**DOODLE UP MANIAC!!!**

****

**Session 2022 – 2026**

**Submitted by:**

Raveeha Mohsin 2022-CS-149

**Supervised by:**

Dr. Awais Hassan

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Table of Contents

[2 Short Description and Story Writing of Game 3](#_Toc130382736)

[3 Game Characters Description 3](#_Toc130382737)

[3.1 Player 3](#_Toc130382738)

[3.1.1 Maniac the Doodler 3](#_Toc130382739)

[3.2 Enemies 3](#_Toc130382740)

[3.2.1 TURDY- The King of Hate 3](#_Toc130382741)

[3.2.2 Dusty Reaper 4](#_Toc130382742)

[3.2.3 BEWARESTONEY 4](#_Toc130382743)

[4 Game Objects Description 4](#_Toc130382744)

[4.1.1 Gold Coins: 4](#_Toc130382745)

[4.1.2 Shooting Easter Eggs 4](#_Toc130382746)

[4.1.3 Steps 4](#_Toc130382747)

[5 Rules & Interactions 4](#_Toc130382748)

[6 Goal of the Game 4](#_Toc130382749)

[7 Wireframes of the Game 5](#_Toc130382750)

[7.1.1 Figure 1 Display 5](#_Toc130382751)

[7.1.2 Figure 2 menu 5](#_Toc130382752)

[7.1.3 Game 6](#_Toc130382753)

[7.1.4 option 2 7](#_Toc130382754)

[7.1.5 keys 7](#_Toc130382755)

[7.1.6 Instructions 7](#_Toc130382756)

[8 Data Structures 8](#_Toc130382757)

[9 Function Prototypes 8](#_Toc130382758)

[10 Complete Code 10](#_Toc130382759)

# Short Description and Story Writing of Game

Maniac lives in a world full of steps & coins. Maniac is greenish in color and this guy can shoot various projectiles from his mouth like Easter Eggs. The goal of the game is to help the character to get higher on the platforms and score as many points as possible. The hero is controlled by using the arrows on the keyboard. As it seems pretty easy to just step on these steps, get the coins and move but its not that easy as it seems because there will be someone who will always interrupt our little maniac in his journey.

Its the small bombs and stones which will continuously fall from upwards and try to kill maniac. But despite such hard journey maniac will not give up. He will try to face them by shooting eggs from his mouth. He is just in hunger of coins so he will get as much as he can to increase the score. Also to protect his health he will face his enemies by shooting them.

The only way you can fail is when you stop trying. Quitting is just like a disease which spreads slowly. Maniac tells us to neve give up on the fight until you win the belts. With such a thought in his mind, maniac is just determined to just step up in his life despite all the hurdles & challenges.

In this game once the player completes the two mazes, the game ends. This game attracts & entertains the players of all age.

# Game Characters Description

## Player

There is only one player in the game.

### Maniac the Doodler

Maniac is the main player in this game. He is yellow greenish and he is in mushroom shape. One must say that maniac is a daredevil. He is bold, venturesome, daring and brave. He never gives up and is always in search of some new gold coins. He is the hero of this game and is surely admired for his dauntless and valorous nature.

## Enemies

There are three enemies in the game.

### TURDY- The King of Hate

The turdy is one of the most aggressive in the game. He will always chase the little maniac and will always try to create problems for our main player. He is trying to decrease maniac health at every point in the game by continuously chasing him. He is the most violent and macho one.

### Dusty Reaper

Dusty reaper also creates hurdles for maniac and is the second enemy of the game but he tries to harm the maniac from the upper left side. He is also one of the evils. He is just giving a message of death to the maniac – the doodler.

### BEWARESTONEY

Here comes the third evil of our game who tries to kill maniac from the upper left of the game. Continuously dropping from the upside and trying to give him as much as he can just show his pugnacious nature.

# Game Objects Description

Following are the Objects in the game:

### Gold Coins:

A gold coin, also known as the stimulus for the maniac in the game. Maniac main aim is to gain as much gold coins as he can and to get his score increased by 10.The amount of gold maniac has in the game is used to measure his progress in game.

### Shooting Easter Eggs

When the three stone enemies (turdy, death reaper, bewarestoney) tries to kill the maniac he will shoot some eggs from his mouth to attack them back. These eggs will play as a role of fire in the game.

### Steps

Maniac – the doodler has to take the small steps to reach as high as he can. Gold coins will be available on these small steps.

# Rules & Interactions

Maniac score increases by 10 by gaining the coins. If the stone(enemies) touch the maniac, his health will be decreased by 10 and eventually if the health level reaches 0, the game will be over at that point.

# Goal of the Game

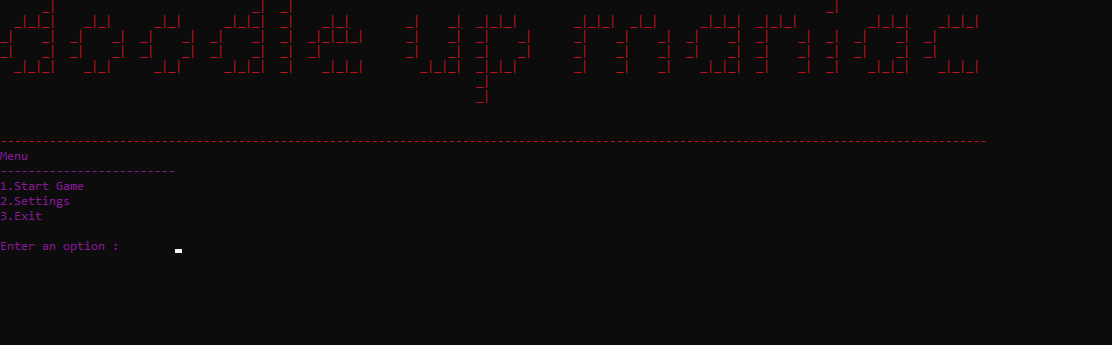
The goal of the game is to get as many coins that have been put on the steps as our player can while avoiding the enemies.

# Wireframes of the Game

### Figure 1 Display

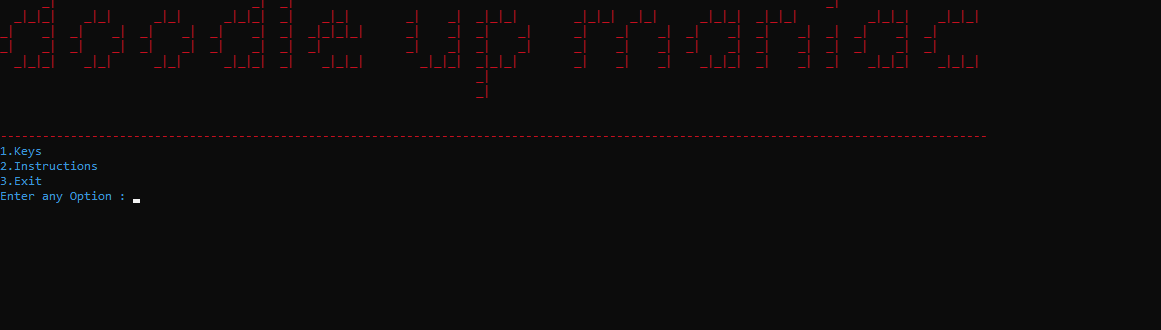


### Figure 2 menu

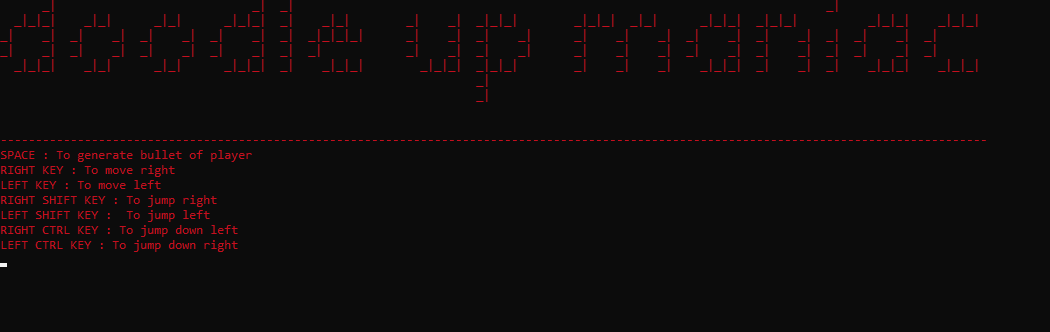


### Game

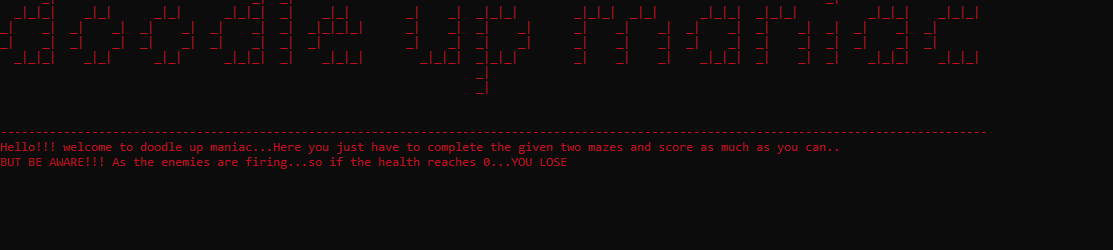
### option 2



### keys



### Instructions



# Data Structures

char player[3][3]= {{' ',face,' '},{' ',body,' '},{' ',foot,' '}};

char enemy[3][13]={{' ',' ',' ','\_','\_','\_','\_','\_','\_','\_',' ',' ',' '},

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

{'|',' ','|',' ',' ',' ',' ',' ',' ',' ','|',' ','|'}};

char enemy2[3][5] = {{p1, p1, p1,' ',' '},

{p1, p1, p1, '-', p2},

{'0', ' ', '0', ' ', ' '}};

# Function Prototypes

void moveBulletplayer();

void generateBulletplayer();

void printBulletplayer(int x, int y);

void eraseBulletplayer(int x, int y);

void removeBulletFromArrayplayer(int index);

void step(int x, int y);

void coin(int x, int y);

void displaypic();

void clearsystem();

void header();

int mainmenu();

void loading();int submenuopt();

void maze();

void playercharacter();

void eraseplayer();

void gotoxy(int x, int y);

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

void enemy3character();

void enemy2character();

void enemy1character();

void moveplayercharacterJumpright();

void moveplayercharacterJumpleft();

void moveplayercharacterright();

void moveplayercharacterleft();

void moveplayercharacterdownright();

void moveplayercharacterdownleft();

void eraseEnemy1();

void moveEnemy1();

void eraseenemy2();

void eraseenemy3();

void moveEnemy3();

void removeBulletFromArrayenemy1(int index);

void removeBulletFromArrayenemy2(int index);

void removeBulletFromArrayenemy3(int index);

void generateBulletenemy1();

void generateBulletenemy2();

void generateBulletenemy3();

void moveBulletEnemy1();

void moveBulletEnemy2();

void moveBulletEnemy3();

void eraseBulletenemy3(int x, int y);

void eraseBulletenemy2(int x, int y);

void printBulletenemy3(int x, int y);

void printBulletenemy2(int x, int y);

void eraseBulletenemy1(int x, int y);

void printBulletenemy1(int x, int y);

void gravityplayer();

void bulletCollisionWithEnemy();

void printhealth();

bool enterToU();

void win();

void lose();

void checkup();

# Complete Code

#include <iostream>

#include <windows.h>

#include <conio.h>

#include<fstream>

using namespace std;

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

char face = 153;

char body = 178;

char foot = 234;

char player[3][3]= {{' ',face,' '},{' ',body,' '},{' ',foot,' '}};

char enemy[3][13]={{' ',' ',' ','\_','\_','\_','\_','\_','\_','\_',' ',' ',' '},

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

{'|',' ','|',' ',' ',' ',' ',' ',' ',' ','|',' ','|'}};

char p1 = 223;

char p2 = 175;

char p3 = 168;

char enemy2[3][5] = {{p1, p1, p1,' ',' '},

{p1, p1, p1, '-', p2},

{'0', ' ', '0', ' ', ' '}};

char face4 = 148;

char foot4 = 202;

char body3 = 219;

char enemy7[3]= {' ',face4,' '};

char enemy8[3]= {' ',body3,' '};

char enemy9[3]= {' ',foot4,' '};

string enemyDirection1 = "Left";

string enemyDirection2 = "Up";

string enemyDirection3 = "Down";

int playerX = 62 ;

int playerY = 45 ;

int enemy1X =37 ;

int enemy1Y = 3 ;

int enemy2X = 1;

int enemy2Y =20 ;

int enemy3X = 79 ;

int enemy3Y = 27 ;

int timer = 0;

int health = 30;

int bulletXplayer[200];

int bulletYplayer[200];

int bulletCountplayer = 0;

void moveBulletplayer();

void generateBulletplayer();

void printBulletplayer(int x, int y);

void eraseBulletplayer(int x, int y);

void removeBulletFromArrayplayer(int index);

void step(int x, int y);

void coin(int x, int y);

void displaypic();

void clearsystem();

void header();

int mainmenu();

void loading();

int submenuopt();

void maze();

void playercharacter();

void eraseplayer();

void gotoxy(int x, int y);

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

void enemy3character();

void enemy2character();

void enemy1character();

void moveplayercharacterJumpright();

void moveplayercharacterJumpleft();

void moveplayercharacterright();

void moveplayercharacterleft();

void moveplayercharacterdownright();

void moveplayercharacterdownleft();

void eraseEnemy1();

void moveEnemy1();

void eraseenemy2();

void eraseenemy3();

void moveEnemy3();

void removeBulletFromArrayenemy1(int index);

void removeBulletFromArrayenemy2(int index);

void removeBulletFromArrayenemy3(int index);

int bulletXenemy1[10000];

int bulletYenemy1[10000];

int bulletXenemy2[10000];

int bulletYenemy2[10000];

int bulletXenemy3[10000];

int bulletYenemy3[10000];

int bulletCountenemy1 = 0;

int bulletCountenemy2 = 0;

int bulletCountenemy3 = 0;

void generateBulletenemy1();

void generateBulletenemy2();

void generateBulletenemy3();

void moveBulletEnemy1();

void moveBulletEnemy2();

void moveBulletEnemy3();

void eraseBulletenemy3(int x, int y);

void eraseBulletenemy2(int x, int y);

void printBulletenemy3(int x, int y);

void printBulletenemy2(int x, int y);

void eraseBulletenemy1(int x, int y);

void printBulletenemy1(int x, int y);

void gravityplayer();

void bulletCollisionWithEnemy();

void printhealth();

bool enterToU();

void win();

void lose();

void checkup();

int score =0;

bool scorescheck=false;

void scorecal()

{if(scorescheck==true)score+=10;}

void printScore(int score);

main()

{

bool flag;

// int c\_count=0;

displaypic();

loading();

clearsystem();

header();

int opt = mainmenu();

if (opt == 1)

{

for (int idx = 0; idx < 2;idx ++)

{

flag = true;

clearsystem();

Sleep(100);

// if(c\_count==0)

maze();

// c\_count++;

playerX = 62 ;

playerY = 45 ;

playercharacter();

enemy1character();

enemy2character();

enemy3character();

while(flag)

{

scorecal();

printScore(score);

printhealth();

checkup();

generateBulletenemy1();

generateBulletenemy2();

generateBulletenemy3();

if (GetAsyncKeyState(VK\_LSHIFT))

{

moveplayercharacterJumpright();

}

if (GetAsyncKeyState(VK\_RSHIFT))

{

moveplayercharacterJumpleft();

}

if (GetAsyncKeyState(VK\_RIGHT))

{

moveplayercharacterright();

}

if (GetAsyncKeyState(VK\_LEFT))

{

moveplayercharacterleft();

}

if (GetAsyncKeyState(VK\_RCONTROL))

{

moveplayercharacterdownright();

}

if (GetAsyncKeyState(VK\_LCONTROL))

{

moveplayercharacterdownleft();

}

if (GetAsyncKeyState(VK\_SPACE))

{

generateBulletplayer();

}

gravityplayer();

if (timer == 3)

{

moveEnemy1();

moveEnemy3();

timer = 0;

}

moveBulletEnemy1();

moveBulletEnemy2();

moveBulletEnemy3();

moveBulletplayer();

bulletCollisionWithEnemy();

timer ++;

flag = enterToU();

Sleep(100);

if(health <= 0)

{

system("cls");

lose();

flag = false;

idx = 3;

}

}

system("cls");

}

system("cls");

win();

}

if (opt ==2)

{

system("cls");

header();

int option = submenuopt();

if (option == 1)

{

system("cls");

header();

cout << "SPACE : To generate bullet of player " << endl;

cout << "RIGHT KEY : To move right" << endl;

cout << "LEFT KEY : To move left" << endl;

cout << "RIGHT SHIFT KEY : To jump right" << endl;

cout << "LEFT SHIFT KEY : To jump left" << endl;

cout << "RIGHT CTRL KEY : To jump down left" << endl;

cout << "LEFT CTRL KEY : To jump down right" << endl;

getch();

}

if (option == 2)

{

system("cls");

header();

cout << "Hello!!! welcome to doodle up maniac...Here you just have to complete the given two mazes and score as much as you can.." << endl;

cout << "BUT BE AWARE!!! As the enemies are firing...so if the health reaches 0...YOU LOSE" << endl;

getch();

}

if (option == 3)

{

return 0;

}

}

if (opt == 3)

{

return 0;

}

}

void displaypic() // print logo

{

SetConsoleTextAttribute(acolor,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<<"%\*===:%#::::::::-%\*%+------------------------------------------=%#--\*%\*=+%= " <<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 loading() // print loading

{

SetConsoleTextAttribute(acolor,10);

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

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

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

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

}

void header() // print header

{

SetConsoleTextAttribute(acolor,4);

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

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

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

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

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

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

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

cout << endl;

cout << endl;

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

}

void clearsystem()

{

SetConsoleTextAttribute(acolor,3);

cout << "Press any key to continue!!" << endl;

getch();

system("cls");

}

int mainmenu() // menu

{

SetConsoleTextAttribute(acolor,5);

int option;

cout << "Menu " <<endl;

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

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

cout << "2.Settings " <<endl;

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

cout << "Enter an option : " ;

cin >> option;

return option;

}

int submenuopt() // menu

{

SetConsoleTextAttribute(acolor,3);

int option;

cout << "1.Keys" << endl;

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

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

cout << "Enter any Option : ";

cin >> option;

return option;

}

void maze() // print maze

{

SetConsoleTextAttribute(acolor,6);

// cout << "&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&" << endl;

// cout << "& &" << endl;

// cout << "&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&" << endl;

// cout << "& &" << endl;

// cout << "& &" << endl;

// cout << "& &" << endl;

// cout << "& @@@ &" << endl;

// cout << "& u &" << 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;

// cout << "&\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*&" << endl;

// cout << "&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&" << endl;

string line;

string display[54];

fstream file;

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

int idx = 0;

while (!file.eof())

{

getline(file,line);

display[idx] = line;

idx++;

}

file.close();

for (int rows = 0; rows < 54; rows++)

{

cout << display[rows];

cout << endl;

}

}

void playercharacter() // print player

{

SetConsoleTextAttribute(acolor,10);

for(int i=0;i<3;i++)

{

gotoxy(playerX,playerY+i);

for(int j=0;j<3;j++)

{

cout<< player[i][j];

}

cout<<endl;

}

}

void eraseplayer() // remove player

{

for(int i=0;i<3;i++)

{

gotoxy(playerX,playerY+i);

for(int j=0;j<3;j++)

{

cout<< " ";

}

cout<<endl;

}

}

void enemy1character() // print ene 1

{

SetConsoleTextAttribute(acolor,4);

for(int i=0;i<3;i++)

{

gotoxy(enemy1X,enemy1Y+i);

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

{

cout<< enemy[i][j];

}

cout<<endl;

}

}

void eraseEnemy1() // erase ene 1

{

for(int i=0;i<3;i++)

{

gotoxy(enemy1X,enemy1Y+i);

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

{

cout<<" ";

}

cout<<endl;

}

}

void moveEnemy1() // move ene 1

{

if (enemyDirection1 == "Left")

{

char next = getCharAtxy(enemy1X-1, enemy1Y);

if (next == ' ')

{

eraseEnemy1();

enemy1X--;

enemy1character();

}

if (next == '&')

{

enemyDirection1 = "Right";

}

}

if (enemyDirection1 == "Right")

{

char next = getCharAtxy(enemy1X+13, enemy1Y);

if (next == ' ')

{

eraseEnemy1();

enemy1X++;

enemy1character();

}

if (next == '&')

{

enemyDirection1 = "Left";

}

}

}

void enemy2character() //print ene 2

{

SetConsoleTextAttribute(acolor,4);

for(int i=0;i<3;i++)

{

gotoxy(enemy2X,enemy2Y+i);

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

{

cout<<enemy2[i][j];

}

cout<<endl;

}

}

void moveEnemy3() // moving ene 3

{

if (enemyDirection3 == "Up")

{

char next = getCharAtxy(enemy3X, enemy3Y - 1);

if (next == ' ')

{

eraseenemy3();

enemy3Y--;

enemy3character();

}

if (next == '$')

{

enemyDirection3 = "Down";

}

}

if (enemyDirection3 == "Down")

{

char next = getCharAtxy(enemy3X, enemy3Y + 4);

if (next == ' ')

{

eraseenemy3();

enemy3Y++;

enemy3character();

}

if (next == '$')

{

enemyDirection3 = "Up";

}

}

}

void eraseenemy2() // erase en 2

{

for(int i=0;i<3;i++)

{

gotoxy(enemy2X,enemy2Y+i);

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

{

cout<<" ";

}

cout<<endl;

}

}

void enemy3character() // printing enemy

{

SetConsoleTextAttribute(acolor,4);

gotoxy(enemy3X,enemy3Y);

for(int idx = 0;idx < 3;idx ++)

{

cout << enemy7[idx];

}

gotoxy(enemy3X,enemy3Y+1);

for(int idx = 0;idx < 3;idx ++)

{

cout << enemy8[idx];

}

gotoxy(enemy3X,enemy3Y+2);

for(int idx = 0;idx < 3;idx ++)

{

cout << enemy9[idx];

}

}

void eraseenemy3() // erasing enemy

{

gotoxy(enemy3X,enemy3Y);

for(int idx = 0;idx < 3;idx ++)

{

cout << " ";

}

gotoxy(enemy3X,enemy3Y+1);

for(int idx = 0;idx < 3;idx ++)

{

cout << " ";

}

gotoxy(enemy3X,enemy3Y+2);

for(int idx = 0;idx < 3;idx ++)

{

cout << " ";

}

}

void moveplayercharacterJumpright() // move jump right

{

char jumpright = getCharAtxy(playerX - 12, playerY - 2);

if (jumpright == '\*' || jumpright == '@')

{

eraseplayer();

playerY = playerY - 5;

playerX = playerX - 12;

playercharacter();

}

if (jumpright == '<' || jumpright == '>' || jumpright == 'v')

{

health--;

}

}

void moveplayercharacterJumpleft() // move jump left

{

char jumpleft = getCharAtxy(playerX + 12, playerY - 2);

if (jumpleft == '\*' || jumpleft == '@')

{

eraseplayer();

playerY = playerY - 5;

playerX = playerX + 12;

playercharacter();

}

if (jumpleft == '<' || jumpleft == '>' || jumpleft == 'v')

{

health--;

}

}

void moveplayercharacterdownright() // move down right

{

char downright = getCharAtxy(playerX + 12, playerY + 8 );

if(downright != '\*')

{

eraseplayer();

playerX = 62;

playerY = 45;

playercharacter();

}

else

{

eraseplayer();

playerY= playerY + 5 ;

playerX= playerX + 12;

playercharacter();

}

}

void moveplayercharacterdownleft() // move down left

{

char downleft = getCharAtxy(playerX- 12 , playerY + 8);

if(downleft != '\*')

{

eraseplayer();

playerX = 62;

playerY = 45;

playercharacter();

}

else

{

eraseplayer();

playerY= playerY + 5;

playerX= playerX - 12;

playercharacter();

}

}

void moveplayercharacterleft() // move left

{

char left = getCharAtxy(playerX-1, playerY);

if (left == ' ' )

{ scorescheck=false;

eraseplayer();

playerX = playerX - 1;

playercharacter();

}

else if(left == '@')

{

scorescheck=true;

eraseplayer();

playerX = playerX - 1;

playercharacter();

}

else if (left == '<' ||left == '>'||left == 'v')

{

health--;

}

}

void moveplayercharacterright() // move right

{

char right = getCharAtxy(playerX+4 , playerY);

if (right == ' ')

{ scorescheck=false;

eraseplayer();

playerX = playerX + 1;

playercharacter();

}

else if(right == '@')

{

scorescheck=true;

eraseplayer();

playerX = playerX + 1;

playercharacter();

}

}

void checkright() //health decrease check

{ char upr = getCharAtxy(playerX+3 , playerY-4);

char upl = getCharAtxy(playerX+3 , playerY-5);

char upl2 = getCharAtxy(playerX+3 , playerY-6);

char upl3 = getCharAtxy(playerX+3 , playerY-7);

if (upr=='<'||upl=='<'||upl2=='<'||upl3=='<'){health=health-1;}

}

void checkup() //health decrease check

{ char a = getCharAtxy(playerX+3 , playerY-4);

char b = getCharAtxy(playerX+3 , playerY-5);

char c = getCharAtxy(playerX+3 , playerY-6);

char d = getCharAtxy(playerX+3 , playerY-7);

char a2 = getCharAtxy(playerX-3 , playerY-4);

char b2 = getCharAtxy(playerX-3 , playerY-5);

char c2 = getCharAtxy(playerX-3 , playerY-6);

char d2 = getCharAtxy(playerX-3 , playerY-7);

if (a=='v'||b=='v'||c=='v'||d=='v'){health=health-1;}

if (a2=='v'||b2=='v'||c2=='v'||d2=='v'){health=health-1;}

}

void generateBulletplayer() // generating bullet

{

bulletXplayer[bulletCountplayer] = playerX ;

bulletYplayer[bulletCountplayer] = playerY - 1;

gotoxy(playerX, playerY-1);

cout << ".";

bulletCountplayer++;

}

void moveBulletplayer() // moving the bullet of player

{

for (int idx= 0; idx < bulletCountplayer; idx++)

{

char next = getCharAtxy(bulletXplayer[idx] + 1, bulletYplayer[idx] - 1);

if (next == '&')

{

eraseBulletplayer(bulletXplayer[idx], bulletYplayer[idx]);

removeBulletFromArrayplayer(idx);

}

else if (next == ' ')

{

eraseBulletplayer(bulletXplayer[idx], bulletYplayer[idx]);

bulletYplayer[idx] = bulletYplayer[idx] - 1;

printBulletplayer(bulletXplayer[idx], bulletYplayer[idx]);

}

else if (next =='\*')

{

eraseBulletplayer(bulletXplayer[idx], bulletYplayer[idx]);

bulletYplayer[idx] = bulletYplayer[idx] -3 ;

printBulletplayer(bulletXplayer[idx], bulletYplayer[idx]);

}

}

}

void printBulletplayer(int x, int y) // printing the bullet of player

{

SetConsoleTextAttribute(acolor,15);

gotoxy(x, y);

cout << ".";

}

void eraseBulletplayer(int x, int y) // erase bullet player

{

gotoxy(x, y);

cout << " ";

}

void removeBulletFromArrayplayer(int index) // bullet of player

{

for (int x = index; x < bulletCountplayer - 1; x++)

{

bulletXplayer[x] = bulletXplayer[x + 1];

bulletYplayer[x] = bulletYplayer[x + 1];

}

bulletCountplayer--;

}

void printScore(int score) // score card

{

SetConsoleTextAttribute(acolor,11);

gotoxy(120, 15);

cout << "SCORE CARD!!!" << endl;

gotoxy(120, 20);

cout << "SCORE: " << score;

}

void printhealth() // score of health

{

SetConsoleTextAttribute(acolor,11);

gotoxy(120, 25);

cout << "HEALTH CARD!!!" << endl;

gotoxy(120, 30);

cout << "HEALTH: " << health;

}

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 generateBulletenemy2() // generating bullet of ene 2

{

SetConsoleTextAttribute(acolor,5);

bulletXenemy2[bulletCountenemy2]= enemy2X +3;

bulletYenemy2[bulletCountenemy2]= enemy2Y;

gotoxy(enemy2X+3,enemy2Y);

cout << "<";

bulletCountenemy2++;

}

void moveBulletEnemy2() //moving bullet

{int count=0;

for (int x = 0; x < bulletCountenemy2; x++)

{

char next = getCharAtxy(bulletXenemy2[x] + 1, bulletYenemy2[x]);

count++;

if (next != ' ' && next!='<')

{eraseBulletenemy2(bulletXenemy2[x], bulletYenemy2[x]);

removeBulletFromArrayenemy2(x);

}

else

{

eraseBulletenemy2(bulletXenemy2[x], bulletYenemy2[x]);

bulletXenemy2[x] = bulletXenemy2[x] + 1;

printBulletenemy2(bulletXenemy2[x], bulletYenemy2[x]);

}

void removeBulletFromArrayenemy2(int index) // removing from array

{

for (int x = index; x < bulletCountenemy2 - 1; x++)

{

bulletXenemy2[x] = bulletXenemy2[x + 1];

bulletYenemy2[x] = bulletYenemy2[x + 1];

}

bulletCountenemy2--;

}

void printBulletenemy2(int x, int y) // printing the bullet of enemy 2

{

SetConsoleTextAttribute(acolor,5);

gotoxy(x, y);

cout << ">";

}

void eraseBulletenemy2(int x, int y) // erasing bullet

{

gotoxy(x, y);

cout << " ";

}

void generateBulletenemy3() // generatig bullet

{

SetConsoleTextAttribute(acolor,5);

bulletXenemy3[bulletCountenemy3]= enemy3X -1;

bulletYenemy3[bulletCountenemy3]= enemy3Y;

gotoxy(enemy3X-1,enemy3Y);

cout << "<";

bulletCountenemy3++;

}

void moveBulletEnemy3() // moving bullet

{int count=0;

for (int x = 0; x < bulletCountenemy3; x++)

{

char next = getCharAtxy(bulletXenemy3[x] - 1, bulletYenemy3[x]);

count++;

if (next != ' ' && next!='<')

{eraseBulletenemy3(bulletXenemy3[x], bulletYenemy3[x]);

removeBulletFromArrayenemy3(x);

}

else

{

eraseBulletenemy3(bulletXenemy3[x], bulletYenemy3[x]);

bulletXenemy3[x] = bulletXenemy3[x] - 1;

printBulletenemy3(bulletXenemy3[x], bulletYenemy3[x]);

}

}

}

void removeBulletFromArrayenemy3(int index) // remove from given array

{

for (int x = index; x < bulletCountenemy3 - 1; x++)

{

bulletXenemy3[x] = bulletXenemy3[x + 1];

bulletYenemy3[x] = bulletYenemy3[x + 1];

}

bulletCountenemy3--;

}

void printBulletenemy3(int x, int y) // print bullet

{

SetConsoleTextAttribute(acolor,5);

gotoxy(x, y);

cout << "<";

}

void eraseBulletenemy3(int x, int y) // remove bullet

{

gotoxy(x, y);

cout << " ";

}

void gravityplayer() // player down

{

bool flag = true;

char next;

char next1;

while(flag)

{

next = getCharAtxy(playerX, playerY+3);

next1 = getCharAtxy(playerX+2, playerY+3);

if(next == ' ' && next1 == ' ')

{

eraseplayer();

playerY = playerY + 1;

playercharacter();

}

else

{

flag = false;

}

}

}

void generateBulletenemy1() // generating bullet

{

SetConsoleTextAttribute(acolor,5);

bulletXenemy1[bulletCountenemy1]= enemy1X;

bulletYenemy1[bulletCountenemy1]= enemy1Y+3;

gotoxy(enemy1X,enemy1Y+3);

cout << "v";

bulletCountenemy1++;

}

void moveBulletEnemy1() // moving bullet

{

int count=0;

for (int x = 0; x < bulletCountenemy1; x++)

{

char next = getCharAtxy(bulletXenemy1[x] , bulletYenemy1[x]+3);

count++;

if (next != ' ' && next!='<' )

{

eraseBulletenemy1(bulletXenemy1[x], bulletYenemy1[x]);

removeBulletFromArrayenemy1(x);

}

else if(next == 'u')

{

eraseBulletenemy1(bulletXenemy1[x], bulletYenemy1[x]);

removeBulletFromArrayenemy1(x);

}

else

{

eraseBulletenemy1(bulletXenemy1[x], bulletYenemy1[x]);

bulletYenemy1[x] = bulletYenemy1[x] + 3;

printBulletenemy1(bulletXenemy1[x], bulletYenemy1[x]);

}

}

}

void removeBulletFromArrayenemy1(int index) // remove from array

{

for (int x = index; x < bulletCountenemy1 - 1; x++)

{

bulletXenemy1[x] = bulletXenemy1[x + 1];

bulletYenemy1[x] = bulletYenemy1[x + 1];

}

bulletCountenemy1--;

}

void printBulletenemy1(int x, int y) // print the bullet of enemy 1

{

SetConsoleTextAttribute(acolor,5);

gotoxy(x, y);

cout << "v";

}

void eraseBulletenemy1(int x, int y) // erase bullet

{

gotoxy(x, y);

cout << " ";

}

void bulletCollisionWithEnemy() // collision with enemy

{

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

{

if (bulletXplayer[x] + 1 == enemy1X

&& (bulletYplayer[x] == enemy1Y || bulletYplayer[x] == enemy1Y + 1 || bulletYplayer[x] == enemy1Y + 2 || bulletYplayer[x] == enemy1Y + 3))

{

scorescheck=true;

eraseBulletplayer(bulletXplayer[x], bulletYplayer[x]);

removeBulletFromArrayplayer(x);

}

}

}

bool enterToU() // when enter in u

{

char left = getCharAtxy(playerX-1, playerY + 2);

if (left == 'u' )

{

return false;

}

return true;

}

void win() // to display when win

{

SetConsoleTextAttribute(acolor,10);

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

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

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

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

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

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

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

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

}

void lose() //to display when lose

{

SetConsoleTextAttribute(acolor,4);

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

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

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

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

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

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

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

}