ДЕПАРТАМЕНТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ТОМСКОЙ ОБЛАСТИ

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

Специальность 09.02.07 «Информационные системы и программирование»

ОТЧЁТ ПО ОПБД

|  |  |  |
| --- | --- | --- |
| Студент  «\_\_» \_\_\_\_\_\_\_\_ 2020 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.М. Сучков |
| Преподаватель  «\_\_» \_\_\_\_\_\_\_\_ 2020 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_ | В.С. Павлов |

Томск 2020

Задание:

Нужно разработать игру «змейку»

Функциональные требования:

✓ управление змейкой должно происходить по клавишам (вверх, вниз, влево, вправо).

✓ Змейка должна неприрывно двигаться в сторону, куда нажал пользователь, по нажатии на другую клавишу, змейка должна поменять свое движение в сторону нажатой клавиши

✓ скорость и размер поля должен меняться в зависимости от выбранной пользователем сложности игры

легко[поле10х10, скорость медленная]

нормально[поле15х15,скорость нормальная]

сложно[после20х20,скорость быстрая]

- out/ref

Требования к оформлению:

✓ поле должно быть очерченым и понятным для пользователя

✓ змейка должна обладать головой и телом (символы ASCII)

✓ «фрукты» должны так же иметь свои собственные символы (ASCII)

Требования к выполнению работы(общ)

✓ работа должна быть выполнена на функциях и классах

✓ поле и змейка не должны «мерцать» при движении

Требования к отчету:

- файл с отчетом должен содержать: титульный лист, задание, результат работы (скриншоты), листинг программы с подробными комментариями.

Скриншотов нет по техническим причинам

Основной код (Program.cs):

using System;

using System.Threading;

namespace Snake

{

class Program

{

static void Main(string[] args)

{

Console.ForegroundColor = ConsoleColor.Cyan;

Console.WriteLine("Выберите уровень слодности и введите его номер:");

Console.WriteLine("1. Лёгкий уровень [поле10х10, скорость медленная]");

Console.WriteLine("2. Средний уровень [поле15х15, скорость нормальная]");

Console.WriteLine("3. Сложный уровень [поле20х20, скорость быстрая]");

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

switch (V)

{

case 1:

Console.Clear();

Class1 class1 = new Class1();

class1.Setup(class1.proigrish);

while (!class1.proigrish)

{

class1.karta\_and\_pers();

class1.Input\_Logic();

class1.dvigenie();

class1.itog();

Thread.Sleep(200);

}

break;

case 2:

Console.Clear();

Class2 class2 = new Class2();

class2.Setup(class2.proigrish);

while (!class2.proigrish)

{

class2.karta\_and\_pers();

class2.Input\_Logic();

class2.dvigenie();

class2.itog();

Thread.Sleep(150);

}

break;

case 3:

Console.Clear();

Class3 class3 = new Class3();

class3.Setup(class3.proigrish);

while (!class3.proigrish)

{

class3.karta\_and\_pers();

class3.Input\_Logic();

class3.dvigenie();

class3.itog();

Thread.Sleep(100);

}

break;

default:

Console.WriteLine("ERROR");

break;

}

Console.ReadKey();

}

}

}

Лёгкий уровень сложности (Class1.cs):

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

namespace Snake

{

class Class1

{

public bool proigrish = false;

int kolvo\_syedaniy = 0;

int snakex;

int snakey;

int fruitX, fruitY;

const int shirina = 12;

const int visota = 12;

int[] hvostX = new int[100];

int[] hvostY = new int[100];

int nomerhv;

Random R = new Random();

ConsoleKeyInfo knopki;

public bool Setup(bool proigrish)

{

proigrish = false;

snakex = shirina / 2 - 1;

snakey = visota / 2 - 1;

fruitX = R.Next(1, 10);

fruitY = R.Next(1, 10);

Console.WriteLine();

return proigrish;

}

public void karta\_and\_pers()

{

Console.WriteLine();

Console.SetCursorPosition(0, 0);

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

Console.Write((char)35);

Console.WriteLine();

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

{

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

{

if (j == 0 || j == shirina - 1)

Console.Write((char)35);

if (i == snakey && j == snakex)

Console.Write((char)64);

else if (i == fruitY && j == fruitX)

Console.Write((char)111);

else

{

bool print = false;

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

{

if (hvostX[k] == j && hvostY[k] == i)

{

print = true;

Console.Write((char)97);

}

}

if (!print)

Console.Write(" ");

}

}

Console.WriteLine();

}

for (int j = 0; j < shirina + 1; j++)

Console.Write((char)35);

Console.WriteLine();

Console.WriteLine("Длина хвоста " + kolvo\_syedaniy);

}

public void Input\_Logic()

{

int predznX = hvostX[0];

int predznY = hvostY[0];

int pred2X;

int pred2Y;

hvostX[0] = snakex;

hvostY[0] = snakey;

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

{

pred2X = hvostX[i];

pred2Y = hvostY[i];

hvostX[i] = predznX;

hvostY[i] = predznY;

predznX = pred2X;

predznY = pred2Y;

}

}

public void dvigenie()

{

if (Console.KeyAvailable == true)

{

knopki = Console.ReadKey(true);

}

switch (knopki.Key)

{

case ConsoleKey.UpArrow:

snakey--;

break;

case ConsoleKey.DownArrow:

snakey++;

break;

case ConsoleKey.RightArrow:

snakex++;

break;

case ConsoleKey.LeftArrow:

snakex--;

break;

}

}

public void itog()

{

if (snakex > shirina)

snakex = 0;

else if (snakex < 0)

snakex = shirina - 1;

if (snakey > visota)

snakey = 0;

else if (snakey < 0)

snakey = visota - 1;

for (int g = 0; g < nomerhv; g++)

{

if (hvostX[g] == snakex && hvostY[g] == snakey)

{

proigrish = true;

Console.WriteLine("GAME OVER");

Console.WriteLine("RESTART 'R'");

}

}

if (proigrish != false)

{

char RESTART = Convert.ToChar(Console.ReadLine());

if (RESTART == 'R')

{

Class1 class1 = new Class1();

class1.Setup(class1.proigrish);

while (!class1.proigrish)

{

class1.karta\_and\_pers();

class1.Input\_Logic();

class1.dvigenie();

class1.itog();

Thread.Sleep(150);

}

Console.ReadKey();

}

}

if (snakex == fruitX && snakey == fruitY)

{

kolvo\_syedaniy++;

fruitX = R.Next(1, visota-2);

fruitY = R.Next(1, shirina-2);

nomerhv++;

}

}

}

}

Нормальный уровень сложности (Class2.cs):

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

namespace Snake

{

class Class2

{

public bool proigrish = false;

int kolvo\_syedaniy = 0;

int snakex;

int snakey;

int fruitX, fruitY;

const int shirina = 17;

const int visota = 17;

int[] hvostX = new int[100];

int[] hvostY = new int[100];

int nomerhv;

Random R = new Random();

ConsoleKeyInfo knopki;

public bool Setup(bool proigrish)

{

proigrish = false;

snakex = shirina / 2 - 1;

snakey = visota / 2 - 1;

fruitX = R.Next(1, 10);

fruitY = R.Next(1, 10);

Console.WriteLine();

return proigrish;

}

public void karta\_and\_pers()

{

Console.WriteLine();

Console.SetCursorPosition(0, 0);

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

Console.Write((char)35);

Console.WriteLine();

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

{

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

{

if (j == 0 || j == shirina - 1)

Console.Write((char)35);

if (i == snakey && j == snakex)

Console.Write((char)64);

else if (i == fruitY && j == fruitX)

Console.Write((char)111);

else

{

bool print = false;

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

{

if (hvostX[k] == j && hvostY[k] == i)

{

print = true;

Console.Write((char)97);

}

}

if (!print)

Console.Write(" ");

}

}

Console.WriteLine();

}

for (int j = 0; j < shirina + 1; j++)

Console.Write((char)35);

Console.WriteLine();

Console.WriteLine("Длина хвоста " + kolvo\_syedaniy);

}

public void Input\_Logic()

{

int predznX = hvostX[0];

int predznY = hvostY[0];

int pred2X;

int pred2Y;

hvostX[0] = snakex;

hvostY[0] = snakey;

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

{

pred2X = hvostX[i];

pred2Y = hvostY[i];

hvostX[i] = predznX;

hvostY[i] = predznY;

predznX = pred2X;

predznY = pred2Y;

}

}

public void dvigenie()

{

if (Console.KeyAvailable == true)

{

knopki = Console.ReadKey(true);

}

switch (knopki.Key)

{

case ConsoleKey.UpArrow:

snakey--;

break;

case ConsoleKey.DownArrow:

snakey++;

break;

case ConsoleKey.RightArrow:

snakex++;

break;

case ConsoleKey.LeftArrow:

snakex--;

break;

}

}

public void itog()

{

if (snakex > shirina)

snakex = 0;

else if (snakex < 0)

snakex = shirina - 1;

if (snakey > visota)

snakey = 0;

else if (snakey < 0)

snakey = visota - 1;

for (int g = 0; g < nomerhv; g++)

{

if (hvostX[g] == snakex && hvostY[g] == snakey)

{

proigrish = true;

Console.WriteLine("GAME OVER");

Console.WriteLine("RESTART 'R'");

}

}

if (proigrish != false)

{

char RESTART = Convert.ToChar(Console.ReadLine());

if (RESTART == 'R')

{

Class2 class2 = new Class2();

class2.Setup(class2.proigrish);

while (!class2.proigrish)

{

class2.karta\_and\_pers();

class2.Input\_Logic();

class2.dvigenie();

class2.itog();

Thread.Sleep(150);

}

Console.ReadKey();

}

}

if (snakex == fruitX && snakey == fruitY)

{

kolvo\_syedaniy++;

fruitX = R.Next(1, visota - 2);

fruitY = R.Next(1, shirina - 2);

nomerhv++;

}

}

}

}

Сложный уровень сложности (Class3.cs):

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

namespace Snake

{

class Class3

{

public bool proigrish = false;

int kolvo\_syedaniy = 0;

int snakex;

int snakey;

int fruitX, fruitY;

const int shirina = 22;

const int visota = 22;

int[] hvostX = new int[100];

int[] hvostY = new int[100];

int nomerhv;

Random R = new Random();

ConsoleKeyInfo knopki;

public bool Setup(bool proigrish)

{

proigrish = false;

snakex = shirina / 2 - 1;

snakey = visota / 2 - 1;

fruitX = R.Next(1, 10);

fruitY = R.Next(1, 10);

Console.WriteLine();

return proigrish;

}

public void karta\_and\_pers()

{

Console.WriteLine();

Console.SetCursorPosition(0, 0);

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

Console.Write((char)35);

Console.WriteLine();

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

{

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

{

if (j == 0 || j == shirina - 1)

Console.Write((char)35);

if (i == snakey && j == snakex)

Console.Write((char)64);

else if (i == fruitY && j == fruitX)

Console.Write((char)111);

else

{

bool print = false;

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

{

if (hvostX[k] == j && hvostY[k] == i)

{

print = true;

Console.Write((char)97);

}

}

if (!print)

Console.Write(" ");

}

}

Console.WriteLine();

}

for (int j = 0; j < shirina + 1; j++)

Console.Write((char)35);

Console.WriteLine();

Console.WriteLine("Длина хвоста " + kolvo\_syedaniy);

}

public void Input\_Logic()

{

int predznX = hvostX[0];

int predznY = hvostY[0];

int pred2X;

int pred2Y;

hvostX[0] = snakex;

hvostY[0] = snakey;

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

{

pred2X = hvostX[i];

pred2Y = hvostY[i];

hvostX[i] = predznX;

hvostY[i] = predznY;

predznX = pred2X;

predznY = pred2Y;

}

}

public void dvigenie()

{

if (Console.KeyAvailable == true)

{

knopki = Console.ReadKey(true);

}

switch (knopki.Key)

{

case ConsoleKey.UpArrow:

snakey--;

break;

case ConsoleKey.DownArrow:

snakey++;

break;

case ConsoleKey.RightArrow:

snakex++;

break;

case ConsoleKey.LeftArrow:

snakex--;

break;

}

}

public void itog()

{

if (snakex > shirina)

snakex = 0;

else if (snakex < 0)

snakex = shirina - 1;

if (snakey > visota)

snakey = 0;

else if (snakey < 0)

snakey = visota - 1;

for (int g = 0; g < nomerhv; g++)

{

if (hvostX[g] == snakex && hvostY[g] == snakey)

{

proigrish = true;

Console.WriteLine("GAME OVER");

Console.WriteLine("RESTART 'R'");

}

}

if (proigrish != false)

{

char RESTART = Convert.ToChar(Console.ReadLine());

if (RESTART == 'R')

{

Class3 class3 = new Class3();

class3.Setup(class3.proigrish);

while (!class3.proigrish)

{

class3.karta\_and\_pers();

class3.Input\_Logic();

class3.dvigenie();

class3.itog();

Thread.Sleep(100);

}

Console.ReadKey();

}

}

if (snakex == fruitX && snakey == fruitY)

{

kolvo\_syedaniy++;

fruitX = R.Next(1, visota - 2);

fruitY = R.Next(1, shirina - 2);

nomerhv++;

}

}

}

}