МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Компьютерная графика»

Тема: Построение фракталов

Студентка гр. 8383 _____ Максимова А.А.

Студент гр. 8383 _____ Ларин А.

Преподаватель _____ Герасимова Т.В.

Санкт-Петербург

2021

Цель работы

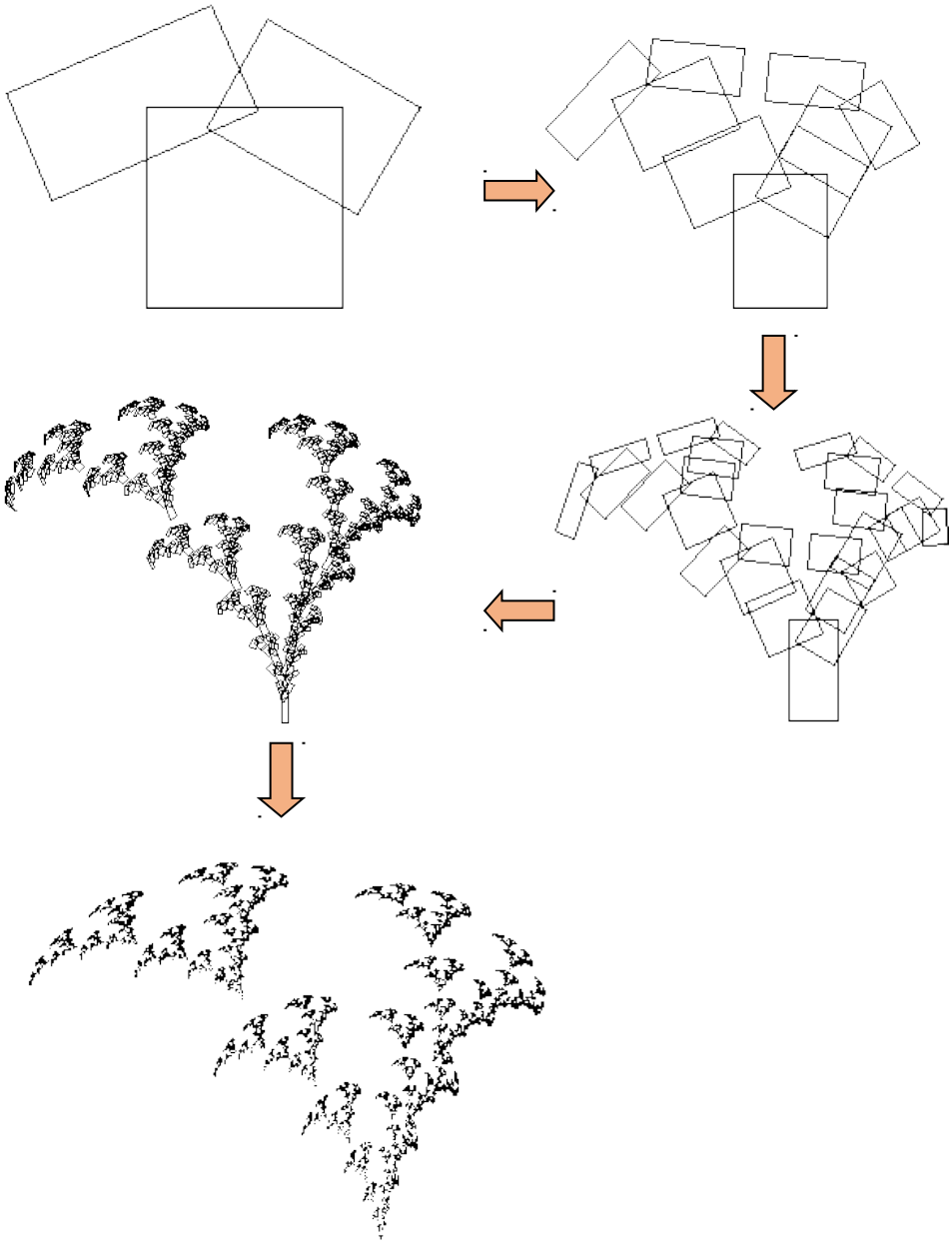
На базе предыдущей лабораторной работы разработать программу, реализующую фрактал по индивидуальному заданию.

Требования и рекомендации к выполнению задания:

- проанализировать полученное задание, выделить информационные объекты и действия;
- разработать программу с использованием требуемых примитивов и атрибутов.

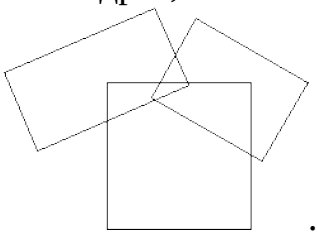
Вариант 54

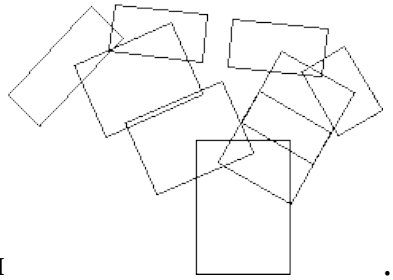
IFS-фракталы “Ветка”



Теоретические сведения

Форма IPS-фракталов состоит из уменьшенных(возможно накладывающихся) копий самих себя. Типичным примером является треугольник Серпинского. Как правило на каждом новом уровне происходит стяжение исходных фигур, что делает их меньше.

Суть фрактала, представленного в задании, состоит в дублировании для каждого уровня его версии меньшего уровня. Так на первом уровне форма представляет из себя квадрат. На втором — квадрат, и на немного выше и по бокам еще два квадрата, образуя «вилку»  .

На третьем - «вилку» и еще две «вилки» по бокам  .
Т.о. рисуется фрактал для любого уровня как три фрактала уровнем меньше. Также правая копия по заданию отдаляется меньше левой.

Текст процедур построения фрактала

Задание выполнено на языке C++ с использованием фреймворка Qt5.

Qt предоставляет возможность использования OpenGL посредством класса `QGLWidget`. От него отнаследован класс `CustomGL`.

`class CustomGL : public QGLWidget`, в котором переопределены методы:

- `void initializeGL();`

  Задает контекст рендеринга. Заключается в установке фона белым
- `void resizeGL(int nWidth, int nHeight);`

  Для корректного масштабирования. Размер окна задается через `glViewport`.
- `void paintGL();`

  Вызов этой функции происходит при рисовании. Она в свою очередь вызывает функцию `scene`, где прописана вся логика рисования.

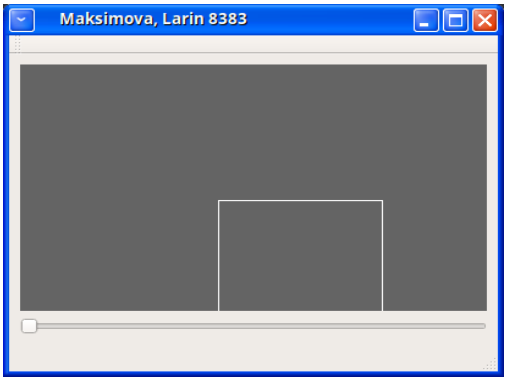
1. Здесь происходит начальная трансформация и масштабирование, чтобы финальный фрактал было видно. Масштабирование зависит от количества итераций. Перемещений происходит т.о. чтобы фрактал «рос» снизу экрана чуть правее центра(т.к. правая ветка меньше левой)

```
void CustomGL::scene()
{
    glPointSize((float)d_size); //задаем размер точки
    glPushMatrix();             //сохраняем текущую матрицу
    glTranslated(.2,-1,0);      //производит перенос объекта, прибавляя к координатам его вершин параметры
    glScaled(.7/std::pow(1.65,nSegments)*10,.9 /std::pow(1.65,nSegments)*10,1); //масштабирование
    drawIter(nSegments);        //рекурсивная функция рисования
    glPopMatrix();              //восстанавливаем текущую матрицу
    return;
}
```

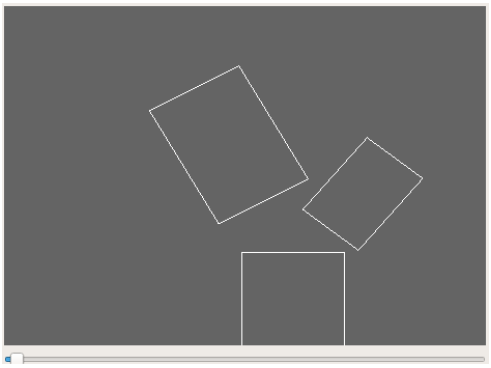
2. Здесь определено рисование фрактала для заданного уровня. На нулевом уровне рисуется простой квадрат. Для уровней n>0 рисуется корневой элемент уровня n-1, а затем еще два элемента уровней n-1 слева и справа. Для этого происходит их поворот и отдаление от корня. Правая ветвь отдаляется меньше левой

```
void CustomGL::drawIter(int n)
{

    if(n==0){
        cnt++;

        glBegin(GL_LINE_STRIP);
        glVertex2d(dw/2,dl/2);
        glVertex2d(-dw/2,dl/2);
        glVertex2d(-dw/2,-dl/2);
        glVertex2d(dw/2,-dl/2);
        glVertex2d(dw/2,dl/2);
        glEnd();
        return;
    }else{
        drawIter(n-1);
        glPushMatrix(); //сохраняем текущую матрицу

        // Left branch
        glRotated(30-(n),0,0,1); //поворот вокруг z
        glTranslated(dw/2,dl*.6 + std::pow(1.8-(0.005*nSegments),n-1)/10 ,0);//производит перенос объекта, прибавляя к координатам его вершин параметры
        glScaled(1,.7,1);  //масштабирование
        drawIter(n-1); //рекурсивный вызов

        glPopMatrix();  //восстанавливаем текущую матрицу
        glPushMatrix(); //сохраняем текущую матрицу

        // Right branch
        glRotated(-40+(n),0,0,1);
        glTranslated(-dw/2,dl*.35 + std::pow(1.7-(0.005*nSegments),n-1)/10 ,0);

        glScaled(.7,.5,1);
        drawIter(n-1);
        glPopMatrix();
    }
}
```

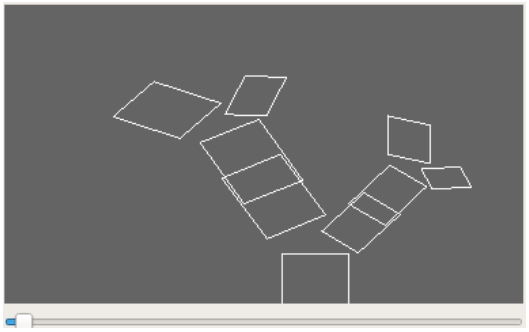
Для масштабирования и изменения количества итераций служит переменная nSegments, которая регулируется бегунком внизу экрана.
Финальный вид фрактала
Для уровня 0:



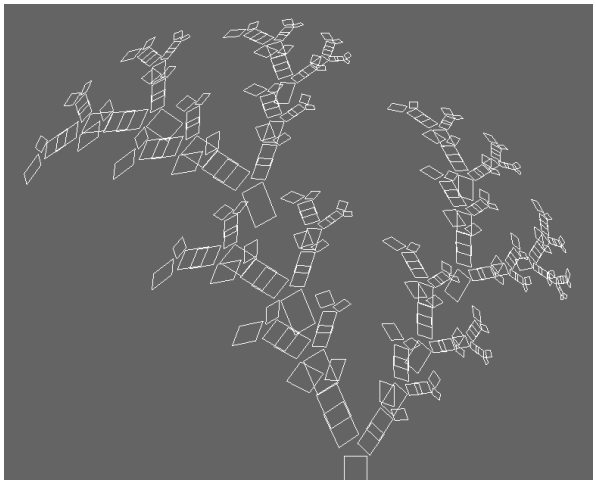
Уровень 1:



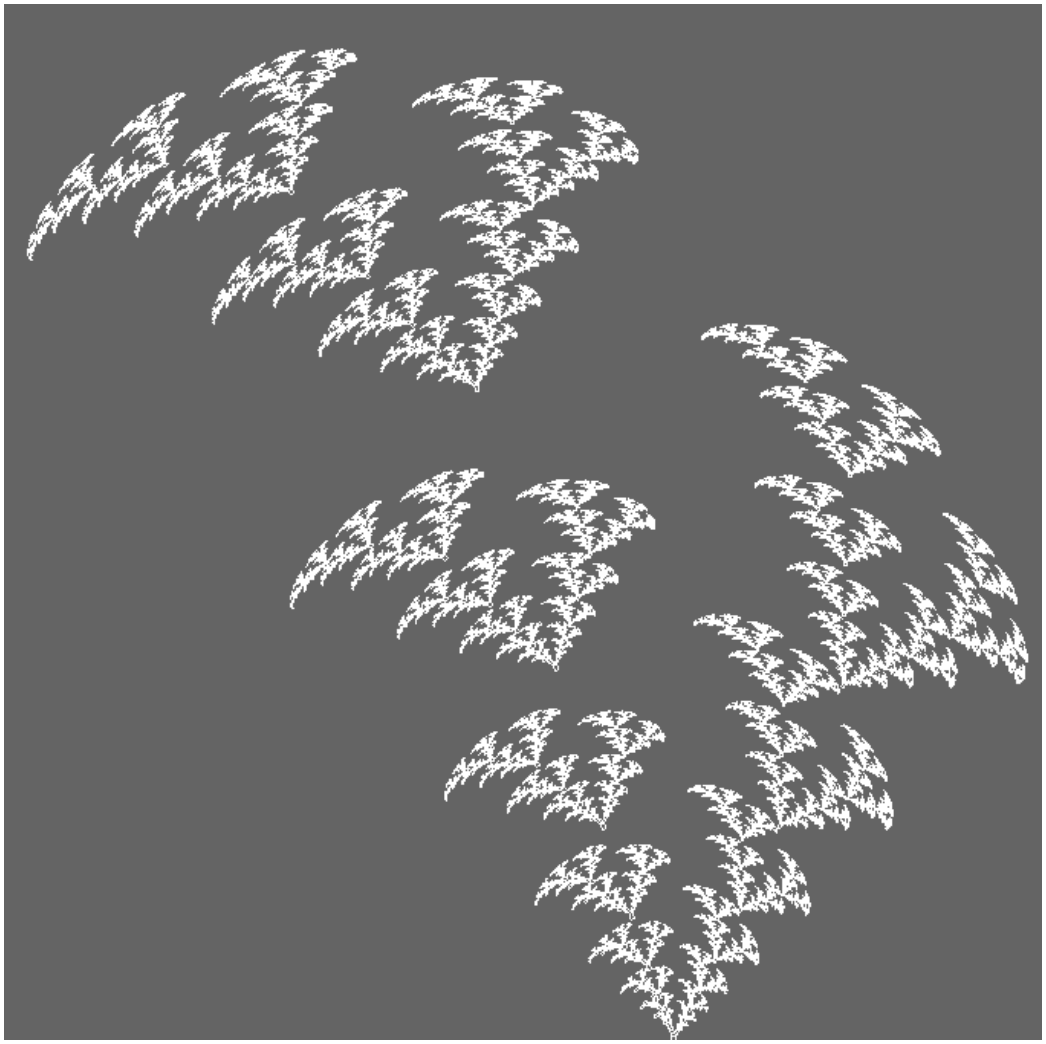
Уровень 2:



Уровень 5:



Уровень 10:



Выводы.

В результате выполнения лабораторной работы была написана программа, реализующая представление IFS-фракталов "Ветка". Программа работает корректно. При выполнении работы были приобретены навыки работы с графической библиотекой OpenGL.