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)

Фрактальная графика

Отчет о выполнении лабораторной 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 алгебраических фракталов.

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

Термин «фрактал» (от лат. fractus - раздробленный) впервые ввел в 1975 году математик исследовательского центра IBM Бенуа Мандельброт.

Фракталы можно разделить на несколько видов:

- Геометрические фракталы – строятся на основе исходной фигуры (линии, многоугольника или многогранника) путем ее дробления и выполнения различных преобразований полученных фрагментов.
- Алгебраические фракталы – строятся на основе алгебраических формул.
- Стохастические фракталы – получаются, если в итерационном процессе случайным образом изменять какие-либо параметры.

Фракталы нашли применение в физике (моделирование сложных процессов и материалов), биологии (моделирование популяций, описание сложных ветвящихся структур), технике (фрактальные антенны), экономике. Существуют алгоритмы сжатия изображений с помощью фракталов. В компьютерной графике фракталы используются для построения изображений природных объектов – растений, ландшафтов, поверхности морей и т. д.

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;
- оснащенность 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

Фрактал Мандельброта

Рассмотрим последовательность комплексных чисел:

$$z_{k+1} = z_k^2 + c, k = 0, 1, 2, \dots, z_0 = c$$

Множество точек c, для которого эта последовательность не расходится, называется множеством Мандельброта. Для построения его графической интерпретации нужно определить исходные данные:

- прямоугольное окно C с разрешением $m \times n$ точек;
- значение rmin = 2 – минимальный радиус расходимости множества Мандельброта;
- максимальное число итераций kmax .

Если точка zk вышла за пределы круга радиуса rmin при $k <$ kmax , то процесс вычисления останавливается.

Построение - для каждой точки Cij ∈ C итерационный процесс:

$$x_{k+1} = x_k^2 - y_k^2 + c_x, x_0 = c_x$$

$$y_{k+1} = 2x_k y_k + c_y, y_0 = c_y$$

Где $k = 0,1, \dots, \mathrm{k_{max}}$ и $\sqrt{x_k^2 + y_k^2} \leq \mathrm{r_{min}}$

Формируем многомерный массив (матрицу) M , элементы которой mij ∈ [1; kmax] равны номерам итераций, на которых процесс был остановлен. Далее матрицу можно вывести на экран как растровое изображение, предварительно сопоставив каждому числу из интервала [1; kmax] некоторый цвет.

Фрагменты листинга

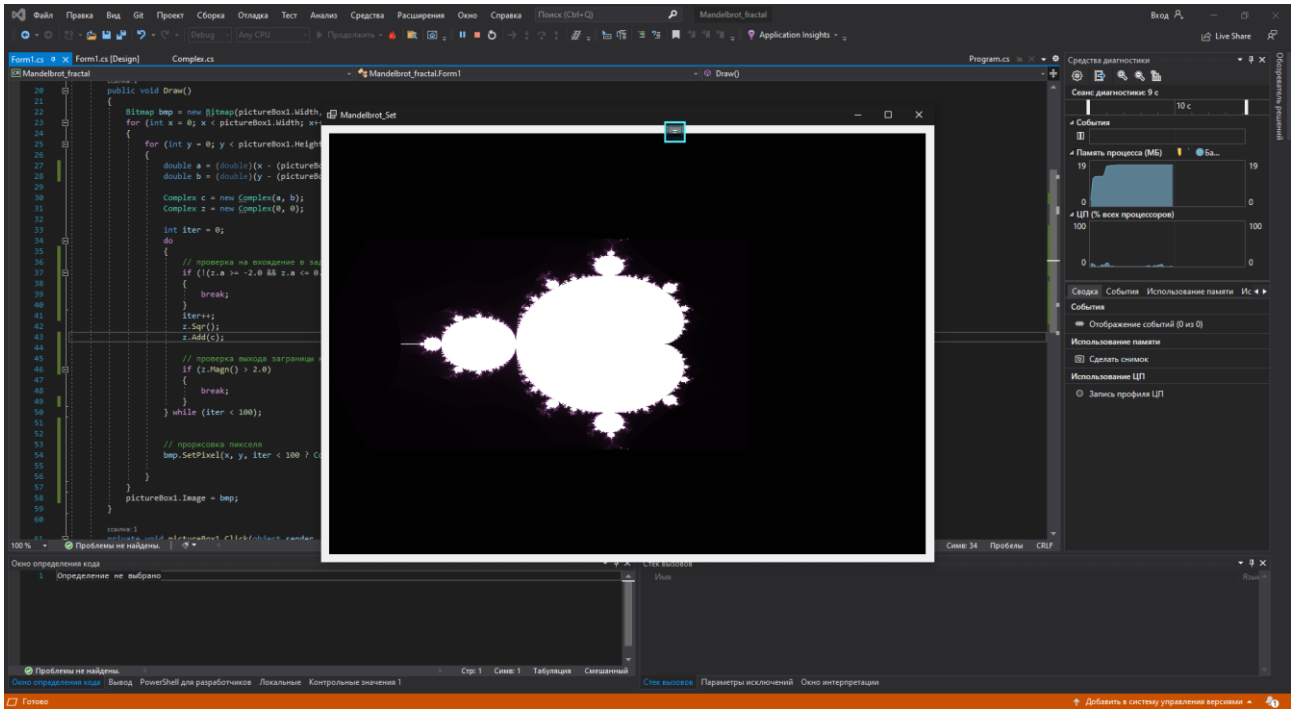
Итерационный метод отрисовки Множества Мандельброта.

```
public void Draw()
{
    Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);
    for (int x = 0; x < pictureBox1.Width; x++)
    {
        for (int y = 0; y < pictureBox1.Height; y++)
        {
            double a = (double)(x - (pictureBox1.Width / 2)) / (double)(pictureBox1.Width / 4);
            double b = (double)(y - (pictureBox1.Height / 2)) / (double)(pictureBox1.Height / 4);

            Complex c = new Complex(a, b);
            Complex z = new Complex(0, 0);

            int iter = 0;
            do
            {
                // проверка на вхождение в заданные границы
                if (!(z.a >= -2.0 && z.a <= 0.8 && z.b >= -1.0 && z.b <= 1.0))
                {
                    break;
                }
                iter++;
                z.Sqr();
                z.Add(c);

                // проверка выхода загранцы круга
                if (z.Magn() > 2.0)
                {
                    break;
                }
            } while (iter < 100);


            // прорисовка пикселя
            bmp.SetPixel(x, y, iter < 100 ? Color.FromArgb(iter*2, iter, iter*2) : Color.FromArgb(255, 255, 255));

        }
    }
    pictureBox1.Image = bmp;
}
```

Для работы с комплексными числами был написан специальный класс

```
class Complex
{
    public double a;
    public double b;

    Ссылок: 2
    public Complex(double a, double b)
    {
        this.a = a;
        this.b = b;
    }

    ссылка: 1
    public void Sqr()
    {
        double tmp = (a * a) - (b * b);
        b = 2.0 * a * b;
        a = tmp;
    }

    ссылка: 1
    public double Magn()
    {
        return Math.Sqrt((a * a) + (b * b));
    }

    ссылка: 1
    public void Add(Complex c)
    {
        a += c.a;
        b += c.b;
    }
}
```

Скриншот работы программы



Заключение

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