

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Компьютерная графика»
Тема: Построение фракталов

Студент гр. 0304

Максименко Е.М.

Преподаватель

Герасимова Т.В.

Санкт-Петербург

2023

Цель работы.

- исследование способов построения фракталов.

Задание.

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



Выполнение работы.

Работы была выполнена с использованием языка C++ и фреймворка Qt6. Каркасом программы послужила программа из ЛР 1.

Для реализации отрисовки фрактала-дерева в классе GLScene был написан метод *drawTreeRecursive*. Описание данного метода см. в листинге 1.

Листинг 1. Метод *drawTreeRecursive*

```
void GLScene::drawTreeRecursive(GLint depth, GLdouble length, GLdouble
angle, GLdouble width)
{
    /* check recursion depth limit */
    if (depth >= recursionDepth)
        return;
    /* setup brush */
    glLineWidth(width);
    /* draw single line */
    glBegin(GL_LINES);
        glVertex2d(0.f, 0.f);
        glVertex2d(0.f, length);
    glEnd();
    /* draw left subtree */
    glPushMatrix();
        glTranslatef(0.f, length, 0.f);
        glRotatef(angle, 0.f, 0.f, 1.f); // rotate to the angle anti-
clockwise around z axis
        drawTreeRecursive(
            depth + 1,
            length * lengthChangeCoefficient,
            angle * angleChangeCoefficient,
            width * widthChangeCoefficient
        );
    glPopMatrix();
    /* draw right subtree */
    glPushMatrix();
        glTranslatef(0.f, length, 0.f);
        glRotatef(-angle, 0.f, 0.f, 1.f); // rotate to the angle clockwise
around z axis
        drawTreeRecursive(
            depth + 1,
            length * lengthChangeCoefficient,
            -angle * angleChangeCoefficient,
            width * widthChangeCoefficient
        );
    glPopMatrix();
}
```

Данный метод принимает 4 параметра: текущую глубину рекурсии, текущую длину линии, угол наклона новых линий относительно текущей, ширину линии. В начале работы метод проверяет, достигнута ли максимальная глубина рекурсии: если достигнута, то происходит выход из рекурсии. Далее происходит установка ширины линии и ее отрисовка. После отрисовки линии должны быть отрисованы левое и правое поддерево. Отрисовка левого и

правого поддерева идентичны за исключением угла наклона поддерева (левое наклонено на angle градусов против часовой стрелки, правое — на тот же угол по часовой стрелке).

Отрисовка поддерева включает в себя несколько этапов. Первый этап состоит в том, чтобы сохранить текущую матрицу в стек матриц преобразования — в будущем понадобится ее восстановить. Вторым этапом является преобразование объектно-видовой матрицы: координатные оси смещаются по оси y вверх на длину линии и поворачиваются на угол, который соответствует углу поворота следующей линии относительно текущей линии (параметр angle). Данные преобразования позволяют отрисовать следующую линию без дополнительных расчетов координат начала и конца линии: линия будет задаваться точками $(0, 0)$ и $(0, \text{length})$ в новой системе координат. Третьим этапом является вызов рекурсивной функции отрисовки дерева для глубины рекурсии $\text{depth} + 1$, для измененных длины, ширины и наклона линии. Наконец, на четвертом этапе происходит восстановление исходной матрицы преобразования объектно-видовой матрицы (нужно для отрисовки второго поддерева).

В интерфейс программы были добавлены настройки различных параметров фрактала: максимальной глубины рекурсии, длины и ширины линии, наклона поддерева, а также коэффициентов изменения последних трех параметров.

Максимальная глубина рекурсии была ограничена значением 16, так как дальнейшее увеличение глубины рекурсии вызывало проблемы с производительностью программы.

Тестирование.

Программа была протестирована для различных настроек отрисовки фрактала. Результаты тестирования см. на рис. 1 — 4.

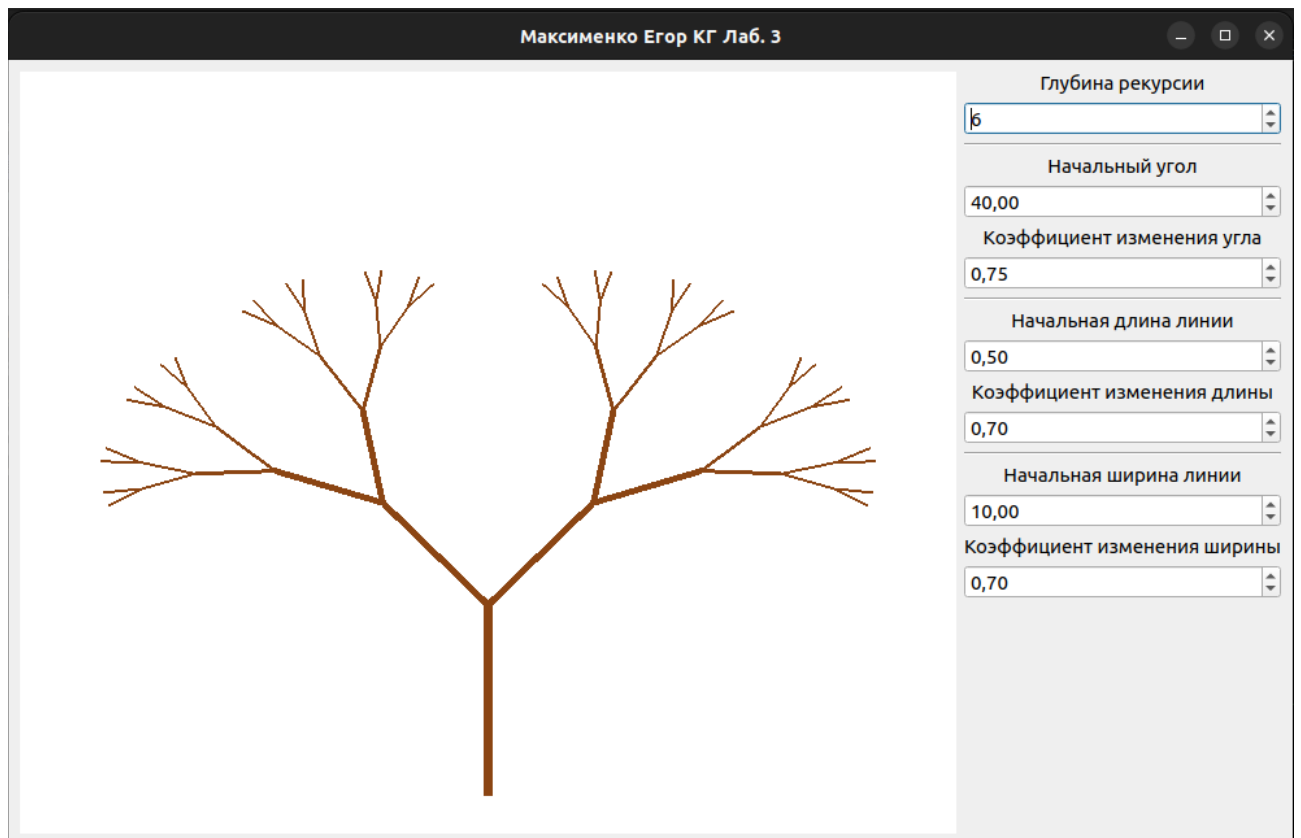


Рисунок 1. Тестирование программы с глубиной рекурсии 6

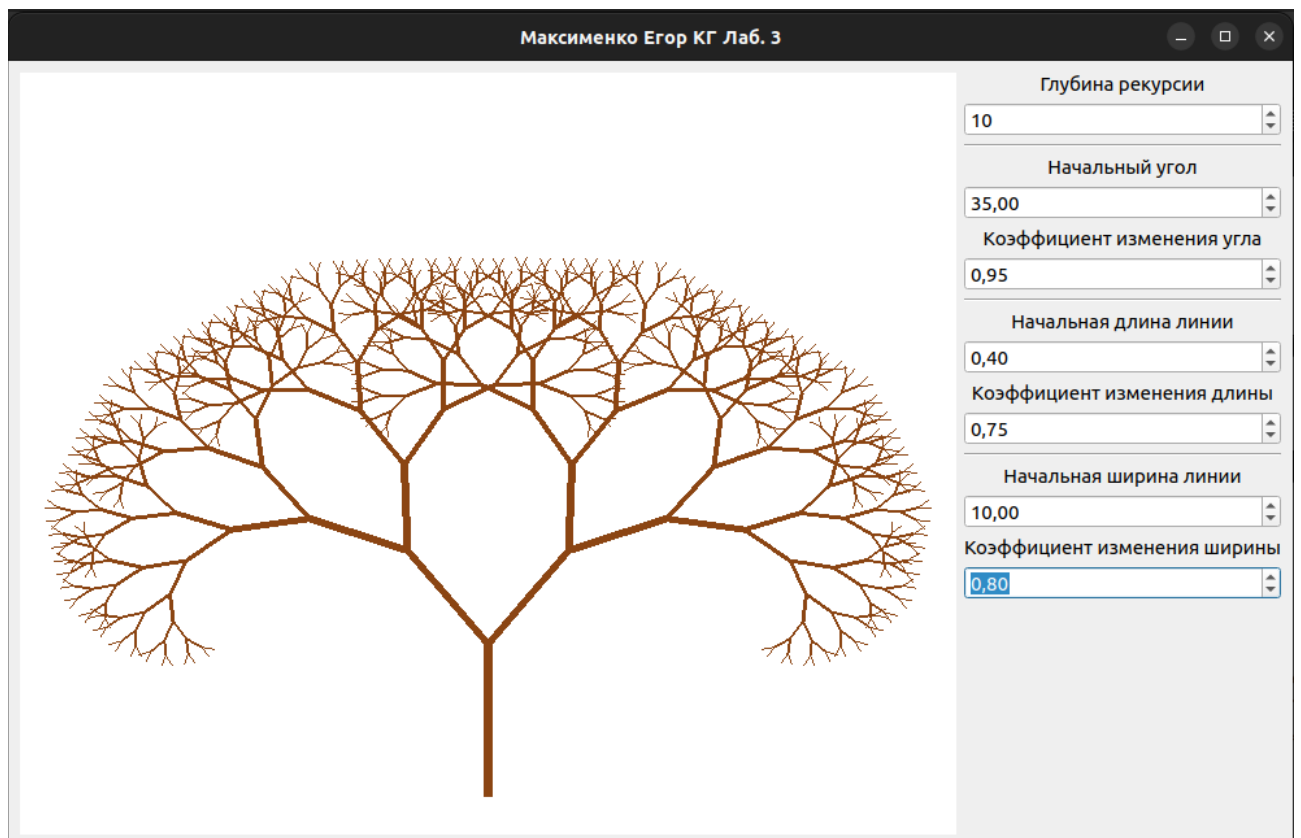


Рисунок 2. Тестирование программы с глубиной рекурсии 10

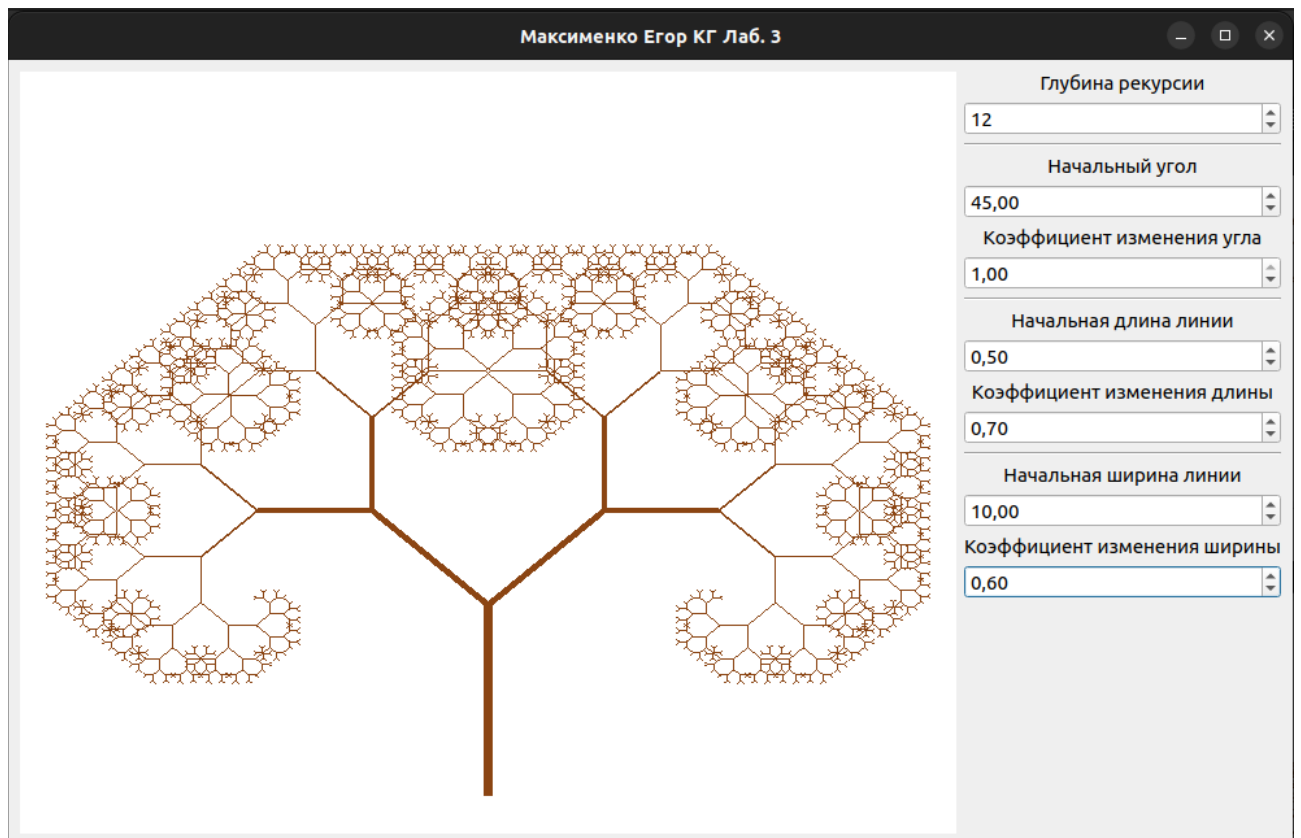


Рисунок 3. Тестирование программы с глубиной рекурсии 12

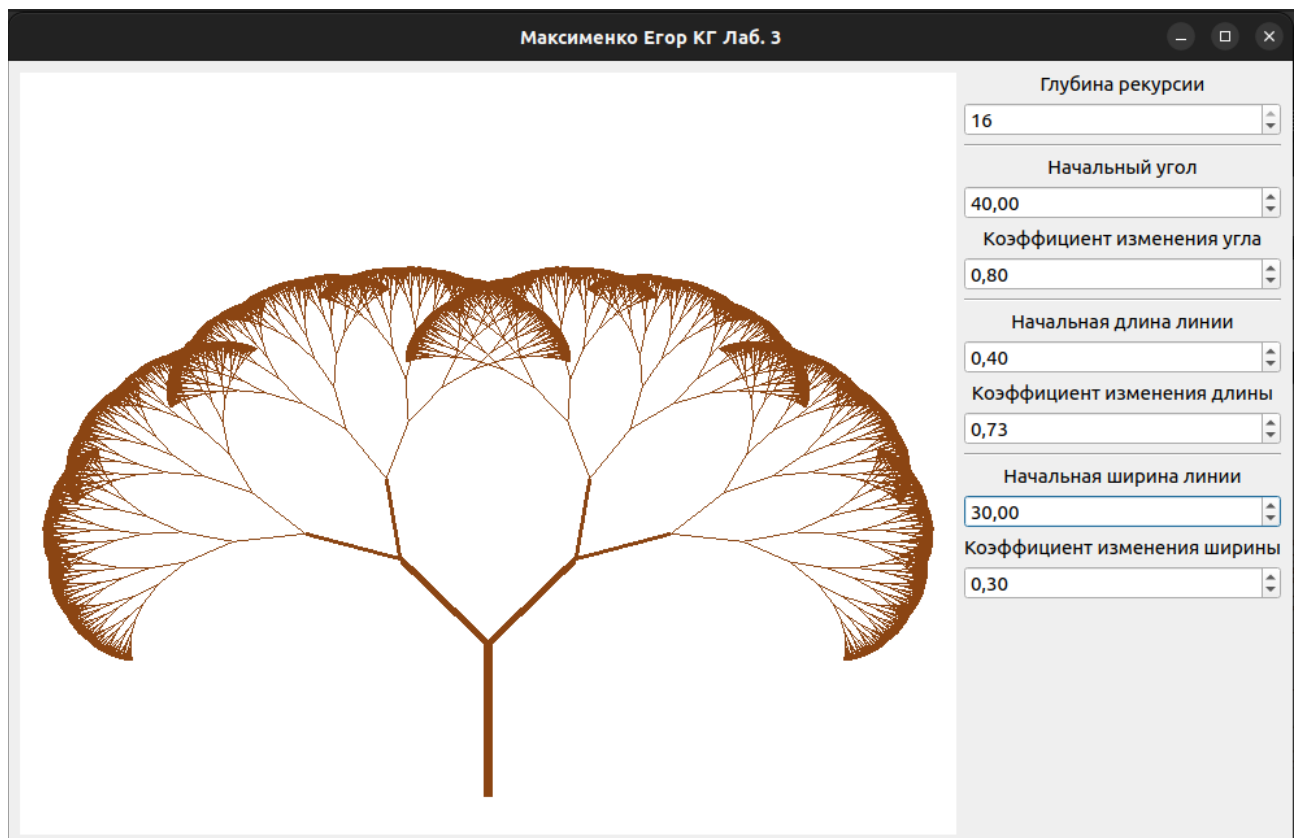


Рисунок 4. Тестирование программы с глубиной рекурсии 16

Выводы.

В ходе работы была разработана программа с использованием языка C++ и фреймворка Qt6, реализующая построение заданного фрактала (дерево). При построении фрактала были использованы преобразования объектно-видовой модели OpenGL.