Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования «Национальный

исследовательский Нижегородский государственный

университет им. Н.И. Лобачевского»

**Институт информационных технологий, математики и механики**

Лабораторная работа №4 по курсу «Технология визуального программирования»:

«Разработка игровых проектов с использованием WPF. Игра «Точки и отрезки»»

Выполнил: студент группы 381808-1

Оганян Роберт Владимирович

Преподаватель: доцент кафедры ПИнж

Борисов Н. А.

Нижний Новгород

2019 г.

# Содержание

[Содержание 2](#_Toc4101358)

[Описание работы программы 3](#_Toc4101359)

[Приложение 7](#_Toc4101361)

## Описание работы программы

Программа выполнена в среде разработки microsoft visual studio 2017 с помощью Windows Form и WPF (Windows Presentation Foundation**)**. Приложение написано для операционной системы Windows.

**Суть игры «Точки и отрезки»**

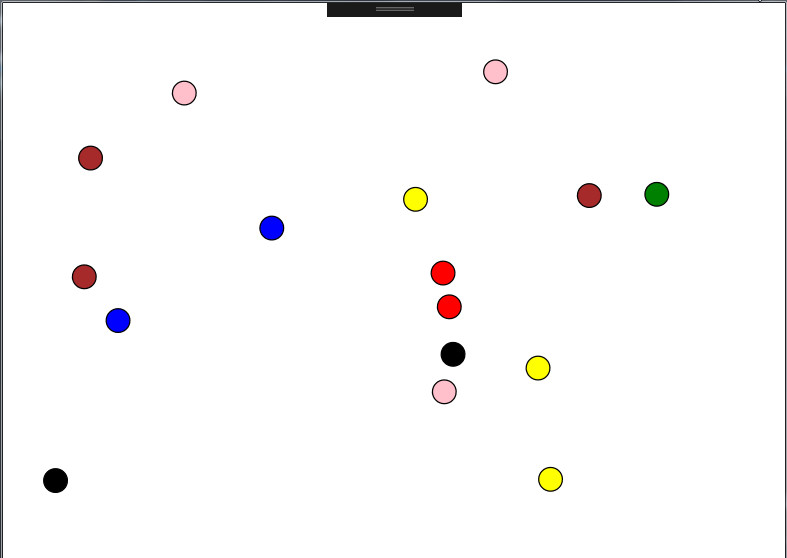
Условия этой игры на бумаге просты: на листе бумаги ставится несколько точек (в нашем случае 16). Играют двое, поочередно соединяя две любые точки отрезком. Захватывать 3-ю точку нельзя, и каждая точка может быть концом лишь одного отрезка. Отрезки не должны пересекаться. Проигрывает тот, кто не сможет сделать ход.

**Реализация:**

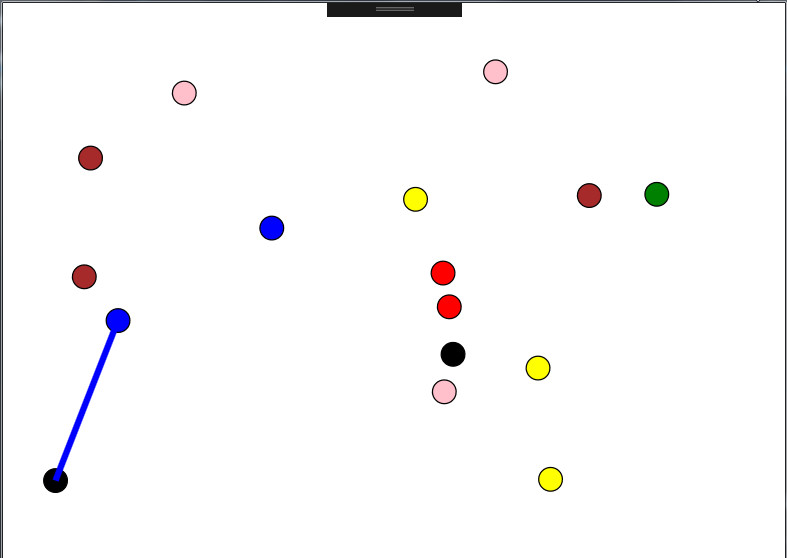
Программа реализована с помощью 4-ех массивов (List):

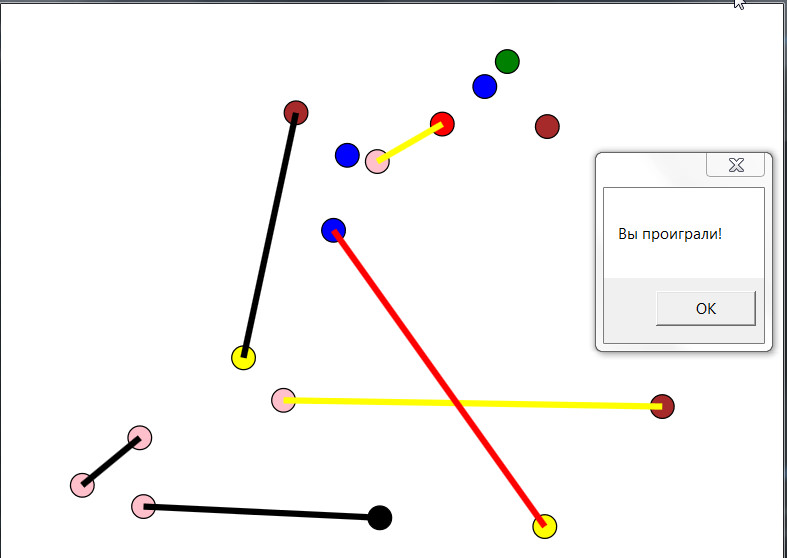
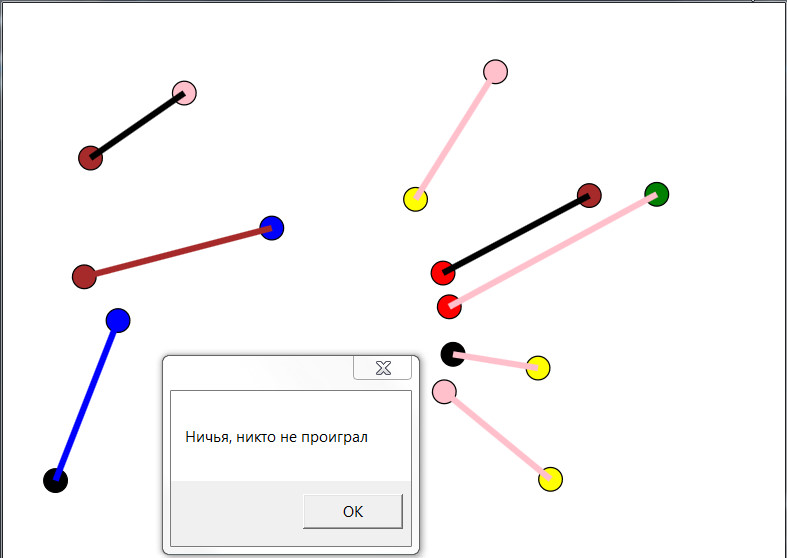
* Эллипсы
* Координаты эллипсов
* Массив флагов, показывающий есть ли у эллипса данного индекса подведенная к нему линия
* Массив линий между точками

Таким образом, при загрузке программа рандомно генерировала 16 эллипсов и заполняла 1-2 массив, а третий массив, отвечающий за использование/неиспользование точек первоначально был равен 0.

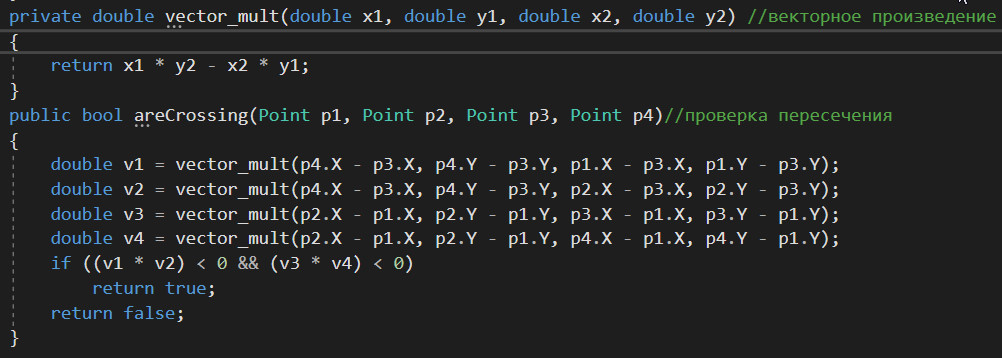


Начинается игра: при попадании нажатия мышки на эллипс вызывается функция Game, которая, если у этой точки нет смежных отрезков, запоминает ее индекс в массиве.

Когда мы выбираем вторую такую точку, то сначала рисуется линия и кидается в массив линий. 

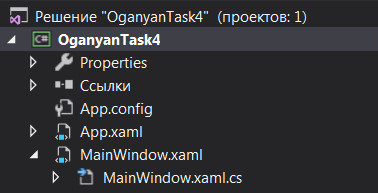
А затем с помощью геометрии ( функции areCrossing и вспомогательная vector\_mult) считаем, пересекаются ли отрезки. Если нет, то игра продолжается до тех пор, пока не закончатся точки (ничья). Если да - вы проиграли. 

Реализация методов проверки отрезков на пересечение:



### Приложение

Схема файла OganyanTask4



**Файл MainWindow.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Windows.Threading;

using System.Drawing;

namespace WpfApp3

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

Random ran;

Brush[] Colors = { Brushes.Black,Brushes.Yellow, Brushes.Red, Brushes.Blue,

Brushes.Pink, Brushes.Pink, Brushes.Green, Brushes.Brown };

int xwidth = 640;

int yheight = 480;

List<Line> li;

List<Ellipse> El;

List<int> vektor;

List<Point> kord;

int flag=0; // отвечает за количество выбранных точек

int number=0; // отвечает за позицию в массиве

Ellipse help;

public MainWindow()

{

InitializeComponent();

ran = new Random();

this.Width = xwidth;

this.Height = yheight;

El = new List<Ellipse>();

li = new List<Line>();

vektor = new List<int>();

kord = new List<Point>();

}

private void Drawpoints(object sender, EventArgs e)

{

int x, y;

Random r = new Random();

for (int i = 0; i < 16; i++)

{

x = r.Next(5, (int)(0.9 \* xwidth));

y = r.Next(5, (int)(0.9 \* yheight));

Ellipse knew = new Ellipse();

knew.Height = 20;

knew.Width = 20;

knew.Margin = new Thickness(x, y, 0, 0);

knew.Stroke = Brushes.Black;

knew.Fill = Colors[ran.Next(Colors.Length)];

knew.HorizontalAlignment = 0;

knew.VerticalAlignment = 0;

El.Add(knew);

knew.MouseUp += Game;

vektor.Add(0);

kord.Add(new Point(x, y));

Grid1.Children.Add(knew);

}

}

private void Game(object sender, MouseEventArgs e)

{

int i, number2 = 0;

bool a = true;

Ellipse rec = sender as Ellipse;

for (i = 0; i < El.Count; i++) // Проверка если эта точка уже взята

{

if (rec == El[i])

if (vektor[i] == 1) return;

else

{

number2 = i;

break;

}

}

if (flag == 0)

{

number = number2;

vektor[number] = 1;

flag++;

}

else if (flag == 1)

{

Line myLine;

myLine = new Line();

myLine.Stroke = Colors[ran.Next(Colors.Length)];

myLine.X1 = kord[number].X+10;

myLine.X2 = kord[number2].X + 10;

myLine.Y1 = kord[number].Y + 10;

myLine.Y2 = kord[number2].Y + 10;

myLine.StrokeThickness = 5;

li.Add(myLine);

Grid1.Children.Add(myLine);

for (int j = 0; j < li.Count; j++)

{

if (areCrossing(kord[number], kord[number2], new Point(li[j].X1, li[j].Y1),

new Point(li[j].X2, li[j].Y2)) == true) {

MessageBox.Show("Вы проиграли!");

App.Current.MainWindow.Close();

}

}

vektor[number2] = 1;

number = number2 = 0;

flag = 0;

}

for (i = 0; i < El.Count; i++)

{

if (vektor[i] == 0)

{

a = false;

break;

}

}

if (a == true)

{

MessageBox.Show("Ничья, никто не проиграл");

App.Current.MainWindow.Close();

}

}

private double vector\_mult(double x1, double y1, double x2, double y2) //векторное произведение

{

return x1 \* y2 - x2 \* y1;

}

public bool areCrossing(Point p1, Point p2, Point p3, Point p4)//проверка пересечения

{

double v1 = vector\_mult(p4.X - p3.X, p4.Y - p3.Y, p1.X - p3.X, p1.Y - p3.Y);

double v2 = vector\_mult(p4.X - p3.X, p4.Y - p3.Y, p2.X - p3.X, p2.Y - p3.Y);

double v3 = vector\_mult(p2.X - p1.X, p2.Y - p1.Y, p3.X - p1.X, p3.Y - p1.Y);

double v4 = vector\_mult(p2.X - p1.X, p2.Y - p1.Y, p4.X - p1.X, p4.Y - p1.Y);

if ((v1 \* v2) < 0 && (v3 \* v4) < 0)

return true;

return false;

}

}

}