МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
КАФЕДРА САПР

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Компьютерная графика»

Тема: «Исследование алгоритмов отсечения отрезков и многоугольников окнами различного вида»

Вариант 1

Студенты гр. 9309 _____ Аль Сайед А.З.

_____ Серов А.В.

_____ Юшин Е.В.

Преподаватель _____ Матвеева И.В.

Санкт-Петербург

2022

Цель работы

Обеспечить реализацию алгоритма отсечения массива произвольных отрезков заданным прямоугольным окном с использование определенного алгоритма.

Задание

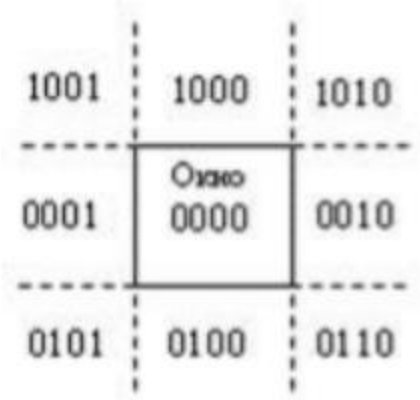
Обеспечить реализацию простого алгоритма отсечения массива произвольных отрезков следует формировать генератором случайных чисел. Вначале следует вывести на экран сгенерированные отрезки полностью, а затем другим цветом или яркостью те, которые полностью или частично попадают в область окна.

Математическая модель

Алгоритм заключается в следующем: четырем концам отрезка присваивается четырехбитный код: b0, b1, b2, b3. Данный код содержит информацию о положении точки относительно области её вывода. Возможны 9 следующих комбинаций:

1001	0001	0101
1000	0000	0100
1010	0010	0110

Схема расположения кодов



b0 = 0, когда x >= xmin

b0 = 1, когда x < xmin

b1 = 0, когда x <= xmin

b1 = 1, когда x > xmin

b2 = 0, когда y >= ymin

b2 = 1, когда y < ymin

b3 = 0, когда y <= ymin

b3 = 1, когда y > ymin

После того, как мы вычислили коды, возможно три варианта:

1. Наши коды содержат только нули, а значит отрезок полностью лежит внутри окна и не должен быть отрезан.

2. Наши коды содержат единичный бит в одной и той же позиции, из этого следует, что отрезок лежит за пределами окна и будет отрезан.

3. Во всех остальных случаях в окне лежит только часть отрезка, и это значит, что есть необходимость в отсечении.

Примеры работы программы

В данной программе мы выбираем количество отрезков с помощью трекбара, после чего отрезки рандомно генерируются в области, после чего по нажатию кнопки “Cut” остаются лишь те линии(и их части), которые попали в область квадрата.

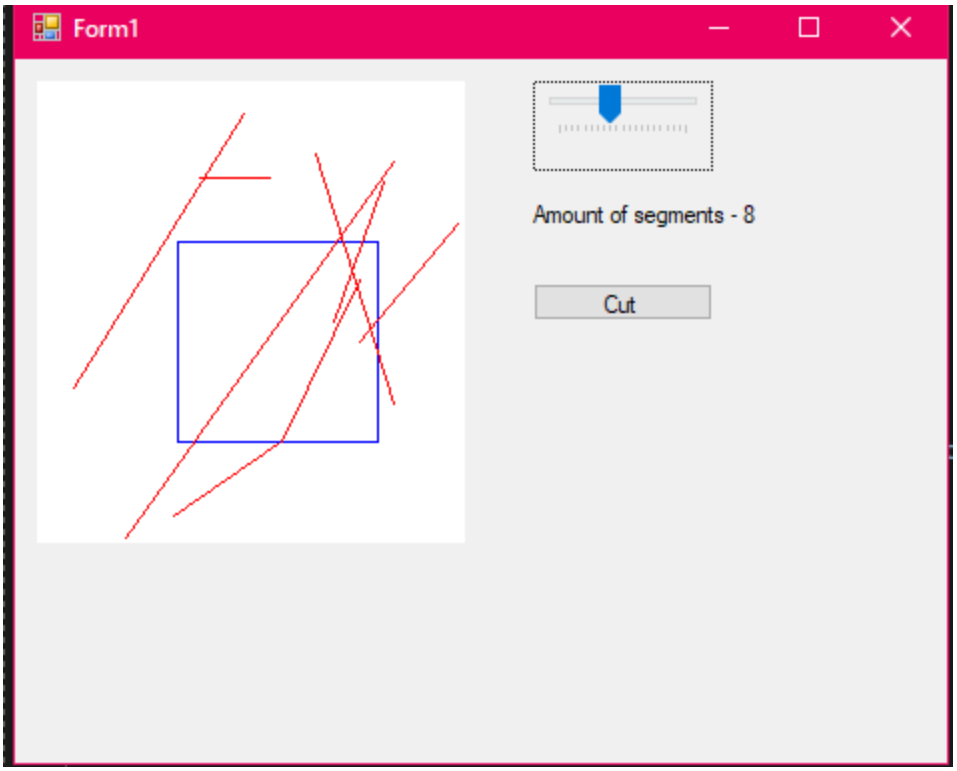


Рис. 1. Генерация линий.

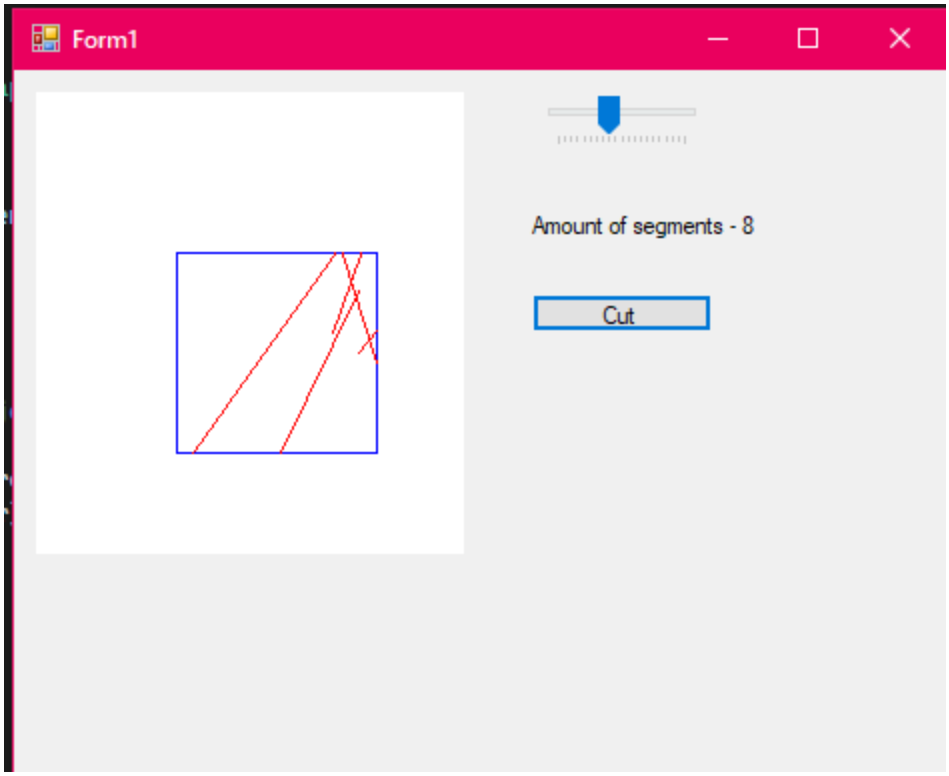


Рис. 2. Отрезание линий по нажатию кнопки “Cut”.

Код программы

Файл Form1.cs

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

namespace Lab_4
{
    public partial class Form1 : Form
    {
        Graphics g;
        PointF[] points;

        Pen pen_red = new Pen(Color.Red);
        Pen pen_blue = new Pen(Color.Blue);
        List<int> outer = new List<int>();

        public Form1()
        {
            InitializeComponent();
            pictureBox1.Image = new Bitmap(pictureBox1.Width, pictureBox1.Height);
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void trackBar1_Scroll(object sender, EventArgs e)
```

```
{
    g = Graphics.FromImage(pictureBox1.Image);
    g.Clear(pictureBox1.BackColor);
    pictureBox1.Invalidate();

    int w = pictureBox1.Width;
    int h = pictureBox1.Height;

    int k = trackBar1.Value * 2;

    points = new PointF[k];

    var rand = new Random();

    for (int i = 0; i < k; ++i)
    {
        points[i].X = rand.Next(w + 1);
        points[i].Y = rand.Next(h + 1);
    }

    if (k != 0)
    {
        g.DrawRectangle(pen_blue, new Rectangle(70, 80, 100, 100));
    }

    for (int i = 0; i < k - 1; i += 2)
    {
        g.DrawLine(pen_red, points[i], points[i + 1]);
    }

    label1.Text = $"Amount of segments - {k / 2}";
}

int get_code(PointF point)
{
    int code = 0b0000;
    code |= point.X < 70 ? 0b0001 : 0b0000;
    code |= point.X > 170 ? 0b0010 : 0b0000;
    code |= point.Y < 80 ? 0b1000 : 0b0000;
    code |= point.Y > 180 ? 0b0100 : 0b0000;
    return code;
}

int clip_segment(int i)
{
    int code_a = get_code(points[i]);
    int code_b = get_code(points[i + 1]);
    int code;
    PointF temp;

    while ((code_a | code_b) != 0b0000)
    {
        if ((code_a & code_b) != 0b0000)
        {
            return 2;
        }

        if (code_a != 0b0000)
        {
            code = code_a;
            temp = points[i];
        }
        else
        {
            code = code_b;
```

```
                    temp = points[i + 1];
                }

                if ((code & 0b0001) != 0b0000)
                {
                    temp.Y += (points[i].Y - points[i + 1].Y) * (70 - temp.X) /
(points[i].X - points[i + 1].X);
                    temp.X = 70;
                }
                else if ((code & 0b0010) != 0b0000)
                {
                    temp.Y += (points[i].Y - points[i + 1].Y) * (170 - temp.X) /
(points[i].X - points[i + 1].X);
                    temp.X = 170;
                }
                else if ((code & 0b0100) != 0b0000)
                {
                    temp.X += (points[i].X - points[i + 1].X) * (180 - temp.Y) /
(points[i].Y - points[i + 1].Y);
                    temp.Y = 180;
                }
                else if ((code & 0b1000) != 0b0000)
                {
                    temp.X += (points[i].X - points[i + 1].X) * (80 - temp.Y) /
(points[i].Y - points[i + 1].Y);
                    temp.Y = 80;
                }

                if(code == code_a)
                {
                    points[i] = temp;
                    code_a = get_code(points[i]);
                }
                else
                {
                    points[i + 1] = temp;
                    code_b = get_code(points[i + 1]);
                }
            }
            return 1;
        }
  private void pictureBox1_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }
        private void button1_Click(object sender, EventArgs e)
        {
            if (points == null)
                return;
            else if (points.Length == 0)
                return;
            g = Graphics.FromImage(pictureBox1.Image);
            g.Clear(pictureBox1.BackColor);
            pictureBox1.Invalidate();

            int w = pictureBox1.Width;
            int h = pictureBox1.Height;

            for (int i = 0; i < points.Length - 1; i += 2)
            {
```

```
                switch (clip_segment(i))
                {
                    case 1:
                        continue;
                    case 2:
                        outer.Add(i);
                        break;
                }
            }

            g.DrawRectangle(pen_blue, new Rectangle(70, 80, 100, 100));

            for (int i = 0; i < points.Length - 1; i += 2)
            {
                if (outer.Contains(i))
                {
                    continue;
                }
                g.DrawLine(pen_red, points[i], points[i + 1]);
            }

            outer.Clear();
        }


    }
}
```

Вывод

В процессе выполнения данной лабораторной работы был получен практический опыт реализации алгоритма отсечения массива произвольных отрезков заданным прямоугольным окном.