

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра автоматизированной обработки информации (АОИ)

Геометрические фракталы

Отчет о выполнении практической работы
по дисциплине «Компьютерная графика»

Студент гр. 429-3

_____ Бабец А. А.

«__» _____ 20__ г.

Принял:

канд. техн. наук, доцент каф.АОИ

_____ Т.О. Перемитина

«__» _____ 2021 г.

Томск 20__

Введение

Цель практической работы – изучение теоретических основ фрактальной графики, приобретение практических навыков построения геометрических фракталов.

Фрактальная графика основана на автоматической генерации изображений путем математических расчетов. Создание фрактальных изображений основано не в рисовании, а в программировании.

Фрактал – геометрическая фигура, обладающая свойством самоподобия, то есть составленная из нескольких частей, каждая из которых подобна всей фигуре целиком.

В двумерном случае геометрические фракталы получают, задав некоторую ломаную (или поверхность в трехмерном случае), называемую генератором. За один шаг алгоритма каждый из отрезков, составляющих ломаную, заменяют на ломаную - генератор, в соответствующем масштабе. В результате бесконечного числа повторений этой процедуры получается фрактальная кривая, т.е. геометрический фрактал. При видимой сложности полученной кривой, её общий вид задается только формой генератора.

Примерами таких кривых служат кривая Коха, кривая Леви, кривая Минковского, кривая Пеано, кривая Хартера - Хейтуэя. К геометрическим фракталам также относят треугольник Серпинского, множество Кантора, ковер Серпинского, губку Менгера, дерево Пифагора.

Описание используемой среды программирования

Рассмотрим особенности среды программирования Microsoft Visual Studio:

- подсветка синтаксиса и простое автозавершение кода;
- анализ кода при загрузке и непосредственно при вводе;
- понятный и удобный интерфейс программы позволяет легко и быстро привыкнуть к работе и повышает её продуктивность;
- оснащённость инструментами для сборки, средой выполнения тестов, инструментами покрытия и встроенным терминальным окном.

Вышеперечисленные особенности послужили тому, что была выбрана среда разработки Microsoft Visual Studio для написания кода.

Описание метода решения

Граница снежинки, придуманной Гельгом фон Кохом в 1904 году, описывается кривой, составленной из трех одинаковых фракталов. Каждая треть снежинки строится итеративно, начиная с одной из сторон равностороннего треугольника.

Затем каждую из сторон этого треугольника разделим на три равные части, уберем среднюю часть и в середине построим равносторонний треугольник.

На следующем шаге такой же процедуре деления на три равные части и достраивания равностороннего треугольника подвергается каждая из сторон новой фигуры, и так до бесконечности. В результате возникает симметричная, похожая на снежинку, бесконечно изломанная кривая, которая представляет собой самоподобное множество, называемое снежинкой Коха.

Фрагменты листинга

Итерационный метод отрисовки фрактала Коха.

Ссылка: 7

```

static int FractalKoh (PointF p1, PointF p2, PointF p3, int iter)
{
    if (iter > 0) // условие выхода из рекурсии
    {
        // средняя часть отрезка
        var p4 = new PointF((p2.X + 2 * p1.X) / 3, (p2.Y + 2 * p1.Y) / 3);
        var p5 = new PointF((p1.X + 2 * p2.X) / 3, (p1.Y + 2 * p2.Y) / 3);
        // координаты вершины угла
        var ps = new PointF((p2.X + p1.X) / 2, (p2.Y + p1.Y) / 2);
        var pn = new PointF((4 * ps.X - p3.X) / 3, (4 * ps.Y - p3.Y) / 3);
        // рисуем его
        canvas.DrawLine(red_pen, p4, pn);
        canvas.DrawLine(red_pen, p5, pn);
        canvas.DrawLine(red_pen, p4, p5);

        // рекурсивно вызываем функцию нужное число раз
        FractalKoh(p4, pn, p5, iter - 1);
        FractalKoh(pn, p5, p4, iter - 1);
        FractalKoh(p1, p4, new PointF((2 * p1.X + p3.X) / 3, (2 * p1.Y + p3.Y) / 3), iter - 1);
        FractalKoh(p5, p2, new PointF((2 * p2.X + p3.X) / 3, (2 * p2.Y + p3.Y) / 3), iter - 1);
    }
    return iter;
}

```

Вызов метода отрисовки

ссылка: 1

```

private void button1_Click(object sender, EventArgs e)
{
    red_pen = new Pen(Color.Red, 1);
    canvas = CreateGraphics();
    canvas.Clear(Color.White);

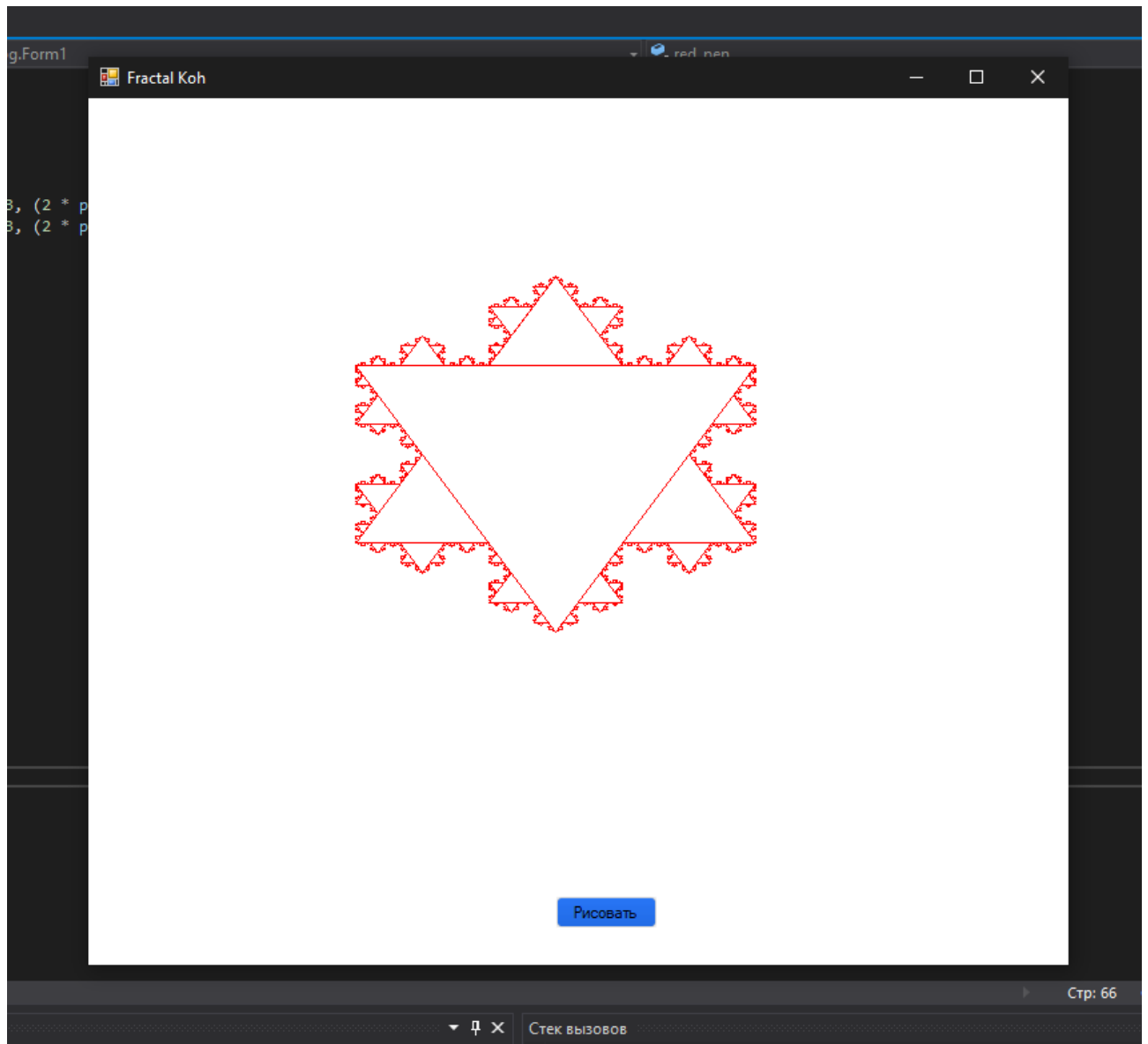
    var point1 = new PointF(200, 200);
    var point2 = new PointF(500, 200);
    var point3 = new PointF(350, 400);

    canvas.DrawLine(red_pen, point1, point2);
    canvas.DrawLine(red_pen, point2, point3);
    canvas.DrawLine(red_pen, point3, point1);

    FractalKoh(point1, point2, point3, 5);
    FractalKoh(point2, point3, point1, 5);
    FractalKoh(point3, point1, point2, 5);
}

```

Скриншот работы программы



Ответы на вопросы

- 1) 2
- 2) 3
- 3) 1,3
- 4) 4
- 5) 1,3
- 6) 1
- 7) 2
- 8) 2
- 9) 2
- 10) 3
- 11) 1
- 12) 1
- 13) 2
- 14) 1,4
- 15) 2,4,7,

Заключение

В ходе практической работы произошло изучение теоретических основ фрактальной графики и приобретение практических навыков построения геометрических фракталов.