ДЕПАРТАМЕНТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ТОМСКОЙ ОБЛАСТИ

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

Специальность 09.02.04 «Информационные системы (по отраслям)»

Отчет по лабораторной работе № 1

|  |  |  |
| --- | --- | --- |
| Студен | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Н.М.Макеев |
| Руководитель  «\_\_»\_\_\_\_\_\_\_\_ 201\_ г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | В.С.Павлов |

Задание: разработать игру «змейку».

Код:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace Snake

{

class Program

{

public static int width = 0, height = 0;

public static int headX, headY, fruitX, fruitY, size = 1;

public static ConsoleKeyInfo f;

public static string move = "";

public static bool eat = false;

public static int[] masX = new int[size];

public static int[] masY = new int[size];

public static bool game = true;

public static int score = 0, speed = 0;

static void Main(string[] args)

{

string[,] map = new string[width, height];

string e = "Легко";

string m = "Средний";

string h = "Сложный";

int menu = 0;

bool start = false;

Console.WriteLine("Выберете сложность и нажмите Enter");

while(start == false)

{

if(menu == 0)

{

Console.ForegroundColor = ConsoleColor.Green;

Console.SetCursorPosition(0, 1);

Console.WriteLine(e);

Console.ForegroundColor = ConsoleColor.White;

Console.SetCursorPosition(0, 2);

Console.WriteLine(m);

Console.SetCursorPosition(0, 3);

Console.WriteLine(h);

}else if(menu == 1)

{

Console.SetCursorPosition(0, 1);

Console.WriteLine(e);

Console.ForegroundColor = ConsoleColor.Green;

Console.SetCursorPosition(0, 2);

Console.WriteLine(m);

Console.ForegroundColor = ConsoleColor.White;

Console.SetCursorPosition(0, 3);

Console.WriteLine(h);

}else if(menu == 2)

{

Console.ForegroundColor = ConsoleColor.White;

Console.SetCursorPosition(0, 1);

Console.WriteLine(e);

Console.SetCursorPosition(0, 2);

Console.WriteLine(m);

Console.ForegroundColor = ConsoleColor.Green;

Console.SetCursorPosition(0, 3);

Console.WriteLine(h);

Console.ForegroundColor = ConsoleColor.White;

}

if (Console.KeyAvailable == true)

{

f = Console.ReadKey();

if (f.Key == ConsoleKey.UpArrow)

if(menu != 0) { --menu; }

if (f.Key == ConsoleKey.DownArrow)

if(menu != 2) { ++menu; }

if (f.Key == ConsoleKey.Enter)

start = true;

}

}

Console.Clear();

if(menu == 0)

{

width = 10;

height = 10;

map = new string[width, height];

speed = 400;

}else if(menu == 1)

{

width = 15;

height = 15;

map = new string[width, height];

speed = 250;

Console.WriteLine(width + " " + height);

}else if(menu == 2)

{

width = 20;

height = 20;

map = new string[width, height];

speed = 100;

}

Random rnd = new Random();

headX = width/2; headY = height/2;

while (true)

{

fruitX = rnd.Next(1, width - 2); fruitY = rnd.Next(1, height - 2);

if (fruitX == headX && fruitY == headY || fruitX != headX && fruitY == headY || fruitX != headX && fruitY != headY)

{

break;

}

}

while (game == true)

{

Game.Draw(width, height, ref headX, ref headY, ref fruitX, ref fruitY, ref map);

Game.Move(width, height, ref headX, ref headY, move, ref map);

Game.EatGenerate(ref score, width, height, ref map, headX, headY, ref fruitX, ref fruitY, ref eat, masX, masY);

Game.Tail(ref masX, ref masY, ref map, ref eat, ref size, headX, headY);

Game.Lose(width, height, masX, masY, headX, headY, ref game, size);

Game.Win(width, height, map, ref game);

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

{

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

{

Console.SetCursorPosition(j, i);

Console.Write(map[i, j]);

}

Console.WriteLine();

}

Console.SetCursorPosition(0, height);

Console.WriteLine("Длина змейки: " + size);

if (Console.KeyAvailable == true)

{

f = Console.ReadKey();

if (f.Key == ConsoleKey.UpArrow)

if (move != "Down")

move = "Up";

if (f.Key == ConsoleKey.DownArrow)

if (move != "Up")

move = "Down";

if (f.Key == ConsoleKey.RightArrow)

if (move != "Left")

move = "Right";

if (f.Key == ConsoleKey.LeftArrow)

if (move != "Right")

move = "Left";

}

System.Threading.Thread.Sleep(speed);

}

}

}

class Game

{

public static void Tail(ref int[] masX, ref int[] masY, ref string[,] map, ref bool eat, ref int size, int headX, int headY)

{

if(eat == true)

{

++size;

Array.Resize<int>(ref masX, size);

Array.Resize<int>(ref masY, size);

eat = false;

}

map[masX[size - 1], masY[size - 1]] = " ";

for(int i = masX.Length - 1; i > 0; --i)

{

masX[i] = masX[i - 1];

masY[i] = masY[i - 1];

}

masX[0] = headX;

masY[0] = headY;

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

{

if(masX[i] == headX && masY[i] == headY)

{

map[masX[i], masY[i]] = "X";

}

map[masX[i], masY[i]] = "X";

}

eat = false;

}

public static void Move(int width, int height, ref int headX, ref int headY, string move, ref string[,] map)

{

{

map[headX, headY] = " ";

switch (move)

{

case "Up":

if (headX - 1 >= 0) headX -= 1;

break;

case "Down":

if (headX + 1 < width) headX += 1;

break;

case "Left":

if (headY - 1 >= 0) headY -= 1;

break;

case "Right":

if (headY + 1 < height) headY += 1;

break;

}

map[headX, headY] = "X";

}

}

public static void Draw(int width, int height, ref int headX, ref int headY, ref int fruitX, ref int fruitY, ref string[,] map)

{

Random rnd = new Random();

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

{

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

{

map[i, j] = " ";

if (i == 0 || j == 0 || i == width - 1 || j == height - 1)

{

map[i, j] = "=";

}

}

}

map[headX, headY] = "X";

map[fruitX, fruitY] = "\*";

}

public static void EatGenerate(ref int score, int width, int height, ref string[,] map, int headX, int headY, ref int fruitX, ref int fruitY, ref bool eat, int[] masX, int[] masY)

{

if (fruitX == headX && fruitY == headY)

{

++score;

eat = true;

Random rnd = new Random();

bool s = true;

while (s == true)

{

fruitX = rnd.Next(1, width - 2);

fruitY = rnd.Next(1, height - 2);

int p = 0;

if (fruitX != headX && fruitY == headY || fruitY != headY && fruitX == headX || fruitX != headX && fruitY != headY)

{

for (int i = 0; i < masX.Length; i++)

{

if (fruitX != masX[i] && fruitY == masY[i] || fruitY != masY[i] && fruitX == masX[i] || fruitX != masX[i] && fruitY != masY[i])

p++;

if (p == masX.Length)

s = false;

}

}

map[fruitX, fruitY] = "\*";

}

}

}

public static void Lose(int width, int height, int[] masX, int[] masY, int headX, int headY, ref bool game, int size)

{

if (headX == 0 || headX == width - 1 || headY == 0 || headY == height - 1)

{

game = false;

Console.WriteLine("You LOSE!!!");

Console.ReadKey();

}

for (int i = 1; i < size; i++)

{

if (masX[i] == headX && masY[i] == headY)

{

game = false;

Console.WriteLine("You LOSE!!!");

Console.ReadKey();

}

}

}

public static void Win(int width, int height, string[,] map, ref bool game)

{

int p = 0;

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

{

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

{

if (i != 0 || j != 0 || i != width - 1 || j != height - 1)

{

if (map[i, j] == "X")

{

p++;

}

}

}

}

if (p == map.Length - width \* 2 - height \* 2 + 4)

{

game = false;

Console.WriteLine("You WIN!!!");

}

}

}

}

Результат:

