**ROAD RAGE**



Session 2023-2027

**Submitted by:**

|  |  |
| --- | --- |
| Muhammad Owais | 2023-CS-131 |

**Supervised by:**

Ma’am Maida Shahid

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Table of Contents

Contents

[**1) Background Story:** 3](#_Toc153948507)

[**2) Description of Game:** 3](#_Toc153948508)

[**3) Game Character Description:** 3](#_Toc153948509)

[**a) Player:** 3](#_Toc153948510)

[**i.** **Kaka:** 3](#_Toc153948511)

[**Enemies:** 3](#_Toc153948512)

[**ii.**  **Raka:** 3](#_Toc153948513)

[**iii.** **Shaka:** 3](#_Toc153948514)

[**Wireframes:** 4](#_Toc153948515)

[**a)** **Startup:** 4](#_Toc153948516)

[**b) Main Menu:** 4](#_Toc153948517)

[**c) Instructions:** 4](#_Toc153948518)

[**d) Level 1:** 5](#_Toc153948519)

[**Prototypes:** 5](#_Toc153948520)

[ **Variables:** 7](#_Toc153948521)

Table of Figures:

[Figure 1. Startup 4](#_Toc153948474)

[Figure 2. Main Menu 4](#_Toc153948475)

[Figure 3. Instructions 4](#_Toc153948476)

[Figure 4. Level 1 5](#_Toc153948477)

# **1) Background Story:**

The racing car is the one of the most famous application in the whole game. In the whole game the car chases each other and they beat each other together. For hurdles ther’s are army tank that shoot you in case of any laziness.

# **2) Description of Game:**

In my game the car moves up down left right. We have to safe from the army tank and from the enemy cars to shield ourself from the car. The player have remain safe from the army tank.

# **3) Game Character Description:**

## **a) Player:**

There is one main player in the game.

### **Army Tank:**

Army Tank is the main character in the game road rage, which shoot the player in any case.

## **Enemies:**

### **ii. Enemy Car:**

Enemy car moves along the same line in the opposite direction and we have to chase it.

# **Wireframes:**

## **Startup:**



Figure 1. Startup

## **b) Main Menu:**



Figure 2. Main Menu

## **c) Instructions:**



Figure 3. Instructions

## **d) Level 1:**

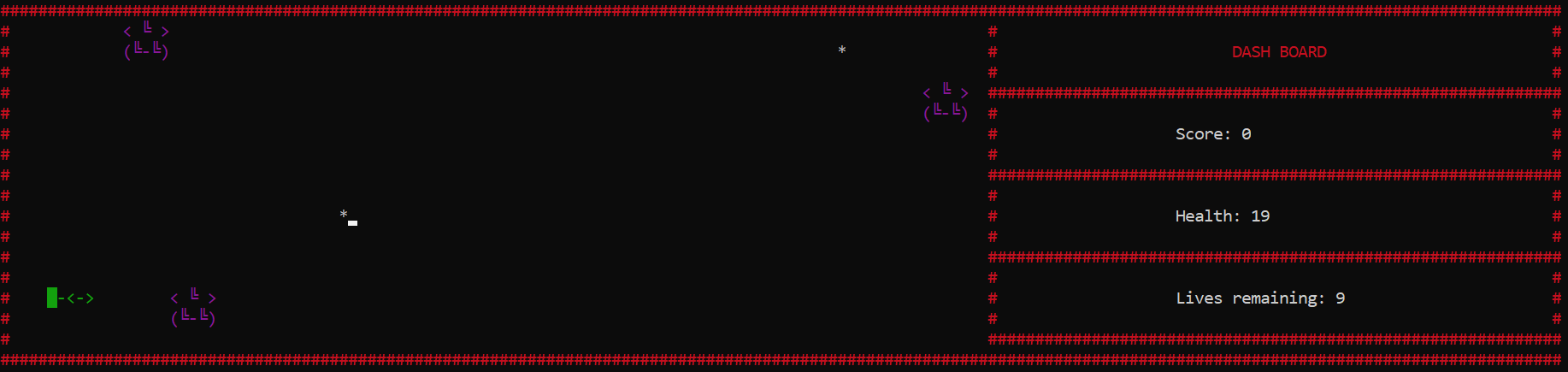
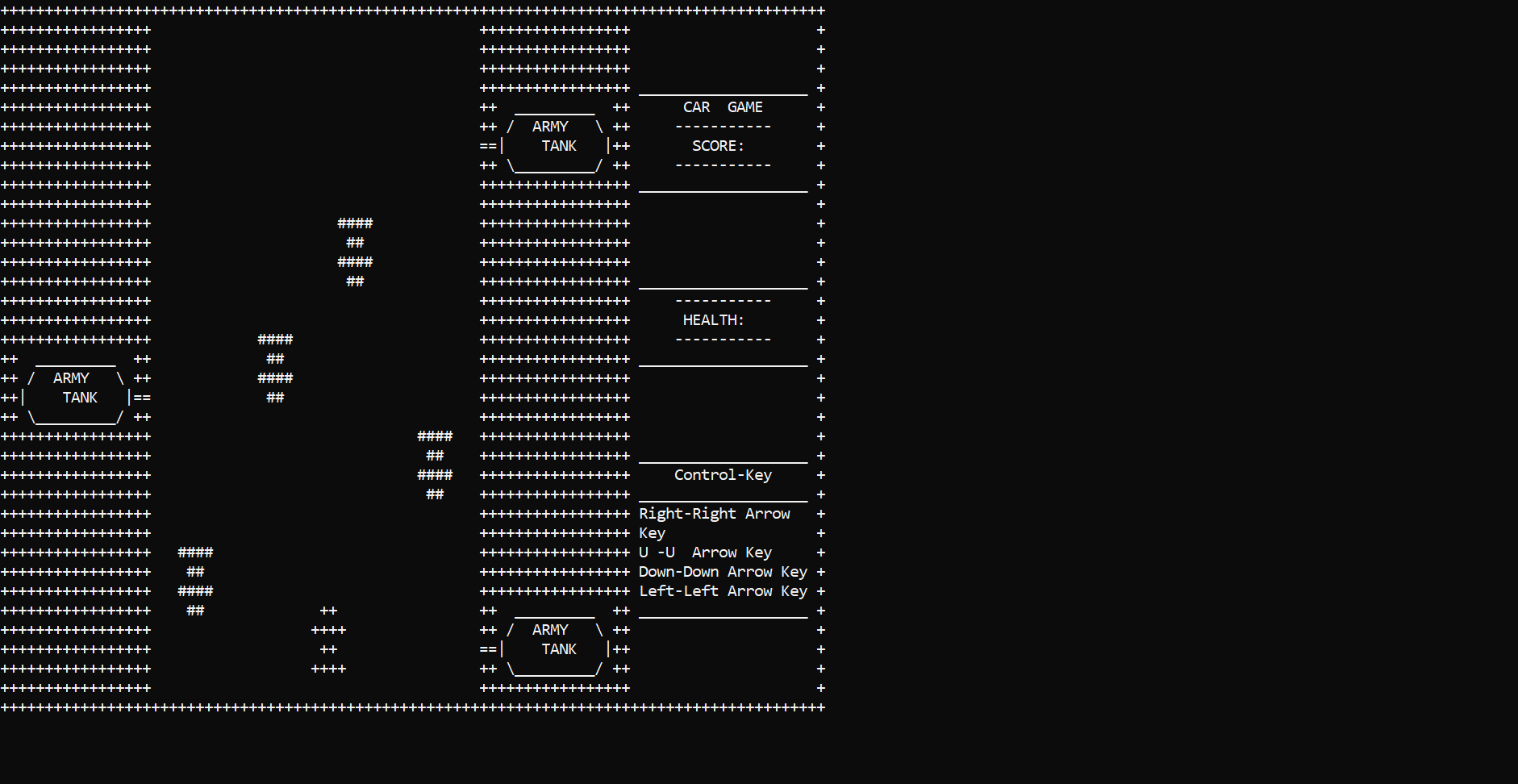


Figure 4. Level 1

# **Prototypes:**

* const int limit =100000;
* int arrx[limit];
* int arry[limit];
* bool active[limit];
* int bulletCount= 0;
* int bulletIndex=0;
* int firelimit =0;
* bool isactive[1000];
* void hideCursor();
* void hideCursor();
* void printmaize();
* void printplayer();
* void movePlayerLeft();
* void movePlayerRight();
* void movePlayerUp();
* void movePlayerDown();
* void menu(char &input);
* void printenemy();
* void printenemy1();
* void printenemy2();
* void printenemy3();
* void eraseenemy();
* void eraseenemy1();
* void eraseenemy2();
* void moveenemy();
* void moveenemy1();
* void erasebullet(int x, int y);
* void moveenemy2();
* void moveenemy3();
* void bullettank();
* void friing();
* void printbullet(int x, int y);
* void interface\_\_();
* void gamemenu();
* void instruction();
* void move();
* char getCharAtxy(short int x, short int y);
* bool collision();
* void clear();
* void bullettank();
* void bulletMove();
* void Score();void gotoxy(int x, int y);
* char getCharAtxy(short int x, short int y);
* void printShip(char ship[][5]);
* void eraseShip();
* void printMaze();
* void moveBullet();
* void generateBullet();
* void removeBulletFromArray(int index);
* void moveShipUp(char ship[][5]);
* void moveShipDown(char ship[][5]);
* void moveShipLeft(char ship[][5]);
* void moveShipRight(char ship[][5]);
* void moveEnemy(char enemy[2][5]);
* void printEnemy(char enemy[2][5]);
* void eraseEnemy();
* void moveenemy1( char enemy[2][5]);
* void printenemy1(char enemy[2][5]);
* void eraseenemy1();
* void moveenemy2( char enemy[2][5]);
* void printenemy2(char enemy[2][5]);
* void eraseenemy2();
* void printenemybullet(int, int);
* void eraseenemybullet(int, int);
* void removeenemybulletfromarray(int);
* void moveenemybullet();
* void generateenemybullet();
* void enemybulletcollisionwithplayer();
* void generateenemy1bullet();
* void moveenemy1bullet();
* void enemy1bulletcollisionwithplayer();
* void printenemy1bullet(int, int);
* void eraseenemy1bullet(int, int);
* void removeenemy1bulletfromarray(int);
* void generateenemy2bullet();
* void printenemy2bullet(int, int);
* void eraseenemy2bullet(int, int);
* void moveenemy2bullet();
* void removeenemy2bulletfromarray(int);
* void enemy2bulletcollisionwithplayer();

# **Variables:**

* char sign = 198;
* char sign3 = 206;
* char box = 219;
* int tx = 53;
* int ty =33;
* int px = 15;
* int py = 15;
* int ex = 20;
* int ey = 1;
* int enemy[3];
* int ex1 = 47;
* int ey1 = 1;
* int ex2 = 29;
* int ey2 = 1;
* int ex3 = 38;
* int ey3 = 1;
* int bx = 54;
* int by = 90;
* int bulletx = 0;
* int bullety = 0;
* int score = 0;
* int health =30;
* char sign1 = 188;
* char sign2 = 200;
* int bulletX[1000];
* int bulletY[1000];
* int bulletCount = 0;
* int count = 0;
* int enemybulletx[1000];
* int enemybullety[1000];
* int enemycount = 0;
* int enemy1bulletx[1000];
* int enemy1bullety[1000];
* int enemy1count = 0;
* int enemy2bulletx[1000];
* int enemy2bullety[1000];
* int enemy2count = 0;
* int bigEnemyBulletX[1000];
* int bigEnemyBulletY[1000];
* int bigEnemycount = 0;

# **Code:**

#include <iostream>

#include <conio.h>

#include <windows.h>

#include <string>

void hideCursor();

void hideCursor();

void printmaize();

void printplayer();

void movePlayerLeft();

void movePlayerRight();

void movePlayerUp();

void movePlayerDown();

void menu(char &input);

void printenemy();

void printenemy1();

void printenemy2();

void printenemy3();

void eraseenemy();

void eraseenemy1();

void eraseenemy2();

void moveenemy();

void moveenemy1();

void erasebullet(int x, int y);

void moveenemy2();

void moveenemy3();

void bullettank();

void friing();

void printbullet(int x, int y);

void interface\_\_();

void gamemenu();

void instruction();

void move();

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

bool collision();

void clear();

void bullettank();

void bulletMove();

void bullettank1();

void bulletMove1();

void bullettank2();

void bulletmove2();

void Score();

void healthplayer();

void gotoxy(int x, int y);

void gamecollision();

int tx = 53;

int ty = 33;

int tx1 = 17;

int ty1 = 20;

int tx2 = 53;

int ty2 = 7;

int px = 15;

int py = 15;

int ex = 20;

int ey = 1;

int enemy[3];

int ex1 = 47;

int ey1 = 1;

int ex2 = 29;

int ey2 = 1;

int ex3 = 38;

int ey3 = 1;

int bx = 54;

int by = 90;

int bulletx = 0;

int bullety = 0;

int score = 0;

int score1 = 0;

int score2=0;

int score3=0;

int score4=0;

int health1=30;

int health =30;

const int limit =100000;

int arrx1[limit];

int arry1[limit];

int arrx2[limit];

int arry2[limit];

int arrx[limit];

int arry[limit];

bool active[limit];

bool active1[limit];

bool active2[limit];

int bulletCount= 0;

int bulletCount1= 0;

int bulletCount2= 0;

int bulletIndex=0;

int bulletIndex1=0;

int bulletIndex2=0;

int firelimit =0;

bool isactive[1000];

using namespace std;

char star[38][94] = {

{"+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"},

{"+++++++++++++++++ +++++++++++++++++ +"},

{"+++++++++++++++++ +++++++++++++++++ +"},

{"+++++++++++++++++ +++++++++++++++++ +"},

{"+++++++++++++++++ +++++++++++++++++ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ +"},

{"+++++++++++++++++ ++ \_\_\_\_\_\_\_\_\_ ++ CAR GAME +"},

{"+++++++++++++++++ ++ / ARMY \\ ++ ----------- +"},

{"+++++++++++++++++ ==| TANK |++ SCORE: +"},

{"+++++++++++++++++ ++ \\\_\_\_\_\_\_\_\_\_/ ++ ----------- +"},

{"+++++++++++++++++ +++++++++++++++++ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ +"},

{"+++++++++++++++++ +++++++++++++++++ +"},

{"+++++++++++++++++ +++++++++++++++++ +"},

{"+++++++++++++++++ +++++++++++++++++ +"},

{"+++++++++++++++++ +++++++++++++++++ +"},

{"+++++++++++++++++ +++++++++++++++++ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ +"},

{"+++++++++++++++++ +++++++++++++++++ ----------- +"},

{"+++++++++++++++++ +++++++++++++++++ HEALTH: +"},

{"+++++++++++++++++ +++++++++++++++++ ----------- +"},

{"++ \_\_\_\_\_\_\_\_\_ ++ +++++++++++++++++ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ +"},

{"++ / ARMY \\ ++ +++++++++++++++++ +"},

{"++| TANK |== +++++++++++++++++ +"},

{"++ \\\_\_\_\_\_\_\_\_\_/ ++ +++++++++++++++++ +"},

{"+++++++++++++++++ +++++++++++++++++ +"},

{"+++++++++++++++++ +++++++++++++++++ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ +"},

{"+++++++++++++++++ +++++++++++++++++ Control-Key +"},

{"+++++++++++++++++ +++++++++++++++++ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ +"},

{"+++++++++++++++++ +++++++++++++++++ Right-Right Arrow +"},

{"+++++++++++++++++ +++++++++++++++++ Key +"},

{"+++++++++++++++++ +++++++++++++++++ Up-Up Arrow Key +"},

{"+++++++++++++++++ +++++++++++++++++ Down-Down Arrow Key +"},

{"+++++++++++++++++ +++++++++++++++++ Left-Left Arrow Key +"},

{"+++++++++++++++++ p ++ \_\_\_\_\_\_\_\_\_ ++ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ +"},

{"+++++++++++++++++ ++ / ARMY \\ ++ +"},

{"+++++++++++++++++ ==| TANK |++ +"},

{"+++++++++++++++++ ++ \\\_\_\_\_\_\_\_\_\_/ ++ +"},

{"+++++++++++++++++ +++++++++++++++++ +"},

{"+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"}};

char player[5][5] = {{" && "},

{"&&&&"},

{" && "},

{"&&&&"}};

char enemycar[5][5] = {{"####"},

{" ## "},

{"####"},

{" ## "}};

char erase[5][5] = {{" "},

{" "},

{" "},

{" "}};

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

main()

{

system("cls");

interface\_\_();

system("cls");

gamemenu();

// printmenu();

while (true)

{

char input;

menu(input);

if (input == '1')

{

string check;

int count = 0;

//Sleep(1000);

hideCursor();

system("cls");

printmaize();

printmaize();

printplayer();

printenemy();

printenemy1();

printenemy2();

printenemy3();

//Sleep(100);

while (true)

{

bulletMove();

bulletMove1();

bulletmove2();

if (GetAsyncKeyState(VK\_LEFT))

{

movePlayerLeft();

}

if (GetAsyncKeyState(VK\_RIGHT))

{

movePlayerRight();

}

if (GetAsyncKeyState(VK\_UP))

{

movePlayerUp();

}

if (GetAsyncKeyState(VK\_DOWN))

{

movePlayerDown();

}

if (GetAsyncKeyState(VK\_SPACE))

{

friing();

}

// move();

// bullettank();

moveenemy();

Score();

healthplayer();

if (count > 5)

moveenemy1();

if (count > 10)

moveenemy2();

if (count > 15)

moveenemy3();

// if(collision)

// return 0;

Sleep(50);

count++;

if(score1+score2+score3+score4==30)

{

system("cls");

cout<<"You Win."<<endl;

getch();

break;

//system("cls");

}

if(health<0)

{

system("cls");

cout<<"You Lose."<<endl;

getch();

break;

}

}

}

else if (input == '2')

{

instruction();

getch();

}

else if (input == '3')

{

break;

}

else

{

cout<<"Invalid Input."<<endl;

}

}

}

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 printmaize()

{

clear();

for (int row = 0; row < 38; row++)

{

for (int col = 0; col < 94; col++)

{

if (star[row][col] == 'p')

{

px = row;

py = col;

printplayer();

}

else

{

gotoxy(col, row);

cout << star[row][col];

}

}

}

}

void printplayer()

{

clear();

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

{

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

{

gotoxy(py + j, px + i);

cout << player[i][j];

}

}

}

void gotoxy(int x, int y)

{

COORD coordinates;

coordinates.X = x;

coordinates.Y = y;

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

}

void hideCursor()

{

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

CONSOLE\_CURSOR\_INFO cursorInfo;

GetConsoleCursorInfo(consoleHandle, &cursorInfo);

cursorInfo.bVisible = false;

SetConsoleCursorInfo(consoleHandle, &cursorInfo);

}

void clear()

{

COORD cursorPos;

cursorPos.X = 0;

cursorPos.Y = 0;

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

}

void erasePlayerLeft()

{

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

{

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

{

gotoxy(py + j, px + i);

cout << " ";

}

}

}

void erasePlayerRight()

{

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

{

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

{

gotoxy(py, px + i);

cout << " ";

}

}

}

void movePlayerLeft()

{

if (star[px][py - 1] == ' ')

{

erasePlayerRight();

star[px][py] = ' ';

py = py - 1;

star[px][py] = 'p';

clear();

printplayer();

}

}

void interface\_\_()

{

system("cls");

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 << " ########## ";

SetConsoleTextAttribute(h, 4);

cout << " \_\_\_\_ \_ \_\_\_\_ ";

SetConsoleTextAttribute(h, 15);

cout << " ######### " << endl;

cout << " ";

SetConsoleTextAttribute(h, 15);

cout << "## ";

SetConsoleTextAttribute(h, 4);

cout << "| \_ \\ \_\_\_ \_\_| | \_\_\_ | \_ \\ \_\_ \_ \_\_ \_ \_\_\_ ";

SetConsoleTextAttribute(h, 15);

cout << "## " << endl;

cout << " ";

SetConsoleTextAttribute(h, 15);

cout << "## ";

SetConsoleTextAttribute(h, 4);

cout << "| |\_) / \_ \\ / \_` |/ \_ \\ | |\_) / \_` |/ \_` |/ \_ \\ ";

SetConsoleTextAttribute(h, 15);

cout << "### " << endl;

cout << " ";

SetConsoleTextAttribute(h, 15);

cout << "## ";

SetConsoleTextAttribute(h, 4);

cout << "| \_ < (\_) | (\_| | \_\_/ | \_ < (\_| | (\_| | \_\_/ ";

SetConsoleTextAttribute(h, 15);

cout << "## " << endl;

cout << " ";

SetConsoleTextAttribute(h, 15);

cout << "### ";

SetConsoleTextAttribute(h, 4);

cout << "|\_| \\\_\\\_\_\_/ \\\_\_,\_|\\\_\_\_| |\_| \\\_\\\_\_,\_|\\\_\_, |\\\_\_\_| ";

SetConsoleTextAttribute(h, 15);

cout << "### " << endl;

cout << " #### ";

SetConsoleTextAttribute(h, 4);

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

SetConsoleTextAttribute(h, 15);

cout << " #### " << endl;

getch();

}

void gamemenu()

{

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 menu(char &input)

{

system("cls");

gamemenu();

gotoxy(65, 23);

cout << "1. Start." << endl;

gotoxy(65, 24);

cout << "2. Instruction." << endl;

gotoxy(65, 25);

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

gotoxy(65, 26);

cout << "Enter you Option...";

cin >> input;

}

void movePlayerRight()

{

if (star[px][py + 4] == ' ')

{

erasePlayerRight();

star[px][py] = ' ';

py = py + 1;

star[px][py] = 'p';

clear();

printplayer();

}

}

void movePlayerUp()

{

if (star[px - 1][py] == ' ')

{

erasePlayerRight();

star[px][py] = ' ';

px = px - 1;

star[px][py] = 'p';

clear();

printplayer();

}

}

void movePlayerDown()

{

if (star[px + 5][py] == ' ')

{

erasePlayerRight();

star[px][py] = ' ';

px = px + 1;

star[px][py] = 'p';

clear();

printplayer();

}

}

void printenemy()

{

gotoxy(ex, ey);

cout << "####" << endl;

gotoxy(ex, ey + 1);

cout << " ## " << endl;

gotoxy(ex, ey + 2);

cout << "####" << endl;

gotoxy(ex, ey + 3);

cout << " ## " << endl;

}

void printenemy1()

{

gotoxy(ex1, ey1);

cout << "####" << endl;

gotoxy(ex1, ey1 + 1);

cout << " ## " << endl;

gotoxy(ex1, ey1 + 2);

cout << "####" << endl;

gotoxy(ex1, ey1 + 3);

cout << " ## " << endl;

}

void printenemy2()

{

gotoxy(ex2, ey2);

cout << "####" << endl;

gotoxy(ex2, ey2 + 1);

cout << " ## " << endl;

gotoxy(ex2, ey2 + 2);

cout << "####" << endl;

gotoxy(ex2, ey2 + 3);

cout << " ## " << endl;

}

void printenemy3()

{

gotoxy(ex3, ey3);

cout << "####" << endl;

gotoxy(ex3, ey3 + 1);

cout << " ## " << endl;

gotoxy(ex3, ey3 + 2);

cout << "####" << endl;

gotoxy(ex3, ey3 + 3);

cout << " ## " << endl;

}

void eraseenemy()

{

gotoxy(ex, ey);

cout << " " << endl;

gotoxy(ex, ey + 1);

cout << " " << endl;

gotoxy(ex, ey + 2);

cout << " " << endl;

gotoxy(ex, ey + 3);

cout << " " << endl;

}

void eraseenemy1()

{

gotoxy(ex1, ey1);

cout << " " << endl;

gotoxy(ex1, ey1 + 1);

cout << " " << endl;

gotoxy(ex1, ey1 + 2);

cout << " " << endl;

gotoxy(ex1, ey1 + 3);

cout << " " << endl;

}

void eraseenemy2()

{

gotoxy(ex2, ey2);

cout << " " << endl;

gotoxy(ex2, ey2 + 1);

cout << " " << endl;

gotoxy(ex2, ey2 + 2);

cout << " " << endl;

gotoxy(ex2, ey2 + 3);

cout << " " << endl;

}

void eraseenemy3()

{

gotoxy(ex3, ey3);

cout << " " << endl;

gotoxy(ex3, ey3 + 1);

cout << " " << endl;

gotoxy(ex3, ey3 + 2);

cout << " " << endl;

gotoxy(ex3, ey3 + 3);

cout << " " << endl;

}

void moveenemy()

{

eraseenemy();

ey = ey + 1;

printenemy();

if (ey == 31)

{

eraseenemy();

moveenemy1();

// Sleep(300);

score1=score1+1;

ey = 1;

}

if(ey=='&')

{

system("cls");

cout<<"a";

getch();

}

}

void moveenemy1()

{

eraseenemy1();

ey1 = ey1 + 1;

printenemy1();

if (ey1 == 31)

{

eraseenemy1();

// Sleep(500);

score2=score2+1;

ey1 = 1;

}

}

void moveenemy2()

{

eraseenemy2();

ey2 = ey2 + 1;

printenemy2();

if (ey2 == 31)

{

eraseenemy2();

// Sleep(1000);

score3=score3+1;

ey2 = 1;

}

}

void moveenemy3()

{

eraseenemy3();

ey3 = ey3 + 1;

printenemy3();

if (ey3 == 31)

{

eraseenemy3();

// Sleep(1000);

score4=score4+1;

ey3 = 1;

}

}

void healthplayer()

{

gotoxy(77, 16);

cout<<"Health: "<<health;

}

void Score()

{

gotoxy(78, 7);

cout<<"Score: "<<score1+score2+score3+score4;

}

bool collision()

{

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

{

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

{

if (player[i - 1][j] == '#')

{

return true;

}

}

}

return false;

}

void friing()

{

if (getCharAtxy(px - 1, py + 2) != '+')

{

arrx[bulletCount] = px - 1;

arry[bulletCount] = py + 2;

isactive[bulletCount] = true;

// gotoxy(px, py - 1);

printbullet(px - 1, py + 2);

bulletCount++;

}

}

void erasebullet(int x, int y)

{

gotoxy(x, y);

cout << " ";

}

void printbullet(int x, int y)

{

gotoxy(x, y);

cout << ".";

}

void move()

{

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

{

if (isactive[i] == true)

{

if (getCharAtxy(arrx[i] - 1, arry[i]) != ' ')

{

erasebullet(arrx[i], arry[i]);

isactive[i] = false;

}

else

{

erasebullet(arrx[i], arry[i]);

arrx[i] = arrx[i] - 1;

printbullet(arrx[i], arry[i]);

}

}

}

}

void instruction()

{

system("cls");

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 << " .. To initiate a leftward movement, kindly depress the 'left' key. .. " << endl;

cout << " .. To initiate a rightward movement, kindly depress the 'right' key. .. " << endl;

cout << " .. To initiate a upward movement, kindly depress the 'up' key. .. " << endl;

cout << " .. To initiate a downward movement, kindly depress the 'down' key. .. " << endl;

cout << " .. We need to evade tank projectiles and pursue the racing car. .. " << endl;

cout << " .. As soon as a car descends below, its score will increase by 2. .. " << endl;

cout << " .. As soon as your score reaches 50, you will continue winning. And game over. .. " << endl;

cout << " .. .. " << endl;

cout << " .. .. " << endl;

cout << " .. .. " << endl;

cout << " .. .. " << endl;

cout << " ................................................................................... " << endl;

}

void bullettank()

{

int tbx = tx;

int tby = ty;

gotoxy(tbx, tby);

cout<<"\*";

arrx[bulletIndex] = tbx;

arry[bulletIndex] = tby;

active[bulletIndex] = true;

bulletCount++;

bulletIndex++;

}

void bulletMove()

{

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

{

if(active[i] == true)

{

if(getCharAtxy(arrx[i] - 1, arry[i]) == ' ')

{

gotoxy(arrx[i], arry[i]);

cout<<" ";

arrx[i] = arrx[i] - 1;

gotoxy(arrx[i], arry[i]);

cout<<"\*";

}

else if(getCharAtxy(arrx[i] - 1, arry[i]) != ' ')

{

gotoxy(arrx[i], arry[i]);

cout<<" ";

}

if(getCharAtxy(arrx[i] - 1, arry[i]) == '&')

{

health=health1-1;

}

}

}

if(firelimit > 15)

{

bullettank();

firelimit=0;

}

else

{

firelimit++;

}

}

void bullettank1()

{

int tbx1 = tx1;

int tby1 = ty1;

gotoxy(tbx1, tby1);

cout<<"\*";

arrx1[bulletIndex1] = tbx1;

arry1[bulletIndex1] = tby1;

active1[bulletIndex1] = true;

bulletCount1++;

bulletIndex1++;

}

void bulletMove1()

{

for(int i=0; i<bulletCount1 ;i++)

{

if(active1[i] == true)

{

if(getCharAtxy(arrx1[i] + 1, arry1[i]) == ' ')

{

gotoxy(arrx1[i], arry1[i]);

cout<<" ";

arrx1[i] = arrx1[i] + 1;

gotoxy(arrx1[i], arry1[i]);

cout<<"\*";

}

else if(getCharAtxy(arrx1[i] + 1, arry1[i]) != ' ')

{

gotoxy(arrx1[i], arry1[i]);

cout<<" ";

}

if(getCharAtxy(arrx1[i] + 1, arry1[i]) == '&')

{

health=health1-1;

}

}

}

if(firelimit > 15)

{

bullettank1();

firelimit=0;

}

else

{

firelimit++;

}

}

void bullettank2()

{

int tbx2 = tx2;

int tby2 = ty2;

gotoxy(tbx2, tby2);

cout<<"\*";

arrx2[bulletIndex2] = tbx2;

arry2[bulletIndex2] = tby2;

active2[bulletIndex2] = true;

bulletCount2++;

bulletIndex2++;

}

void bulletmove2()

{

for(int i=0; i<bulletCount2 ;i++)

{

if(active2[i] == true)

{

if(getCharAtxy(arrx2[i] - 1, arry2[i]) == ' ')

{

gotoxy(arrx2[i], arry2[i]);

cout<<" ";

arrx2[i] = arrx2[i] - 1;

gotoxy(arrx2[i], arry2[i]);

cout<<"\*";

}

else if(getCharAtxy(arrx2[i] - 1, arry2[i]) != ' ')

{

gotoxy(arrx2[i], arry2[i]);

cout<<" ";

}

if(getCharAtxy(arrx2[i] - 1, arry2[i]) == '&')

{

health=health1-1;

break;

}

}

}

if(firelimit > 15)

{

bullettank2();

firelimit=0;

}

else

{

firelimit++;

}

}

void gamecollision()

{

}