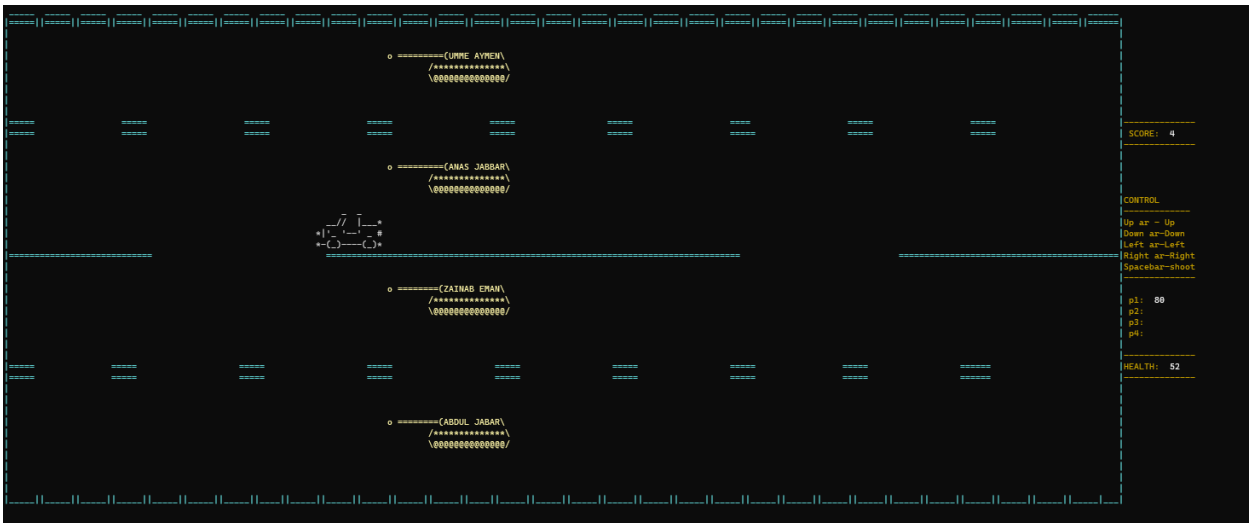

Figure 2: Main game

Figure 3: Game Over menu

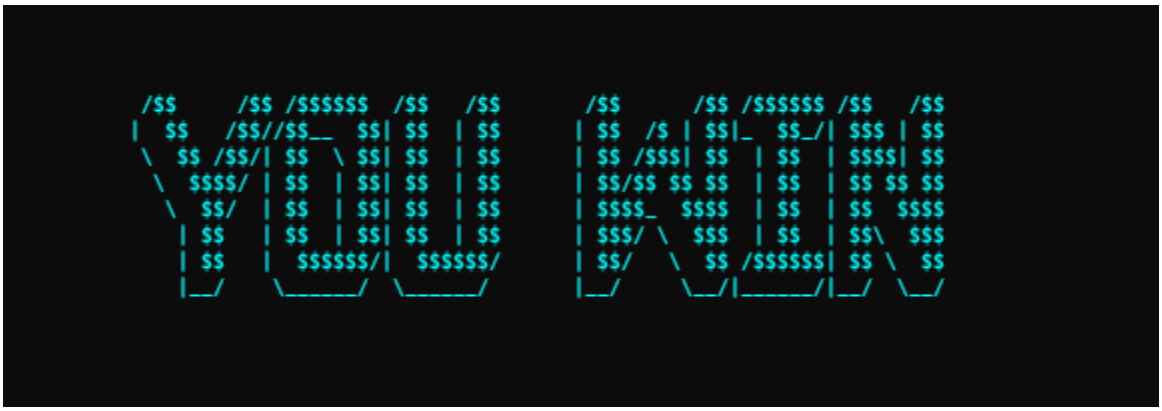

Figure 5: Win Screen

Data Structures & Function Prototypes

```
#include <iostream>
#include <windows.h>
#include <conio.h>
using namespace std;
char getCharAtxy(short int x, short int y);
void gotoxy(int x, int y);
void hideCursor();

void setTextColor(int colorCode);
void setBackgroundColor(int colorCode);
void resetColors();
```

```
void printheader();

int xP = 202, yP = 2, x1 = 3, y1 = 4, x2 = 3, y2 = 14, x3 = 3, y3 = 25, x4 = 3,
y4 = 37;

void printmaze();
void printcar();
void PrintWin();
void printgamelose();

void printPlayer();
void printEnemy1();
void printEnemy2();
void printEnemy3();
void printEnemy4();
void eraseEnemy1();
void eraseEnemy2();
void eraseEnemy3();
void eraseEnemy4();
void moveplayerUp();
void moveplayerDown();
void moveplayerLeft();
void moveplayerRight();
void moveEnemy1();
void moveEnemy2();
void moveEnemy3();
void moveEnemy4();

void generatebullet();
void movebullet();
void printbullet(int x, int y);
void erasebullet(int x, int y);
void makebulletInactive(int index);
void deletebullet(int index);
void bulletcollisionwidenemy();

void updatescore();
void updateHealth();
void addscore();
void update_power1();
void update_power2();
void update_power3();
void update_power4();
void DecreasePlayerHealth();
void d_p1();
```

```
void d_p2();
void d_p3();
void d_p4();

int score = 0;
int health = 100;
int enemy1_p = 100;
int enemy2_p = 100;
int enemy3_p = 100;
int enemy4_p = 100;
void print_info();

void setColor(int colorCode);

int bulletx[100];
int bullety[100];
int bulletcount = 0;
char bulletchar = '@';
bool isBulletActive[100];

int B1x = 0, B1y = 0;
bool Enemy1Fire = false;
void Enemy1Generate_Fire();
void Enemy1move_fire();

int B2x = 0, B2y = 0;
bool Enemy2Fire = false;
void Enemy2Generate_Fire();
void Enemy2move_fire();

int B3x = 0, B3y = 0;
bool Enemy3Fire = false;
void Enemy3Generate_Fire();
void Enemy3move_fire();

int B4x = 0, B4y = 0;
bool Enemy4Fire = false;
void Enemy4Generate_Fire();
void Enemy4move_fire();
```

Rubrics:

Student Reg. No.: 2023-CS-112 **Student Name.** Umme Aymen

	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) **Grade:**	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 **Grade:**	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.

Code:

```
      #include <iostream>

#include <windows.h>
#include <conio.h>
using namespace std;
char getCharAtxy(short int x, short int y);
void gotoxy(int x, int y);
void hideCursor();

void setTextColor(int colorCode);
void setBackgroundColor(int colorCode);
void resetColors();
void printheader();

int xP = 202, yP = 2, x1 = 3, y1 = 4, x2 = 3, y2 = 14, x3 = 3, y3 = 25, x4 = 3,
y4 = 37;

void printmaze();
void printcar();
void PrintWin();
void printgamelose();

void printPlayer();
void printEnemy1();
void printEnemy2();
void printEnemy3();
void printEnemy4();
void eraseEnemy1();
void eraseEnemy2();
void eraseEnemy3();
void eraseEnemy4();
void moveplayerUp();
void moveplayerDown();
void moveplayerLeft();
void moveplayerRight();
void moveEnemy1();
void moveEnemy2();
void moveEnemy3();
void moveEnemy4();

void generatebullet();
void movebullet();
void printbullet(int x, int y);
```

```
void erasebullet(int x, int y);
void makebulletInactive(int index);
void deletebullet(int index);
void bulletcollisionwidenemy();

void updatescore();
void updateHealth();
void addscore();
void update_power1();
void update_power2();
void update_power3();
void update_power4();
void DecreasePlayerHealth();
void d_p1();
void d_p2();
void d_p3();
void d_p4();

int score = 0;
int health = 100;
int enemy1_p = 100;
int enemy2_p = 100;
int enemy3_p = 100;
int enemy4_p = 100;
void print_info();

void setColor(int colorCode);

int bulletx[100];
int bullety[100];
int bulletcount = 0;
char bulletchar = '@';
bool isBulletActive[100];

int B1x = 0, B1y = 0;
bool Enemy1Fire = false;
void Enemy1Generate_Fire();
void Enemy1move_fire();

int B2x = 0, B2y = 0;
bool Enemy2Fire = false;
void Enemy2Generate_Fire();
void Enemy2move_fire();

int B3x = 0, B3y = 0;
```

```
bool Enemy3Fire = false;
void Enemy3Generate_Fire();
void Enemy3move_fire();

int B4x = 0, B4y = 0;
bool Enemy4Fire = false;
void Enemy4Generate_Fire();
void Enemy4move_fire();

main()
{
  printheader();
  printcar();
  printmaze();
  hideCursor();
  print_info();
  printPlayer();
  printEnemy1();
  printEnemy2();
  printEnemy3();
  printEnemy4();
  while (true)
  {

    if (GetAsyncKeyState(VK_LEFT))
    {
      moveplayerLeft();
    }
    if (GetAsyncKeyState(VK_RIGHT))
    {
      moveplayerRight();
    }
    if (GetAsyncKeyState(VK_UP))
    {
      moveplayerUp();
    }
    if (GetAsyncKeyState(VK_DOWN))
    {
      moveplayerDown();
    }
    if (GetAsyncKeyState(VK_SPACE))
    {
      generatebullet();
    }
    if (GetAsyncKeyState(VK_ESCAPE))
```

```
    {
      break;
    }
    if (enemy1_p != 0)
    {
      moveEnemy1();
    }
    if (enemy2_p)
    {
      moveEnemy2();
    }
    if (enemy3_p)
    {
      moveEnemy3();
    }
    if (enemy4_p)
    {
      moveEnemy4();
    }
    movebullet();
    bulletcollisionwidenemy();
    if (!Enemy1Fire)
    {
      if (enemy1_p > 0)
        Enemy1Generate_Fire();
    }
    if (Enemy1Fire)
    {
      Enemy1move_fire();
    }
    if (!Enemy2Fire)
    {
      if (enemy2_p > 0)
        Enemy2Generate_Fire();
    }
    if (Enemy2Fire)
    {
      Enemy2move_fire();
    }
    if (!Enemy3Fire)
    {
      if (enemy3_p > 0)
        Enemy3Generate_Fire();
    }
    if (Enemy3Fire)
```

```
    {
      Enemy3move_fire();
    }
    if (!Enemy4Fire)
    {
      if (enemy4_p > 0)
        Enemy4Generate_Fire();
    }
    if (Enemy4Fire)
    {
      Enemy4move_fire();
    }
    if (enemy1_p == 0 && enemy2_p == 0 && enemy3_p == 0 && enemy4_p == 0 &&
health != 0)
    {
      system("cls");
      PrintWin();
      break;
    }
    else if (health <= 0)
    {
      system("cls");
      printgamelose();
      break;
    }
    Sleep(20);
  }
}
void setTextColor(int colorCode)
{
  std::cout << "\033[38;5;" << colorCode << "m";
}
void setBackgroundColor(int colorCode)
{
  std::cout << "\033[48;5;" << colorCode << "m";
}
void resetColors()
{
  std::cout << "\033[0m";
}
void printheader()
{
  system("cls");
  setBackgroundColor(0);
  setTextColor(51);
```

```
  cout << endl;
  cout << endl;
  cout << endl;
  cout << endl;
  cout << endl;
  cout <<
"\t\t\t\t\t\t\t     /$$$$$$$                  /$$
  /$$$$$$$ /$$   /$$                                     " << endl;
  cout << "\t\t\t\t\t\t\t    |__  $$__/                 |
$$                                |__  $$__/|__/  |
$$                                        " << endl;
  cout << "\t\t\t\t\t\t\t       | $$  /$$$$$$  /$$$$$$$ |
$$   /$$  /$$$$$$   /$$$$$$          | $$    /$$
/$$$$$$    /$$$$$$  /$$$$$$$   /$$$$$$$        " << endl;
  cout << "\t\t\t\t\t\t\t       | $$ |____  $$| $$__  $$| $$  /$$/ /$$__  $$
/$$__  $$         | $$   | $$|_  $$_/   |____  $$| $$__  $$ /$$_____/        " <<
endl;
  cout << "\t\t\t\t\t\t\t       | $$  /$$$$$$$| $$  \\ $$| $$$$$$/ | $$$$$$$$|
$$  \\__/         | $$   | $$  | $$      /$$$$$$$| $$  \\ $$|  $$$$$$      " <<
endl;
  cout << "\t\t\t\t\t\t\t       | $$ /$$__  $$| $$  | $$| $$_  $$ | $$_____/|
$$               | $$   | $$  | $$ /$$ /$$__  $$| $$  | $$ \\____  $$       " <<
endl;
  cout << "\t\t\t\t\t\t\t       | $$|  $$$$$$$| $$  | $$| $$ \\  $$|  $$$$$$$|
$$               | $$   | $$  |  $$$$/|  $$$$$$$| $$  | $$ /$$$$$$$/       " <<
endl;
  cout << "\t\t\t\t\t\t\t       |__/ \\_______/|__/  |__/|__/  \\__/
\\_______/|__/               |__/   |__/   \\___/   \\_______/|__/  |__/|_______/
   " << endl;
  resetColors();
}
void printcar()
{
  setBackgroundColor(0);
  setTextColor(201);
  cout <<
"


                          " << endl;
  cout <<
"


                        " << endl;
  cout <<
"
```

```
                                " << endl;
  cout <<
"
                                " << endl;
  cout <<
"                                                         ___________
                               " << endl;
  cout << "                                                       __..--
\"\"\"\"\"           \"\"\"\"--
..__
                             " << endl;
  cout << "                                                   _.-\"\"\"\"\"\"\"\"\"\"\"-----
...      ______
`.
                            " << endl;
  cout << "                                                 .-\"                        l ,-
\"    \\ \"-
.`.
                             " << endl;
  cout << "                                             .-\"                         ;
;        ;   \\ \"\"--.._                                   --------------------
----------                                        " << endl;
  cout << "                                           .'                         :
:          |    ;      .l                                         |
       |                                        " << endl;
  cout << "                                        _.._.'                          ;
;  ___     |    ;    .' :                                         |     1.  START
GAME          |                                       " << endl;
  cout << "                                       (   .'                           :
:  :   \".  :..-
'   .'    ;                                      |                            |
                               " << endl;
  cout << "                                        )'                            | ;  ;
__.'-\"(     .'  .--.:                                    -------------------------
-----                                            " << endl;
  cout << "                              ___...-'\"\"\"\"----....____         ______.-' :-
/.'       \\_.-'  .' .-
.\\l                                          |                            |
                       " << endl;
  cout << "              __..--
\"                           \"\"\"\"\"\"          /\\\"           ;    /
```

```
.gs./\\;                                    |     2.  OPTIONS            |
                          " << endl;
  cout << "        _.-
\"                                            /  ;          |   .
d$P\"Tb                                  |                            |
                       " << endl;
  cout << "     .-\"-
,                        ____                        /   |          :   ;:$$   $;
                                  ----------------------------
-                                 " << endl;
  cout << "   .'     ;                      ,\"    \"\"--
..__               /    :          |   $$$;   :$
  |                              |                                    " << endl;
  cout << "  /\"-._  /                       ;       ____..-'    .-\"\"\"-
.  /     :          ;  _$$$;   :$                                    |     3.   EXI
T              |                                  " << endl;
  cout << " :     \"\"--..._          ___....+---
\"\"\"\"\"          .'  _._  \\/      |         _:-\"
$$$;   :$                                 |                            |
                       " << endl;
  cout << "
;                                          /  .d$$$b./       ;      .-
\".'   :$$$   $P                             ----------------------------
-                                 " << endl;
  setTextColor(226);
  cout << " :            .------...____                        :  dP'
`T$P        |   .-\" .'
_.gd$$$$b_d$'
                         " << endl;
  cout << " ;    __.....----|    bug    |----....____         | :$     $b        :
.'   (.-
\"  `T$$$$$$P'
                           " << endl;
  cout << " ;  .';       '-----...____;       /    \"-.      ; $;     :$;_____..-
\"  .-
\"
                            " << endl;
  cout << " : /  :                          /        \\__..-':$       $$ ;-
.    .-
\"
                              " << endl;
  cout << " Y    ;                        /          ;     $;       :$;|  `.-
\"
                                    " << endl;
```

```
  cout << " :    :                        /           |     $$       $$;:.-
\"
                                          " << endl;
  cout << "
'$$$ggggp...____           /            :     :$;     :$$

                       " << endl;
  cout << "  $$$$$$$$$$$$   \"\"\"\"----
...:________....gggg$$$$$$     $$;
                                                                    " <<
endl;
  cout <<
"  'T$$$$$$$$P'                          T$$$$$$$$$b._.d$P

                        " << endl;
  cout <<
"    `T$$$$P'                            T$$$$$$$$$$$$$P

                         " << endl;
  cout <<
"                                         `T$$$$$$$$$P'

                        " << endl;

  resetColors(); // Reset colors to default
  system("pause");
}
void printmaze()
{
  system("cls");
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 11);
  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 <<
"|

                                                    | " << 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 <<
"|

                                                    | " << 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;
}
void gotoxy(int x, int y)
{
  COORD coordinates;
  coordinates.X = x;
  coordinates.Y = y;
  SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
char getCharAtxy(short int x, short int y)
{
  CHAR_INFO ci;
```

```
  COORD xy = {0, 0};
  SMALL_RECT rect = {x, y, x, y};
  COORD coordBufSize;
  coordBufSize.X = 1;
  coordBufSize.Y = 1;
  return ReadConsoleOutput(GetStdHandle(STD_OUTPUT_HANDLE), &ci, coordBufSize,
xy, &rect) ? ci.Char.AsciiChar

        : ' ';
}
void hideCursor()
{
  HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
  CONSOLE_CURSOR_INFO cursorInfo;
  GetConsoleCursorInfo(consoleHandle, &cursorInfo);
  cursorInfo.bVisible = false; // Set cursor visibility to false
  SetConsoleCursorInfo(consoleHandle, &cursorInfo);
}
void printPlayer()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
  gotoxy(xP, yP);
  cout << "       _  _     ";
  gotoxy(xP, yP + 1);
  cout << "    __//  |___* ";
  gotoxy(xP, yP + 2);
  cout << "  *|'_ '--' _ # ";
  gotoxy(xP, yP + 3);
  cout << "  *-(_)----(_)*";
}
void erasePlayer()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
  gotoxy(xP, yP);
  cout << "                ";
  gotoxy(xP, yP + 1);
  cout << "                ";
  gotoxy(xP, yP + 2);
  cout << "                ";
  gotoxy(xP, yP + 3);
  cout << "                ";
}

void printEnemy1()
{
```

```
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  if (xP > x1)
  {
    gotoxy(x1, y1);
    cout << "  /UMME AYMEN)=========";
    gotoxy(x1, y1 + 1);
    cout << "/**************\\      ";
    gotoxy(x1, y1 + 2);
    cout << "\\@@@@@@@@@@@@@@/      ";
  }
  else if (xP < x1)
  {
    gotoxy(x1, y1);
    cout << "=========(UMME AYMEN\\ ";
    gotoxy(x1, y1 + 1);
    cout << "      /**************\\";
    gotoxy(x1, y1 + 2);
    cout << "      \\@@@@@@@@@@@@@@/";
  }
}
void printEnemy2()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  if (xP > x2)
  {
    gotoxy(x2, y2);
    cout << "  /ANAS JABBAR)========";
    gotoxy(x2, y2 + 1);
    cout << "/**************\\      ";
    gotoxy(x2, y2 + 2);
    cout << "\\@@@@@@@@@@@@@@/      ";
  }
  else if (xP < x2)
  {
    gotoxy(x2, y2);
    cout << "=========(ANAS JABBAR\\\";
    gotoxy(x2, y2 + 1);
    cout << "      /**************\\\";
    gotoxy(x2, y2 + 2);
    cout << "      \\@@@@@@@@@@@@@@/";
  }
}
void printEnemy3()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
```

```
  if (xP > x3)
  {
    gotoxy(x3, y3);
    cout << "  /ZAINAB EMAN)========";
    gotoxy(x3, y3 + 1);
    cout << "/**************\\      ";
    gotoxy(x3, y3 + 2);
    cout << "\\@@@@@@@@@@@@@@/      ";
  }
  else if (xP < x3)
  {
    gotoxy(x3, y3);
    cout << "========(ZAINAB EMAN\\ ";
    gotoxy(x3, y3 + 1);
    cout << "      /**************\\";
    gotoxy(x3, y3 + 2);
    cout << "      \\@@@@@@@@@@@@@@/";
  }
}
void printEnemy4()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  if (xP > x4)
  {
    gotoxy(x4, y4);
    cout << "  /ABDUL JABAR)========";
    gotoxy(x4, y4 + 1);
    cout << "/**************\\      ";
    gotoxy(x4, y4 + 2);
    cout << "\\@@@@@@@@@@@@@@/      ";
  }
  else if (xP < x4)
  {
    gotoxy(x4, y4);
    cout << "========(ABDUL JABAR\\ ";
    gotoxy(x4, y4 + 1);
    cout << "      /**************\\";
    gotoxy(x4, y4 + 2);
    cout << "      \\@@@@@@@@@@@@@@/";
  }
}
void eraseEnemy1()
{
  {
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
```

```
    gotoxy(x1, y1);
    cout << "                          ";
    gotoxy(x1, y1 + 1);
    cout << "                          ";
    gotoxy(x1, y1 + 2);
    cout << "                          ";
  }
}
void eraseEnemy2()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  gotoxy(x2, y2);
  cout << "                          ";
  gotoxy(x2, y2 + 1);
  cout << "                          ";
  gotoxy(x2, y2 + 2);
  cout << "                          ";
}
void eraseEnemy3()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  gotoxy(x3, y3);
  cout << "                          ";
  gotoxy(x3, y3 + 1);
  cout << "                          ";
  gotoxy(x3, y3 + 2);
  cout << "                          ";
}
void eraseEnemy4()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  gotoxy(x4, y4);
  cout << "                          ";
  gotoxy(x4, y4 + 1);
  cout << "                          ";
  gotoxy(x4, y4 + 2);
  cout << "                          ";
}
void moveplayerUp()
{
  if ((getCharAtxy((xP), yP - 1) == ' ') && (getCharAtxy((xP + 1), yP - 1) == '
') && (getCharAtxy((xP + 2), yP - 1) == ' ') && (getCharAtxy((xP + 3), yP - 1) ==
' ') && (getCharAtxy((xP + 4), yP - 1) == ' ') && (getCharAtxy((xP + 5), yP - 1)
== ' ') && (getCharAtxy((xP + 6), yP - 1) == ' ') && (getCharAtxy((xP + 7), yP -
1) == ' ') && (getCharAtxy((xP + 8), yP - 1) == ' ') && (getCharAtxy((xP + 9), yP
```

```
- 1) == ' ') && (getCharAtxy((xP + 10), yP - 1) == ' ') && (getCharAtxy((xP +
11), yP - 1) == ' ') && (getCharAtxy((xP + 12), yP - 1)) && (getCharAtxy((xP +
13), yP - 1) == ' ') && (getCharAtxy((xP + 14), yP - 1) == ' ') &&
(getCharAtxy((xP + 15), yP - 1) == ' '))
  {

    erasePlayer();
    yP = yP - 1;
    printPlayer();
  }
}
void moveplayerDown()
{
  if ((getCharAtxy(xP, yP + 4) == ' ') && (getCharAtxy(xP + 1, yP + 4) == ' ') &&
(getCharAtxy(xP + 2, yP + 4) == ' ') && (getCharAtxy(xP + 3, yP + 4) == ' ') &&
(getCharAtxy(xP + 4, yP + 4) == ' ') && (getCharAtxy(xP + 5, yP + 4) == ' ') &&
(getCharAtxy(xP + 6, yP + 4) == ' ') && (getCharAtxy(xP + 7, yP + 4) == ' ') &&
(getCharAtxy(xP + 8, yP + 4) == ' ') && (getCharAtxy(xP + 9, yP + 4) == ' ') &&
(getCharAtxy(xP + 10, yP + 4) == ' ') && (getCharAtxy(xP + 11, yP + 4) == ' ') &&
(getCharAtxy(xP + 12, yP + 4) == ' ') && (getCharAtxy(xP + 13, yP + 4) == ' ') &&
(getCharAtxy(xP + 14, yP + 4) == ' ') && (getCharAtxy(xP + 15, yP + 4) == ' '))
  {
    erasePlayer();
    yP = yP + 1;
    printPlayer();
  }
}
void moveplayerLeft()
{
  if ((getCharAtxy(xP - 1, yP) == ' ') && (getCharAtxy(xP - 1, yP + 1) == ' ') &&
(getCharAtxy(xP - 1, yP + 2) == ' ') && (getCharAtxy(xP - 1, yP + 3) == ' '))
  {
    erasePlayer();
    xP = xP - 1;
    printPlayer();
  }
}
void moveplayerRight()
{
  if ((getCharAtxy(xP + 16, yP) == ' ') && (getCharAtxy(xP + 16, yP + 1) == ' ')
&& (getCharAtxy(xP + 16, yP + 2) == ' ') && (getCharAtxy(xP + 16, yP + 3) == '
'))
  {
    erasePlayer();
    xP = xP + 1;
```

```
    printPlayer();
  }
}
void moveEnemy1()
{

  static bool movingRight = true;
  eraseEnemy1();

  if (movingRight)
  {
    x1 = x1 + 1;
    if (x1 == 190)
    {
      movingRight = false;
    }
  }
  else
  {
    x1 = x1 - 1;
    if (x1 == 3)
    {
      movingRight = true;
    }
  }
  printEnemy1();
}
void moveEnemy2()
{

  static bool movingRight = true;

  eraseEnemy2();

  if (movingRight)
  {
    x2 = x2 + 1;
    if (x2 == 190)
    {
      movingRight = false;
    }
  }
  else
  {
    x2 = x2 - 1;
```

```
    if (x2 == 3)
    {
      movingRight = true;
    }
  }
  printEnemy2();
}
void moveEnemy3()
{
  static bool movingRight = true;
  eraseEnemy3();

  if (movingRight)
  {
    x3 = x3 + 1;
    if (x3 == 190)
    {
      movingRight = false;
    }
  }
  else
  {
    x3 = x3 - 1;
    if (x3 == 3)
    {
      movingRight = true;
    }
  }
  printEnemy3();
}
void moveEnemy4()
{

  static bool movingRight = true;

  eraseEnemy4();

  if (movingRight)
  {
    x4 = x4 + 1;
    if (x4 == 190)
    {
      movingRight = false;
    }
  }
```

```
  else
  {
    x4 = x4 - 1;
    if (x4 == 3)
    {
      movingRight = true;
    }
  }
  printEnemy4();
}

void generatebullet()
{
  bulletx[bulletcount] = xP - 1;
  bullety[bulletcount] = yP + 2;
  isBulletActive[bulletcount] = true;
  gotoxy(bulletx[bulletcount], bullety[bulletcount]);
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4);
  cout << bulletchar;
  bulletcount++;
}
void movebullet()
{

  for (int x = bulletcount - 1; x >= 0; x--)
  {

    // if (isBulletActive[x])
    // {

      int nextX = bulletx[x] - 1;
      int nextY = bullety[x];

      char next = getCharAtxy(nextX, nextY);

      if (next == ' ')
      {
        // Move bullet
        erasebullet(bulletx[x], bullety[x]);

        bulletx[x] = nextX;
        bullety[x] = nextY;

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

```
      else if (next == '|' || next == '=')
      {
        erasebullet(bulletx[x], bullety[x]);
        // makebulletInactive(x);

        deletebullet(x);
      }
       else
      {
        // makebulletInactive(x);
        erasebullet(bulletx[x], bullety[x]);
        deletebullet(x);
     }
    }
  // }
}

void printbullet(int x, int y)
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 8);
  gotoxy(x, y);
  cout << bulletchar;
}
void erasebullet(int x, int y)
{
  gotoxy(x, y);
  cout << " ";
}

void makebulletInactive(int index)
{
  // isBulletActive[index] = false;
}
void deletebullet(int index)
{
  for (int i = index; i < bulletcount - 1; i++)
  {
    bulletx[i] = bulletx[i + 1];
    bullety[i] = bullety[i + 1];
    isBulletActive[i] = isBulletActive[i+1];
  }
  bulletcount--;
}

void bulletcollisionwidenemy()
```

```
{
  for (int x = 0; x < bulletcount; x++)
  {
    if (isBulletActive[x])
    {
      if ((bullety[x] >= y1 && bullety[x] <= (y1 + 3)) && (bulletx[x] == x1 + 14)
&& enemy1_p != 0)
      {
        // Handle collision with enemy1
        addscore();
        updatescore();
        d_p1();
        update_power1();
        erasebullet(bulletx[x], bullety[x]);
        isBulletActive[x] = false;
        if (enemy1_p <= 0)
        {
          eraseEnemy1();
          deletebullet(x);
        }
      }
      if ((bulletx[x] >= x2 && bulletx[x] <= (x2 + 14)) && ((bullety[x] == y2 ||
bullety[x] == y2 + 1 || bullety[x] == y2 + 2 || bullety[x] == y2 + 3)) &&
enemy2_p != 0)
      {
        // Handle collision with enemy1
        addscore();
        updatescore();
        d_p2();
        erasebullet(bulletx[x], bullety[x]);
        isBulletActive[x] = false;
        update_power2();

        if (enemy2_p <= 0)
        {
          eraseEnemy2();
          deletebullet(x);
          break;
        }
      }
      if ((bulletx[x] >= x3 && bulletx[x] <= (x3 + 14)) && ((bullety[x] == y3 ||
bullety[x] == y3 + 1 || bullety[x] == y3 + 2 || bullety[x] == y3 + 3)) &&
enemy3_p != 0)
      {
        // Handle collision with enemy1
```

```
        addscore();
        updatescore();
        d_p3();
        erasebullet(bulletx[x], bullety[x]);
        isBulletActive[x] = false;
        update_power3();
        if (enemy3_p <= 0)
        {
          eraseEnemy3();
          deletebullet(x);
          break;
        }
      }

      if ((bulletx[x] >= x4 && bulletx[x] <= (x4 + 14)) && ((bullety[x] == y4) ||
(bullety[x] == y4 + 1) || (bullety[x] == y4 + 2) || (bullety[x] == y4 + 3)) &&
enemy4_p != 0)
      {
        addscore();
        updatescore();
        d_p4();
        erasebullet(bulletx[x], bullety[x]);
        isBulletActive[x] = false;
        update_power4();
        if (enemy4_p <= 0)
        {
          eraseEnemy4();
          deletebullet(x);
          break;
        }
      }
    }
  }
}
void updatescore()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
  gotoxy(228, 11);
  cout << score << endl;
}
void updateHealth()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
  gotoxy(228, 32);
  cout << health << endl;
```

```
}

void addscore()
{
  score = score + 1;
}
void print_info()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
  gotoxy(219, 10);
  cout << "--------------" << endl;
  gotoxy(219, 11);
  cout << " SCORE:       " << endl;
  gotoxy(219, 12);
  cout << "--------------" << endl;
  gotoxy(219, 17);
  cout << "CONTROL " << endl;
  gotoxy(219, 18);
  cout << "-------------" << endl;
  gotoxy(219, 19);
  cout << "Up ar - Up " << endl;
  gotoxy(219, 20);
  cout << "Down ar-Down" << endl;
  gotoxy(219, 21);
  cout << "Left ar-Left" << endl;
  gotoxy(219, 22);
  cout << "Right ar-Right" << endl;
  gotoxy(219, 23);
  cout << "Spacebar-shoot" << endl;
  gotoxy(219, 24);
  cout << "--------------" << endl;
  gotoxy(219, 26);
  cout << " p1: ";
  gotoxy(219, 27);
  cout << " p2: ";
  gotoxy(219, 28);
  cout << " p3: ";
  gotoxy(219, 29);
  cout << " p4: ";
  gotoxy(219, 31);
  cout << "--------------" << endl;
  gotoxy(219, 32);
  cout << "HEALTH:       " << endl;
  gotoxy(219, 33);
  cout << "--------------" << endl;
```

```
}

void setColor(int colorCode)
{
  HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
  SetConsoleTextAttribute(consoleHandle, colorCode);
}
void d_p1()
{
  enemy1_p = enemy1_p - 5;
}
void d_p2()
{
  enemy2_p = enemy2_p - 5;
}
void d_p3()
{
  enemy3_p = enemy3_p - 5;
}
void d_p4()
{
  enemy4_p = enemy4_p - 5;
}
void DecreasePlayerHealth()
{
  health = health - 10;
}
void update_power1()
{
  gotoxy(225, 26);
  cout << enemy1_p;
}
void update_power2()
{
  gotoxy(225, 27);
  cout << enemy2_p;
}
void update_power3()
{
  gotoxy(225, 28);
  cout << enemy3_p;
}
void update_power4()
{
  gotoxy(225, 29);
```

```
  cout << enemy4_p;
}
void Enemy1Generate_Fire()
{
  if (x1 < xP)
  {
    B1x = (x1 + 24) + 1;
    B1y = y1;
    Enemy1Fire = true;
  }
  else if (x1 > xP)
  {
    B1x = x1 - 1;
    B1y = y1;
    Enemy1Fire = true;
  }
}
void Enemy1move_fire()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  gotoxy(B1x, B1y);
  cout << " ";
  if (x1 < xP && ((getCharAtxy(B1x + 1, B1y) == ' ') || (getCharAtxy(B1x + 1,
B1y) == '@')))
  {
    B1x = B1x + 1;
    gotoxy(B1x, B1y);
    cout << "o";
  }
  else if (x1 > xP && ((getCharAtxy(B1x - 1, B1y) == ' ') || (getCharAtxy(B1x -
1, B1y) == '@')))
  {
    B1x = B1x - 1;
    gotoxy(B1x, B1y);
    cout << "o";
  }
  else if (getCharAtxy(B1x + 1, B1y == '*') || getCharAtxy(B1x + 1, B1y == '#'))
  {
    health = health - 1;
    updateHealth();
    gotoxy(B1x, B1y);
    cout << " ";
    Enemy1Fire = false;
  }
  else if (getCharAtxy(B1x - 1, B1y == '*') || getCharAtxy(B1x - 1, B1y == '#'))
```

```
  {
    health = health - 1;
    updateHealth();
    gotoxy(B1x, B1y);
    cout << " ";
    Enemy1Fire = false;
  }
  else
  {
    Enemy1Fire = false;
  }
}
void Enemy2Generate_Fire()
{
  if (x2 < xP)
  {
    B2x = (x2 + 24) + 1;
    B2y = y2;
    Enemy2Fire = true;
  }
  else if (x2 > xP)
  {
    B2x = x2 - 1;
    B2y = y2;
    Enemy2Fire = true;
  }
}
void Enemy2move_fire()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  gotoxy(B2x, B2y);
  cout << " ";
  if (x2 < xP && ((getCharAtxy(B2x + 1, B2y) == ' ') || (getCharAtxy(B2x + 1,
B2y) == '@')))
  {
    B2x = B2x + 1;
    gotoxy(B2x, B2y);
    cout << "o";
  }
  else if (x2 > xP && ((getCharAtxy(B2x - 1, B2y) == ' ') || (getCharAtxy(B2x -
1, B2y) == '@')))
  {
    B2x = B2x - 1;
    gotoxy(B2x, B2y);
    cout << "o";
```

```
  }
  else if (getCharAtxy(B2x + 1, B2y == '*') || getCharAtxy(B2x + 1, B2y == '#'))
  {
    health = health - 1;
    updateHealth();
    gotoxy(B2x, B2y);
    cout << " ";
    Enemy2Fire = false;
  }
  else if (getCharAtxy(B2x - 1, B2y == '*') || getCharAtxy(B2x - 1, B2y == '#'))

  {
    health = health - 1;
    updateHealth();
    gotoxy(B2x, B2y);
    cout << " ";
    Enemy2Fire = false;
  }
  else
  {
    Enemy2Fire = false;
  }
}
void Enemy3Generate_Fire()
{
  if (x3 < xP)
  {
    B3x = (x3 + 24) + 1;
    B3y = y3;
    Enemy3Fire = true;
  }
  if (x3 > xP)
  {
    B3x = x3 - 1;
    B3y = y3;
    Enemy3Fire = true;
  }
}
void Enemy3move_fire()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  gotoxy(B3x, B3y);
  cout << " ";
  if (x3 < xP && ((getCharAtxy(B3x + 1, B3y) == ' ') || (getCharAtxy(B3x + 1,
B3y) == '@')))
```

```
  {
    B3x = B3x + 1;
    gotoxy(B3x, B3y);
    cout << "o";
  }
  else if (x3 > xP && ((getCharAtxy(B3x - 1, B3y) == ' ') || (getCharAtxy(B3x -
1, B3y) == '@')))
  {
    B3x = B3x - 1;
    gotoxy(B3x, B3y);
    cout << "o";
  }
  else if (getCharAtxy(B3x + 1, B3y == '*') || getCharAtxy(B3x + 1, B3y == '#'))
  {
    health = health - 1;
    updateHealth();
    gotoxy(B3x, B3y);
    cout << " ";
    Enemy3Fire = false;
  }
  else if (getCharAtxy(B3x - 1, B3y == '#') || getCharAtxy(B3x - 1, B3y == '*'))
  {
    health = health - 1;
    updateHealth();
    gotoxy(B3x, B3y);
    cout << " ";
    Enemy3Fire = false;
  }
  else
  {
    Enemy3Fire = false;
  }
}
void Enemy4Generate_Fire()
{
  if (x4 < xP)
  {
    B4x = (x4 + 24) + 1;
    B4y = y4;
    Enemy4Fire = true;
  }
  else if (x4 > xP)
  {
    B4x = x4 - 1;
    B4y = y4;
```

```
    Enemy4Fire = true;
  }
}
void Enemy4move_fire()
{
  SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
  gotoxy(B4x, B4y);
  cout << " ";
  if (x4 < xP && ((getCharAtxy(B4x + 1, B4y) == ' ') || (getCharAtxy(B4x + 1,
B4y) == '@')))
  {
    B4x = B4x + 1;
    gotoxy(B4x, B4y);
    cout << "o";
  }
  else if (x4 > xP && ((getCharAtxy(B4x - 1, B4y) == ' ') || (getCharAtxy(B4x -
1, B4y) == '@')))
  {
    B4x = B4x - 1;
    gotoxy(B4x, B4y);
    cout << "o";
  }

  else if (getCharAtxy(B4x, B4y) == '*' || getCharAtxy(B4x, B4y) == '#')
  {
    health = health - 1;
    updateHealth();
    gotoxy(B4x, B4y);
    cout << " ";
    Enemy4Fire = false;
  }
  else if (getCharAtxy(B4x, B4y) == '*' || getCharAtxy(B4x, B4y) == '#')
  {
    health = health - 1;
    updateHealth();
    gotoxy(B4x, B4y);
    cout << " ";
    Enemy4Fire = false;
  }
  else
  {
    Enemy4Fire = false;
  }
}
void PrintWin()
```

```
{
  system("cls");
  setBackgroundColor(0);
  setTextColor(51);
  cout << endl;
  cout << endl;
  cout << endl;
  cout << endl;
  cout << endl;
  cout << "\t\t\t\t\t\t\t\t /$$     /$$ /$$$$$$  /$$   /$$       /$$      /$$
/$$$$$$ /$$   /$$\n";
  cout << "\t\t\t\t\t\t\t\t|  $$   /$$//$$__  $$| $$  | $$      | $$  /$ |
$$|_  $$_/| $$$ | $$\n";
  cout << "\t\t\t\t\t\t\t\t \\  $$ /$$/| $$  \\ $$| $$  | $$      | $$ /$$$|
$$  | $$  | $$$$| $$\n";
  cout << "\t\t\t\t\t\t\t\t  \\  $$$$/ | $$  | $$| $$  | $$      | $$/$$ $$ $$  |
$$  | $$ $$ $$\n";
  cout << "\t\t\t\t\t\t\t\t   \\  $$/  | $$  | $$| $$  | $$      | $$$$_  $$$$  |
$$  | $$  $$$$\n";
  cout << "\t\t\t\t\t\t\t\t    | $$   | $$  | $$| $$  | $$      | $$$/ \\  $$$  |
$$  | $$\\  $$$\n";
  cout << "\t\t\t\t\t\t\t\t    | $$   |  $$$$$$/|  $$$$$$/      | $$/   \\  $$
/$$$$$$| $$ \\  $$\n";
  cout <<
"\t\t\t\t\t\t\t\t    |__/    \\______/  \\______/       |__/     \\__/|______/|__
/  \\__/\n";
  resetColors();
}
void printgamelose()
{
  system("cls");
  setBackgroundColor(0);
  setTextColor(51);
  cout <<
"\t\t\t\t\t\t\t\t  /$$$$$$                                        /$$$$$$\n"
;
  cout << "\t\t\t\t\t\t\t\t
/$$__  $$                                     /$$__  $$\n";
  cout << "\t\t\t\t\t\t\t\t| $$  \\__/  /$$$$$$  /$$$$$$/$$$$   /$$$$$$      |
$$  \\ $$ /$$    /$$ /$$$$$$   /$$$$$$\n";
  cout << "\t\t\t\t\t\t\t\t| $$ /$$$$ |____  $$| $$_  $$_  $$ /$$__  $$      |
$$  | $$|  $$  /$$//$$__  $$ /$$__  $$\n";
  cout << "\t\t\t\t\t\t\t\t| $$|_  $$  /$$$$$$$| $$ \\ $$ \\ $$| $$|
$$$$$$$$      | $$  | $$ \\  $$/$$/| $$$$$$$$| $$  \\__/\n";
```

```
  cout << "\t\t\t\t\t\t\t\t\t| $$  \\ $$ /$$__  $$| $$ | $$ | $$| $$_____/      |
$$  | $$  \\  $$$/ | $$_____/| $$      \n";
  cout << "\t\t\t\t\t\t\t\t\t|  $$$$$$/|  $$$$$$$| $$ | $$ |
$$|  $$$$$$$      |  $$$$$$/   \\  $/  |  $$$$$$$| $$      \n";
  cout << "\t\t\t\t\t\t\t\t\t \\______/  \\_______/|__/ |__/ |__/
\\_______/       \\______/     \\_/    \\_______/|__/      \n";
  resetColors();
}
```