

```

#include <iostream>
#include <conio.h>
#include <windows.h>
using namespace std;
bool gameOver;
const int width = 20;
const int height = 20;
int x, y, fruitX, fruitY, score;
int tailX[100], tailY[100];
int nTail;
enum eDirecton { STOP = 0, LEFT, RIGHT, UP, DOWN};
eDirecton dir;
void Setup()
{
    gameOver = false;
    dir = STOP;
    x = width / 2;
    y = height / 2;
    fruitX = rand() % width;
    fruitY = rand() % height;
    score = 0;
}
void Draw()
{
    system("cls"); //system("clear");
    for (int i = 0; i < width+2; i++)
        cout << "#";
    cout << endl;

    for (int i = 0; i < height; i++)
    {
        for (int j = 0; j < width; j++)
        {
            if (j == 0)
                cout << "#";
            if (i == y && j == x)
                cout << "O";
            else if (i == fruitY && j == fruitX)
                cout << "F";
            else
            {
                bool print = false;
                for (int k = 0; k < nTail; k++)
                {
                    if (tailX[k] == j && tailY[k] == i)
                    {
                        cout << "o";
                        print = true;
                    }
                }
                if (!print)
                    cout << " ";
            }

            if (j == width - 1)
                cout << "#";
        }
        cout << endl;
    }

    for (int i = 0; i < width+2; i++)
        cout << "#";
    cout << endl;
}

```

```

        cout << "Score:" << score << endl;
    }
    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 = true;
                    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 || y > height || y < 0)
        //    gameOver = true;
        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 = true;

    if (x == fruitX && y == fruitY)
    {
        score += 10;
        fruitX = rand() % width;
        fruitY = rand() % height;
        nTail++;
    }
}

int main()
{
    Setup();
    while (!gameOver)
    {
        Draw();
        Input();
        Logic();
        Sleep(10); //sleep(10);
    }
    return 0;
}
```