Snake Game

+

PPS PROJECT

By Dhatrish Bali
RA2111001010003

BASICS OF THE GAME

The snake is represented with a **0**(zero) symbol.

The fruit is represented with an *****(asterisk) symbol.

The snake can move in any direction according to the user with the help of the keyboard (**W**, **A**, **S**, **D** keys).

When the snake eats a fruit the score will increase by 10 points.

The fruit will generate automatically within the boundaries.

Whenever the snake will touch the boundry the game is over.

STEPS

There will be four user-defined functions.

Build a boundary within which the game will be played.

The fruits are generated randomly.

Then increase the score whenever the snake eats a fruit.

USER DEFINED FUNCTIONS

1

Draw(): This function creates the boundary in which the game will be played.

2

Setup(): This function will set the position of the fruit within the boundary.

3

Input(): This function will take the input from the keyboard.

4

Logic(): This function will set the movement of the snake.

```c
#include <conio.h> Untitled-2
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int i, j, height = 20, width = 20;
int gameover, score;
int x, y, fruitx, fruity, flag;

```

The header file and Variable used in this programe

```
####################
#     *            #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#        0         #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
####################
score = 0
press X to quit the game
```

```c
// Function to draw the boundaries
void draw()
{
    system("cls");
    for (i = 0; i < height; i++) {
        for (j = 0; j < width; j++) {
            if (i == 0 || i == width - 1
                || j == 0
                || j == height - 1) {
                printf("#");
            }
            else {
                if (i == x && j == y)
                    printf("0");
                else if (i == fruitx
                        && j == fruity)
                    printf("*");
                else
                    printf(" ");
            }
        }
        printf("\n");
    }

```

raw(): This function is responsible to build the boundary within which the game will be played

setup():nThisbfunction is used to write the code to generate the fruit within the boundary using **rand()** function.

- Using **rand()%20** because the size of the boundary is **length = 20** and **width = 20** so the fruit will generate within the boundary.

```
void setup()
{
    gameover = 0;

    // Stores height and width
    x = height / 2;
    y = width / 2;
label1:
    fruitx = rand() % 20;
    if (fruitx == 0)
        goto label1;
label2:
    fruity = rand() % 20;
    if (fruity == 0)
        goto label2;
    score = 0;
}
```

Input(): In this function, the programmer writes the code to take the input from the keyboard (W, A, S, D, X keys).

```
void input()
{
    if (kbhit()) {
        switch (getch()) {
        case 'a':
            flag = 1;
            break;
        case 's':
            flag = 2;
            break;
        case 'd':
            flag = 3;
            break;
        case 'w':
            flag = 4;
            break;
        case 'x':
            gameover = 1;
            break;
        }
    }
}
```

logic(): Here, write all the logic for this program like for the movement of the snake, for increasing the score, when the snake will touch the boundary the game will be over, to exit the game and the random generation of the fruit once the snake will eat the fruit.

```c
// Function for the logic behind
// each movement
void logic()
{
    sleep(0.01);
    switch (flag) {
    case 1:
        y--;
        break;
    case 2:
        x++;
        break;
    case 3:
        y++;
        break;
    case 4:
        x--;
        break;
    default:
        break;
    }

    // If the game is over
    if (x < 0 || x > height
        || y < 0 || y > width)
        gameover = 1;

    // If snake reaches the fruit
    // then update the score
    if (x == fruitx && y == fruity) {
    label3:
        fruitx = rand() % 20;
        if (fruitx == 0)
            goto label3;

    // After eating the above fruit
    // generate new fruit
    label4:
        fruity = rand() % 20;
        if (fruity == 0)
            goto label4;
        score += 10;
    }
}
```

sleep(): This function in C is a function that delays the program execution for the given number of seconds. In this code sleep() is used to slow down the movement of the snake so it will be easy for the user to play.

```c
// Driver Code
void main()
{
    int m, n;

    // Generate boundary
    setup();

    // Until the game is over
    while (!gameover) {

        // Function Call
        draw();
        input();
        logic();
    }
}
```

THE FINAL PRODUCT