**BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN**

**A colorful logo with a white background

Description automatically generated**

**Department of CSE Certified that this is a bonafide record of the work done by:**

**G.Keerti – 23WH1A05I7**

**P.Praneetha – 23WH1A05F2**

**Komma Tejasree - 23WH1A05I1**

**Sania – 23WH1A05G0**

**Of Class CSE C of Year 1 of Semester 1 in PPS Laboratory**

**DATE: SIGNATURE:**

#### PROBLEM STATEMENT:-

* As we see that in todays world people miss ancient mobile games of which one we have chosen is snake game. So, we built a snake game which will be robust and fast.

This game called “Simple snake game” typically involve the player controlling a line or snake. The game involves the snake eating items which make it longer. The food is indicated by letter ‘F’ which will be displayed on the screen at random positions.

**Task:-**

We need to write a c program that will:

* Allow the user to play this game in a one player mode.
* Increase the snake size as the user collects the food indicated by ‘F’.
* Display the score during the game.
* Display the time after which user stopped the game.

Press ‘A’ 🡪 Snake moves left

Press ‘S’ 🡪 Snake moves down

Press ‘W’ 🡪 Snake moves up

Press ‘D’ 🡪 Snake moves right

Press ‘X’ 🡪 Stop the game

A computer screen shot of a code

Description automatically generated

**SOURCE CODE**

#include <stdio.h>

#include <conio.h>

#include <windows.h>

int width = 20, height = 20;

int x, y, fruitX, fruitY, score, gameover;

int tailX[100], tailY[100];

int nTail;

enum eDirection { STOP = 0, LEFT, RIGHT, UP, DOWN };

enum eDirection dir;

void Setup() {

gameover = 0;

dir = STOP;

x = width / 2;

y = height / 2;

fruitX = rand() % width;

fruitY = rand() % height;

score = 0;

}

void Draw() {

system("cls");

for (int i = 0; i < width + 2; i++)

printf("#");

printf("\n");

for (int i = 0; i < height; i++) {

for (int j = 0; j < width; j++) {

if (j == 0)

printf("#");

if (i == y && j == x)

printf("O");

else if (i == fruitY && j == fruitX)

printf("F");

else {

int printTail = 0;

for (int k = 0; k < nTail; k++) {

if (tailX[k] == j && tailY[k] == i) {

printf("o");

printTail = 1;

}

}

if (printTail == 0)

printf(" ");

}

if (j == width - 1)

printf("#");

}

printf("\n");

}

for (int i = 0; i < width + 2; i++)

printf("#");

printf("\n");

printf("Score: %d\n", score);

}

void Input() {

if (\_kbhit()) {

switch (\_getch()) {

case 'a':

dir = LEFT;

break;

case 'd':

dir = RIGHT;

break;

case 'w':

dir = UP;

break;

case 's':

dir = DOWN;

break;

case 'x':

gameover = 1;

break;

}

}

}

void Logic() {

int prevX = tailX[0];

int prevY = tailY[0];

int prev2X, prev2Y;

tailX[0] = x;

tailY[0] = y;

for (int i = 1; i < nTail; i++) {

prev2X = tailX[i];

prev2Y = tailY[i];

tailX[i] = prevX;

tailY[i] = prevY;

prevX = prev2X;

prevY = prev2Y;

}

switch (dir) {

case LEFT:

x--;

break;

case RIGHT:

x++;

break;

case UP:

y--;

break;

case DOWN:

y++;

break;

default:

break;

}

if (x >= width) x = 0; else if (x < 0) x = width - 1;

if (y >= height) y = 0; else if (y < 0) y = height - 1;

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

if (tailX[i] == x && tailY[i] == y)

gameover = 1;

if (x == fruitX && y == fruitY) {

score += 10;

fruitX = rand() % width;

fruitY = rand() % height;

nTail++;

}

}

int main() {

Setup();

while (!gameover) {

Draw();

Input();

Logic();

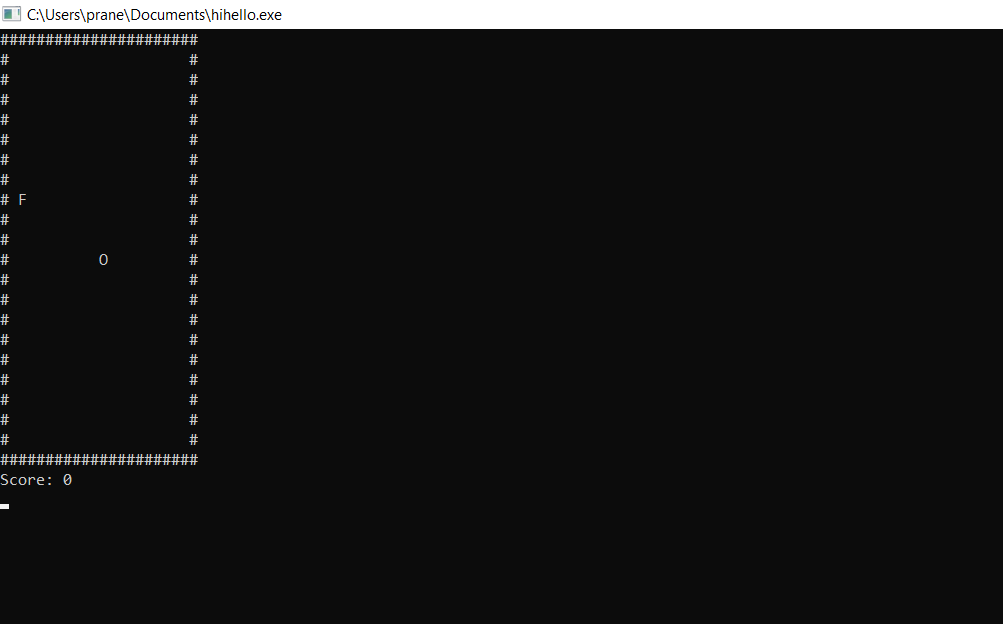
Sleep(100); // Adjust speed here (milliseconds)

}

return 0;

}

**OUTPUT:**



A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated