

ГБОУ "Президентский ФМЛ № 239" или Государственное бюджетное  
общеобразовательное учреждение "Президентский физико-математической лицей №  
239"

Ученицы 10-3 Власовой Аллы



## Информатика. Проектная деятельность в 10-м классе

### Постановка задачи

1. Заданы два множества точек на плоскости. Построить пересечения и разность этих множеств.

Вводятся 2 множества и необходимо найти общие и несовпадающие точки.

2. Создание игры, представляющей собой все увеличивающую скорость точку, которая должна избежать соприкосновений с краями панели и при этом наибольшее количество раз соприкоснуться с автоматически задающимися синими точками.

### Математическая модель

1

## Множество

**Множество – это набор, совокупность объектов.**

**Элемент множества – это объект из этого множества.**



ПЕРЕСЕЧЕНИЕ МНОЖЕСТВ — понятие

теории множеств; пересечение множеств множество, состоящее из всех тех элементов, которые принадлежат одновременно всем данным множествам.

Разностью множеств А и В называется множество, содержащее те и только те элементы, которые принадлежат множеству А и не принадлежат множеству В.

2.

Движение происходит перерисовыванием игрового тела в выбранном направлении.

Координаты еды выбираются случайно.

Движение происходит по кнопкам.

Уточнение исходных и выходных данных

Входные данные:

1. 2 клика на панель или 2 натуральных числа в специально отведенные поле.

Введенные числа ограничиваются размерами панели ввода и вывода.

Поскольку нам не известно количество точек в каждом множестве, стоит использовать Array List.

2. отсутствуют

Выходные данные:

1. Два Array List, содержащие точки пересечения множеств и их разности соответственно.

2. отсутствуют

Анализ используемой структуры данных

1. Вводимые данные- 4 листа типа int, 2 из которых формируют множества A, а другие – B. Имеется проверка корректности.

Выходные данные, которые необходимо хранить- 4 листа int, представляющие собой координаты множества пересечения и разности.

2.

Входные данные- строка (string), представляющая собой ник, который будет сохранен в случае побития рекорда.

Так же входные данные- 6 int, характеризующие цвета, 3- игрового тела, 3- поля.

Так же необходимо хранить 4 переменные типа int- место положение тела по x и y, и аналогичные данные для бонуса.

2 int- текущий счет и рекорд.

Bool- отвечающий на вопрос, закончена ли игра.

И Direction dir – определяющий направление.

Выбор метода решения (описание алгоритма)

1.

1) Введение данных в два листа

- 2) Сравнение листов в цикле
- 3) Выявление полных совпадений и занесение их в еще два листа
- 4) Занесение несовпадения в последние 2 листа
- 5) Вывод получившихся точек

2.

- 1) Оформление дизайна поля
- 2) Составления реакции программы на нажатие различных кнопок
- 3) алгоритм движения
- 4) Алгоритм получения очков
- 5) Выведение рекорда
- 6) описания окончания игры

Комментированный листинг

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;
```

```
namespace проект_13  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
            pic_A_X = new List<int>();  
            pic_A_Y = new List<int>();  
        }  
    }  
}
```

```

//-----Старт
private void button1_Click(object sender, EventArgs e)
{
    DialogResult okay = MessageBox.Show("Введите множество A");
    if (okay == DialogResult.OK)
    {
        pic1 pic_ = new pic1(false);
        pic_.ShowDialog();
        pic_A_X = pic_.pic_X;
        pic_A_Y = pic_.pic_Y;
    }

    DialogResult okby = MessageBox.Show("Введите множество B");
    if (okby == DialogResult.OK)
    {
        pic1 pic_ = new pic1(true);
        pic_.ShowDialog();
        pic_B_X = pic_.pic_X;
        pic_B_Y = pic_.pic_Y;
    }
}

private List<int> pic_A_X;
private List<int> pic_A_Y;
private List<int> pic_B_X;
private List<int> pic_B_Y;
private List<int> pic_C_X;
private List<int> pic_C_Y;
private List<int> pic_D_X;
private List<int> pic_D_Y;

//показ множества A
private void button2_Click(object sender, EventArgs e)
{

```

```

String s = "множество A";
Brush p = Brushes.Red;
Result res_ = new Result(p, pic_A_X, pic_A_Y,s);
res_.ShowDialog();
}

//показ B
private void button3_Click(object sender, EventArgs e)
{
    String s="множество B";
    Brush p = Brushes.Blue;
    Result res_ = new Result(p, pic_B_X, pic_B_Y,s);
    res_.ShowDialog();
}

private void button4_Click(object sender, EventArgs e)
{
    String s = "пересечение множеств A и B";
    Brush p = Brushes.Plum;
    pic_C_X = new List<int>();
    pic_C_Y = new List<int>();

    for (int i = 0; i < pic_A_X.Count; i++)
    {
        for (int j = 0; j < pic_B_X.Count; j++)
        {
            if((pic_A_X[i]==pic_B_X[j])&&(pic_A_Y[i]==pic_B_Y[j])){
                pic_C_X.Add(pic_A_X[i]);
                pic_C_Y.Add(pic_A_Y[i]);
            }
        }
    }

    Result res_ = new Result(p, pic_C_X, pic_C_Y, s);

```

```

        res_.ShowDialog();
    }

private void button5_Click(object sender, EventArgs e)
{
    String s = "разность множеств A и B";
    Brush p = Brushes.Green;
    pic_D_X = new List<int>();
    pic_D_Y = new List<int>();

    bool k=true;
    for (int i = 0; i < pic_A_X.Count; i++)
    {
        for (int j = 0; j < pic_B_X.Count; j++)
        {
            if ((pic_A_X[i] == pic_B_X[j]) && (pic_A_Y[i] == pic_B_Y[j]))
            {
                k = false;
            }
        }
        if (k) {
            pic_D_X.Add(pic_A_X[i]);
            pic_D_Y.Add(pic_A_Y[i]);
        }
        k = true;
    }

    Result res_ = new Result(p, pic_D_X, pic_D_Y, s);
    res_.ShowDialog();
}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace проект_13
{
    public partial class pic1 : Form
    {

        public pic1(bool blue)
        {
            InitializeComponent();
            pic_X = new List<int>();
            pic_Y = new List<int>();
            pic_X_temp = new List<int>();
            pic_Y_temp = new List<int>();
            if (blue)
                p = Brushes.Blue;
            else p = Brushes.Red;
            w = panel1.Width;
            h = panel1.Height;

        }

        private void pic1_Load(object sender, EventArgs e)
        {

        }
    }
}

```



```

private void pic1_MouseDoubleClick(object sender, MouseEventArgs e)
{

    //MessageBox.Show(e.Location.ToString());
    //Graphics g = this.CreateGraphics();
    //g.DrawEllipse(p, e.Location.X, e.Location.Y, 10, 10);
    //pic_X.Add(e.Location.X);
    //pic_Y.Add(e.Location.Y);

}

private void panel1_Paint(object sender, PaintEventArgs e)
{

}

private void panel1_MouseDoubleClick(object sender, MouseEventArgs e)
{
    label2.Text = "x= " + e.Location.X + " " + "y= " + (panel1.Height -
e.Location.Y).ToString();
    // MessageBox.Show(label2.Text);
    Graphics g = panel1.CreateGraphics();
    g.FillEllipse(p, e.Location.X-5, e.Location.Y-5, 10, 10);
    pic_X_temp.Add(e.Location.X);
    pic_Y_temp.Add(panel1.Height - e.Location.Y);
}

public List<int> pic_X;
public List<int> pic_Y;
private Brush p;
private int w;
private int h;
private List<int> pic_X_temp;
private List<int> pic_Y_temp;

```

```

private void label1_Click(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{
    int x;
    int y;
    if (int.TryParse(textBox1.Text, out x))
    {
        if (int.TryParse(textBox2.Text, out y))
        {
            pic_X_temp.Add(x);
            pic_Y_temp.Add(y);
            if ((x <= w) && (y <= h))
            {
                Graphics g = panel1.CreateGraphics();
                g.FillEllipse(p, x - 5, panel1.Height - y - 5, 10, 10);
            }
        }
        else MessageBox.Show("Исправьте Y");
    }
    else MessageBox.Show("Исправьте X");
}

```

```

private void pic1_Paint(object sender, PaintEventArgs e)
{

    Graphics g = panel1.CreateGraphics();
    Pen ps = Pens.Black;
    for (int i = 100; i < panel1.Width; i = i + 100)
    {
        g.DrawLine(ps, 0, panel1.Height - i, panel1.Width, panel1.Height - i);
    }
}

```

```

    }
    for (int i = 100; i < panel1.Height; i = i + 100)
    {
        g.DrawLine(ps, i, panel1.Height, i, 0);
    }
    for (int i = 0; i < pic_X_temp.Count; i++)
    {
        g.FillEllipse(p, pic_X_temp[i] - 5, panel1.Height - pic_Y_temp[i] - 5, 10, 10);
    }

}

private void button2_Click(object sender, EventArgs e)
{
    //сохранить
    for (int i = 0; i < pic_X_temp.Count; i++)
    {
        pic_X.Add(pic_X_temp[i]);
        pic_Y.Add(pic_Y_temp[i]);
    }
    this.Close();
}

}

}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;

```

```
using System.Windows.Forms;
```

```
namespace проект_13
```

```
{
```

```
    public partial class Result : Form
```

```
    {
```

```
        public Result(Brush p, List<int> pic_X, List<int> pic_Y, String s)
```

```
        {
```

```
            InitializeComponent();
```

```
            this.pic_X = pic_X;
```

```
            this.pic_Y = pic_Y;
```

```
            w = panel1.Width;
```

```
            h = panel1.Height;
```

```
            label2.Text = s;
```

```
            this.p= p;
```

```
        }
```

```
        private void panel1_Paint(object sender, PaintEventArgs e)
```

```
        {
```

```
            Graphics g = panel1.CreateGraphics();
```

```
            Pen ps = Pens.Black;
```

```
            for (int i = 100; i < panel1.Width; i = i + 100)
```

```
            {
```

```
                g.DrawLine(ps, 0, panel1.Height - i, panel1.Width, panel1.Height - i);
```

```
            }
```

```
            for (int i = 100; i < panel1.Height; i = i + 100)
```

```
            {
```

```
                g.DrawLine(ps, i, panel1.Height, i, 0);
```

```
            }
```

```
            for (int i = 0; i < pic_X.Count; i++)
```

```
            {
```

```
                g.FillEllipse(p, pic_X[i] - 5, panel1.Height - 5 - pic_Y[i], 10, 10);
```

```
            }
```

```

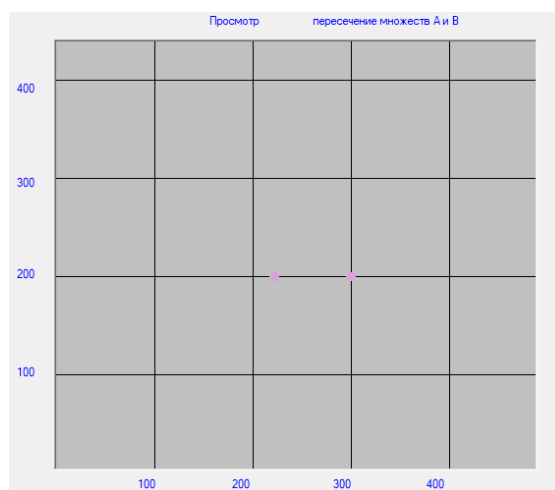
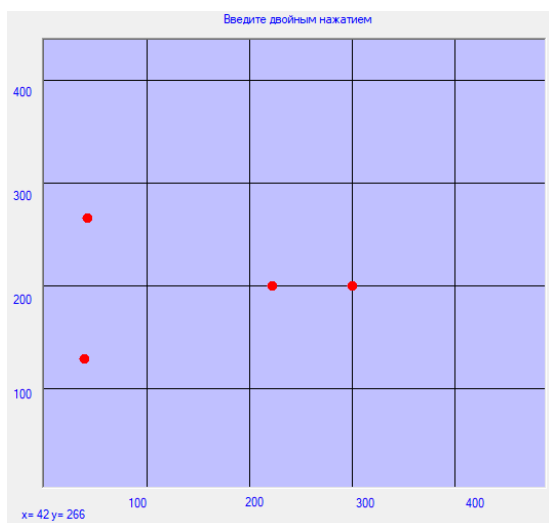
    }

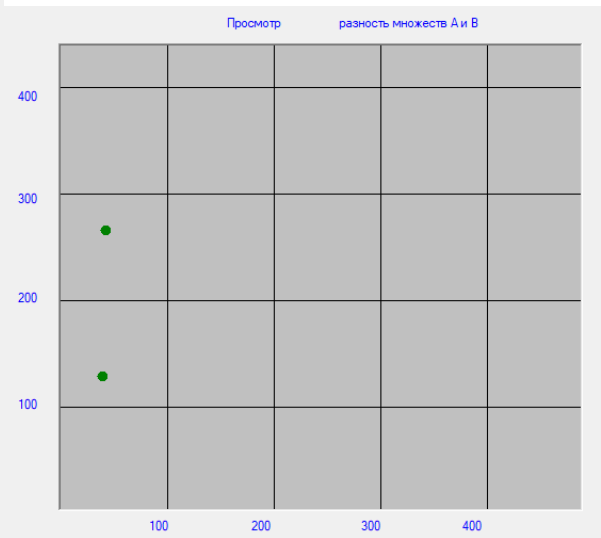
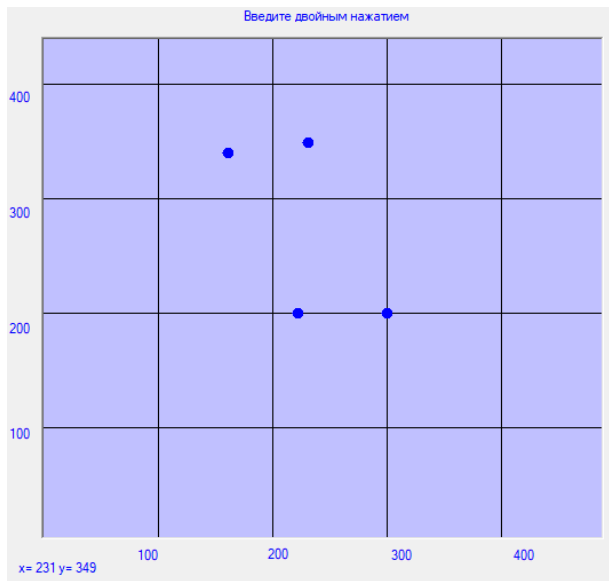
    public List<int> pic_X;
    public List<int> pic_Y;
    private Brush p;
    private int w;
    private int h;

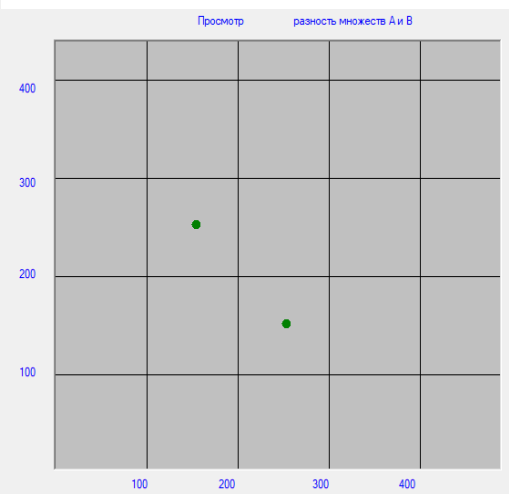
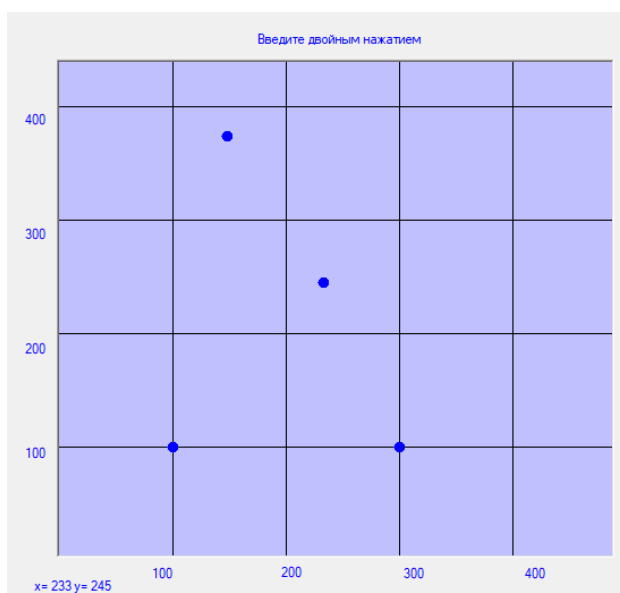
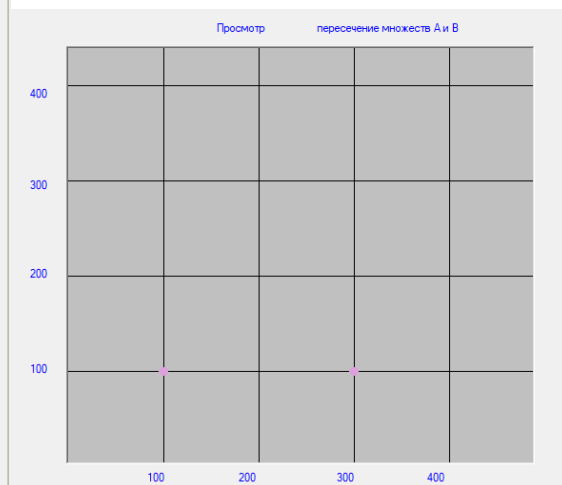
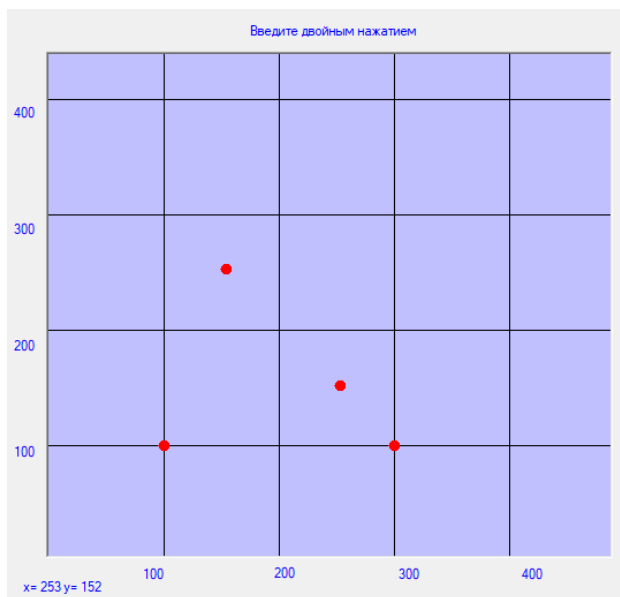
    }
}

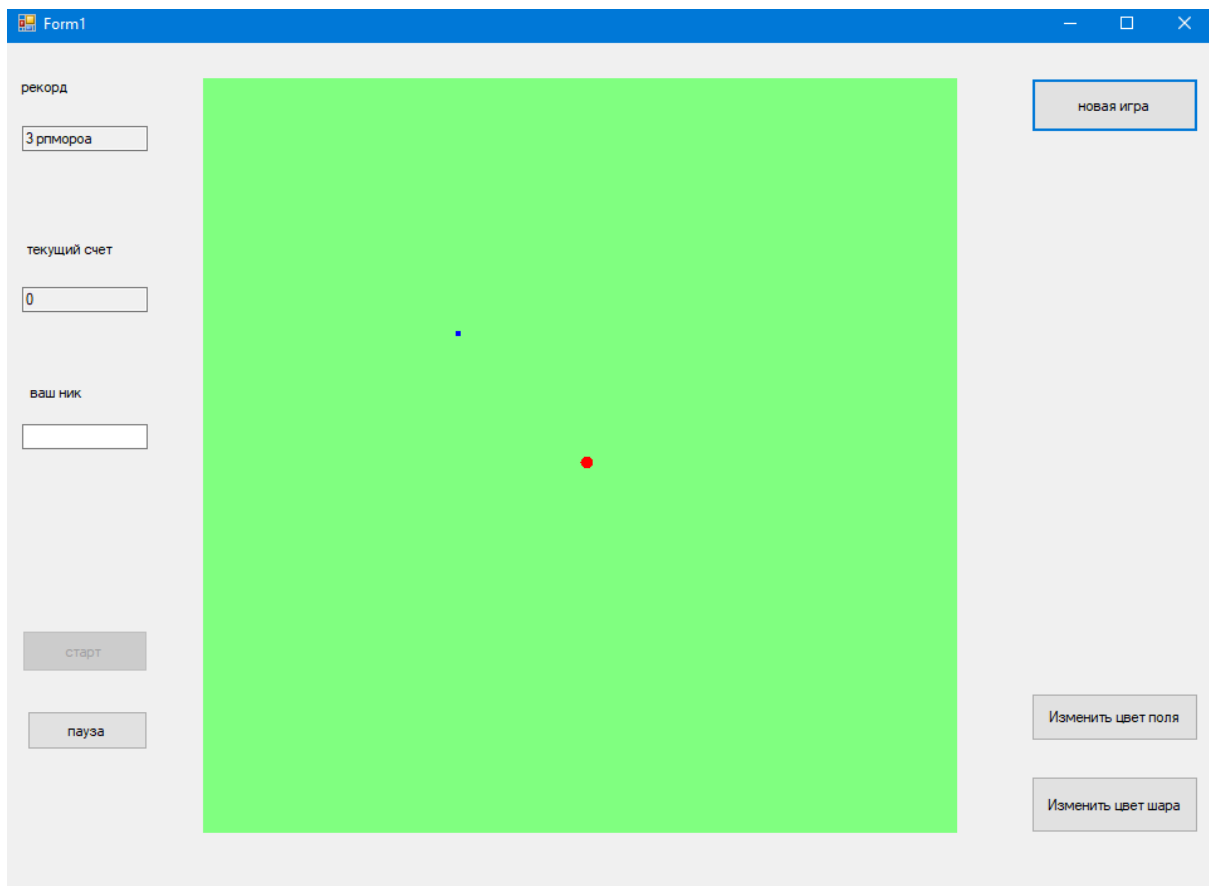
```

## Пример работы программы









#### Анализ правильности решения

1. Программа работает верно, поскольку она точно отображает требуемые множества и находит их пересечения и разность.
2. Программа работает верно, поскольку она верно показывает количество очков игрока и верно реагирует на все кнопки, так же она является конечной.