#include <iostream> // осноная библиотека

#include <conio.h>// библиотека отслеживания нажатий пользователя

using namespace std;

bool gameOver;// переменная обозначающая конец игры

const int width = 20;//переменная ширины поля где находится змейка

const int height = 20; // переменная высоты полля где находится змея

int x, y, fruithX, fruithY, score;// координаты расположения змеи ,координаты расположения фруктов ,переменная score для записи счета игры

int tailX[100], tailY[100];//масив переменой добовляющий элемент в хвост змеи по координатм

int nTail;

enum eDirection { STOP = 0, LEFT, RIGHT, UP, DOWN };//передвижение змейки ,змейка в начале игры не двигается ,

eDirection dir;//переменная на основе движения змейки dir инициализация

void Setup() {

gameOver = false;

dir = STOP; //нстанта инициализирущая при старте игры не подвижность змейки

x = width / 2;// функция при которой змейка при старте игры стоит на ровно в центр ширина и высоа по осям х у / 2

y = height / 2;

fruithX = rand() % width;// рандомное появление фруктов на карте по оси х

fruithY = rand() % height;// рандомное определение фруктов на карте по оси У

score = 0;// общий счет равен 0 при старте

}// функция при старте игры определяющая все параметры

void Draw() {

system("cls");//очищает терминал при вызове функции Draw

for (int i = 0; i < width + 1; i++)

cout << "#";

cout << endl;// верхняя граница карты (цикл)

for (int i = 0; i < height; i++) {//цикл for описывает карту по высоте

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

if (j == 0 || j == width - 1)//заполнение последней ячейки решеткой

cout << "#";//вывод решетки карты в терминал обозначение границы

if (i == y && j == x)//вывод змеи в параметры карты

cout << "0";// обозначение змеи в терминале

else if (i == fruithY && j == fruithX)

cout << "F"; // вывод и обозначение фрукта в терминале

else {

bool print = false;

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

if (tailX[k] == j && tailY[k] == i) { //еслт тайл х с индексом к и тайл у с индексом съедание фрукта будет добавлять новый элемент к змее

print = true;

cout << "0";

}

}

if(!print)//если принт равен фолс то ставится пробел

cout << " ";//обозначение свободного поля знаком пустота

}

}

cout << endl;//после прохождения ряда перекидывание на новую строку

}

for (int i = 0; i < width + 1; i++)

cout << "#";

cout << endl;//нижняя граница карты (цикл )

cout << "Score:" << score << endl;//вывод счета в консоль

} //функция которая рисует карту и змею

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;

}

}

}//отслеживание нажатий от пользователя

void logic() {

int prevX = tailX[0];

int prevY = tailY[0];

int prev2X, prev2Y;

tailX[0] = x; // непосредственное добавление элементов к змееке по координатом первого

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;//добавление и постоянная перезапись массивного элемента в хвосте по координатам

}

switch (dir)

{

case LEFT://управление змейкой влево вправо вверх вниз

x--;

break;

case UP:

y--;

break;

case DOWN:

y++;

break;

case RIGHT:

x++;

break;

}

//if (x > width || x<0 || y >height || y < 0) //проверка на врезание в стену логика проверки

//gameOver = true;

if (x >= width)//если координата х больше или равна 0 или иначе наша кроррдината равна обратному числу для выхода змейки с обратной стороны карты

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;// цикл и его проверка хвоста по х и у сходятся с координатом головы то конец игры

}

if (x == fruithX && y == fruithY) {

score += 10;//количесво баллов прибовлеямых ха съеденный фрукт

fruithX = rand() % width;

fruithY = rand() % height;

nTail++;//хвост будет становиться на одну единицу больше при поедании

}

} //функция логики

int main() {

Setup();

while (!gameOver) { //ЦИКЛ ПРОВЕРЕЮЩИЙ ПЕРЕМЕННУЮ КОНЕЦ ИГРЫ

Draw();

input();//постоянная прорисовка карты

logic();

}

return 0;

}//ОСНОВНОЙ ЦИКЛ