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 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А. А. Гойман |
| Руководитель  «\_\_»\_\_\_\_\_\_\_\_ 201\_ г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | В. С. Павлов |

2020

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

using System.Net.Security;

namespace Змеюка

{

class Program

{

static public int height = 40;

static public int width = 80;

public ConsoleKeyInfo keypress = new ConsoleKeyInfo();

static public bool gameOver, isprinted, horizontal, vertical;

static public int[] TailX = new int[50];

static public int[] TailY = new int[50];

int step = 0;

int e = 0;

static public int foodX, foodY, headX, headY, score, w, h, nTail;

public enum Direction { Stop, Left, Right, Up, Down };

static public Direction dir;

static public void ssset()

{

dir = Direction.Right;

gameOver = false;

w = width / 2 - 1;

h = height / 2 - 1; //Установка начальных параметров

headX = width / 2;

headY = height / 2;

var rnd = new Random();

foodX = rnd.Next(1, width - 1);

foodY = rnd.Next(1, height - 1);

score = 0;

nTail = 0;

}

public void WriteBoard()

{

Console.SetCursorPosition(0, 0);

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

{

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

{

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

{

Console.Write("\*");

}

else if (j == 0 || j == width - 1) //Создание поля

{

Console.Write("\*");

}

else if (j == foodX && i == foodY)

{

Console.Write("+");

}

else if (j == headX && i == headY)

{

Console.Write("$");

}

else

{

isprinted = false;

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

{

if (TailX[k] == j && TailY[k] == i)

{

Console.Write("o");

isprinted = true;

}

}

if (!isprinted)

{

Console.Write(" ");

}

}

}

Console.WriteLine();

}

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine(" Игра Змейка ");

Console.WriteLine();

Console.WriteLine("Ваш счет:" + score);

}

public void difficult()

{

Console.SetWindowSize(width, height);

Console.WriteLine(" Приветствую в игре повелитель змея ");

Console.WriteLine(" Выбери уровень сложности:" +

"\n 1 - Easy peasy" +

"\n 2 - Medium" +

"\n 3 - Death"

+

"\n Р - пауза");

int D = Convert.ToInt32(Console.ReadLine());

Console.Clear();

switch (D)

{

case 1:

e = 100;

height = 20;

width = 20;

break;

case 2:

e = 50;

height = 25;

width = 25; //Выбор сложности

break;

case 3:

e = 20;

height = 30;

width = 30;

break;

}

Console.Clear();

}

void Input()

{

while (Console.KeyAvailable)

{

keypress = Console.ReadKey(true);

if (keypress.Key == ConsoleKey.Escape)

{

Environment.Exit(0);

}

if (keypress.Key == ConsoleKey.P)

{

dir = Direction.Stop;

}

else if (keypress.Key == ConsoleKey.LeftArrow && dir != Direction.Right)

{

dir = Direction.Left;

}

else if (keypress.Key == ConsoleKey.RightArrow && dir != Direction.Left)

{

dir = Direction.Right;

}

else if (keypress.Key == ConsoleKey.LeftArrow)

{

dir = Direction.Left;

}

else if (keypress.Key == ConsoleKey.UpArrow && dir != Direction.Down)

{

dir = Direction.Up;

}

else if (keypress.Key == ConsoleKey.DownArrow && dir != Direction.Up)

{

dir = Direction.Down;

}

}

Thread.Sleep(e);

}

public void Logic()

{

int PreX = TailX[0];

int PreY = TailY[0];

int tempX, tempY;

if (dir != Direction.Stop)

{

TailX[0] = headX;

TailY[0] = headY;

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

{

tempX = TailX[i];

tempY = TailY[i]; //Построение логики игры

TailX[i] = PreX;

TailY[i] = PreY;

PreX = tempX;

PreY = tempY;

}

}

switch (dir)

{

case Direction.Right:

headX += step;

break;

case Direction.Left:

headX -= step;

break;

case Direction.Up:

headY -= step;

break;

case Direction.Down:

headY += step;

break;

case Direction.Stop:

break;

}

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

{

gameOver = true;

}

else //Логика проигрыша

{

gameOver = false;

}

if (headX == foodX && headY == foodY)

{

score += 10; //Поедание и начисление очков

nTail++;

var rnd = new Random();

foodX = rnd.Next(1, width - 1);//размещение еды по новой

foodY = rnd.Next(1, height - 1);//размещение еды по новой

}

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

{

if (TailX[i] == headX && TailY[i] == headY)

{

if (horizontal || vertical)

{

gameOver = false;

}

else

{

gameOver = true;

}

}

var rnd = new Random();

if (TailX[i] == foodX && TailY[i] == foodY)

{

foodX = rnd.Next(1, width - 1);

foodY = rnd.Next(1, height - 1);

}

}

//установка последовательности элементов змеи

switch (dir)

{

case Direction.Up:

headY--;

break;

case Direction.Down:

headY++;

break;

case Direction.Right:

headX++;

break;

case Direction.Left:

headX--;

break;

case Direction.Stop:

dir = 0;

break;

}

}

void Update()

{

while (!gameOver)

{

WriteBoard();

Logic();

Input();

}

}

static void Main(string[] args)

{

Console.ForegroundColor = ConsoleColor.Green;

Console.BackgroundColor = ConsoleColor.Black;

Program program = new Program();

program.difficult();

while (true)

{

ssset();

program.Update();

Console.Clear();

}

}

}

}



