

Лабораторная работа 3 по курсу компьютерной графики: Основы построения фотореалистичных изображений

Выполнил студент группы М8О-306Б-18 МАИ Семенов Илья.

Условие:

Используя результаты ЛР 2 аппроксимировать заданное тело выпуклым многогранником. Точность аппроксимации задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника света. Параметры освещения и отражающие свойства материала задаются пользователем.

Вариант 20:

Слой эллипсоида

Материалы:

Лекции по компьютерной графике

Методические указания к лабораторным работам по компьютерной графике

Документация Qt — doc.qt.io

learnopengl.com

Описание программы:

Большая часть кода программы взята из предыдущей работы и доработана. Теперь все преобразователи-обработчики работают с полигонами фигуры и наследуются от абстрактного класса. Фигура больше не имеет метода для отрисовки, а может только выполнить расчет своих полигонов. В программу добавлен класс, представляющий собой пайплайн, соединяющий между собой несколько обработчиков, которые вызываются последовательно и принимают результат работы предыдущего в качестве входных данных. На выходе получается набор объектов, хранящих двумерный многоугольник и цвет, которым он должен быть залит. Этот набор далее отрисовывается через QPainter. Для реализации освещения добавлен отдельный обработчик, рассчитывающий цвет полигона, используя для этого нормальный вектор полигона, вектор падения света и вектор, направленный от наблюдателя на центр полигона. Реализована простая модель освещения, рассчитываются ambient, diffuse и specular компоненты света. Источник имеет белый свет, а пользователю предлагается выбрать материал фигуры из заранее заготовленного множества. Материал фигуры влияет на то, как она взаимодействует со светом и представляет из себя набор коэффициентов. Материалы были найдены на сайте learnopengl.com.

Структура программы

Программа состоит из файла `main.cpp`, абстрактного класса `Figure` (а так же класса `Ellip` — единственного наследника), а так же из классов `Transformer`, `InvisibleDeleter`, `Rasterizer`, `Projecter` — классов, для преобразования точек или полигонов фигуры (не объединенные, однако, в общую иерархию). Так же в программе присутствуют классы `ParamHandler` и `GraphicWidget` — их функционал схож с функционалом аналогичных классов предыдущей работы, но несколько доработан.

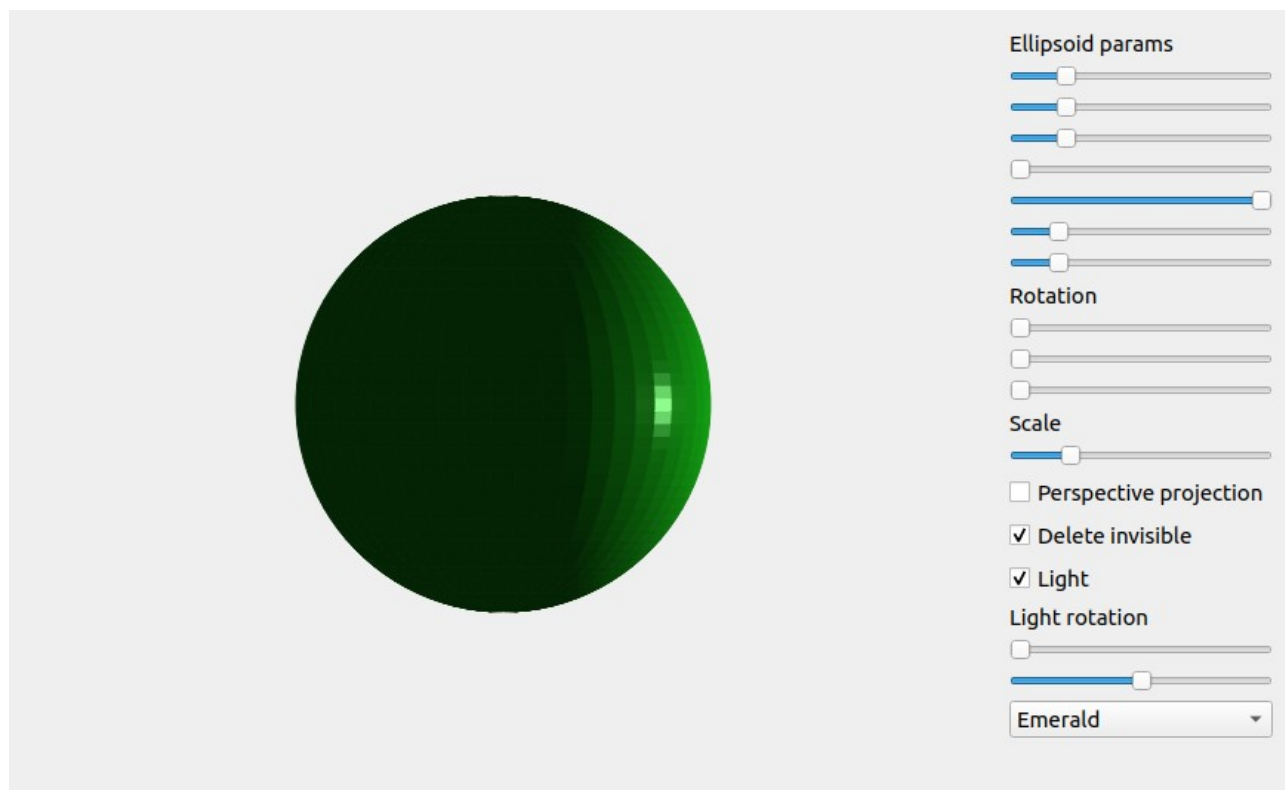
Описание файлов:

- `main.cpp` — содержит точку входа, в которой создается `QApplication` и `MainWindow`, у последнего вызывается метод `show`, после чего приложение входит в `event loop`.
- Класс `MainWindow` — основное окно. В конструкторе создает `ui` на основе формы и соединяет сигнал `ParamHandler::SendParameters` со слотом `GraphicWidget::SetParameters`, для соответствующих объектов, созданных во время создания `ui` и являющихся его детьми. После чего вызывает функцию `Setup` для обработчика параметров

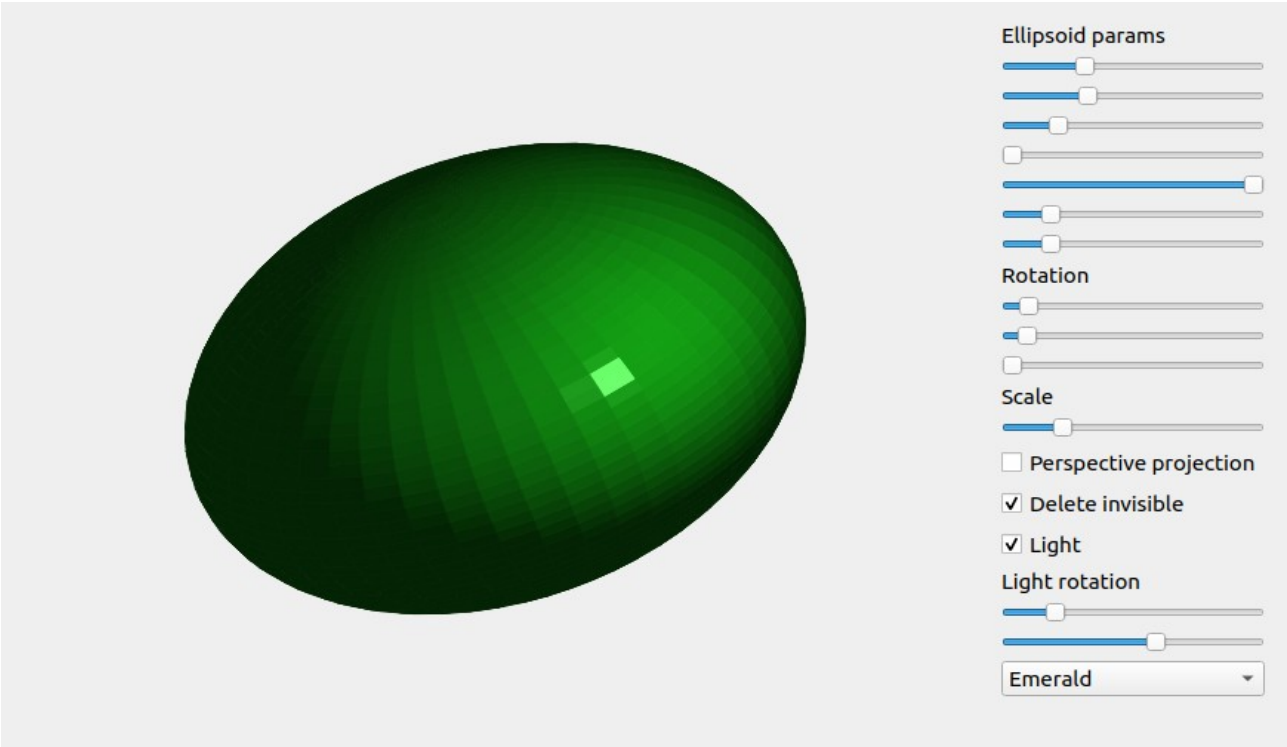
- Класс ParamHandler — предназначен для пересылки параметров в GraphicWidget. Добавлен шаблонный класс-обертка для внутренних виджетов, это позволило уменьшить дублирование кода. При изменении любого из них, вызывается событие, отсылающее параметры адресату(в данном случае это GraphicWidget). Необходимо вызвать функцию Setup, чтобы соединить внутренние виджеты со слотом UpdateParameters и добавить указатели на них в контейнер. Это действие не может быть выполнено в конструкторе, так как на момент вызова конструктора, дочерние виджеты еще не инициализированы.
- Класс GraphicWidget — занимается отрисовкой(с помощью QPainter) фигуры. Управление осуществляется с помощью виджетов в ParamHandler. Класс занимается по большей части настройкой параметров у преобразователей и фигуры, после чего запускает пайплайн и отображает на экране его результат.
- Класс InvisibleDeleter — реализует алгоритм Джонсона удаления невидимых линий. Алгоритм работает в мировом пространстве и корректно удаляет невидимые полигоны для выпуклых многогранников(эллипсоид является выпуклым, как и его слой).
- Класс Transformer содержит несколько четырехмерных матриц, которые умножаются на точки фигуры, тем самым изменяя масштаб, поворачивая и двигая фигуру в пространстве.
- Класс Projecter искажает фигуру таким образом, чтобы получить перспективную проекцию при окончательном отображении точек фигуры на двумерную плоскость.
- Класс Rasterizer — преобразует фигуру в набор двумерных многоугольников, которые представляют собой полигоны фигуры, спроецированные на плоскость xOz. Эти точки далее отрисовываются на экране с помощью QPainter.
- Класс Ellipsoid — предназначен для расчета полигонов слоя эллипсоида.
- Класс LightProcessor — реализует простую модель освещения, рассчитывая цвет каждого полигона, основываясь на материале, направлении света и направлении взгляда пользователя.

Демонстрация работы программы

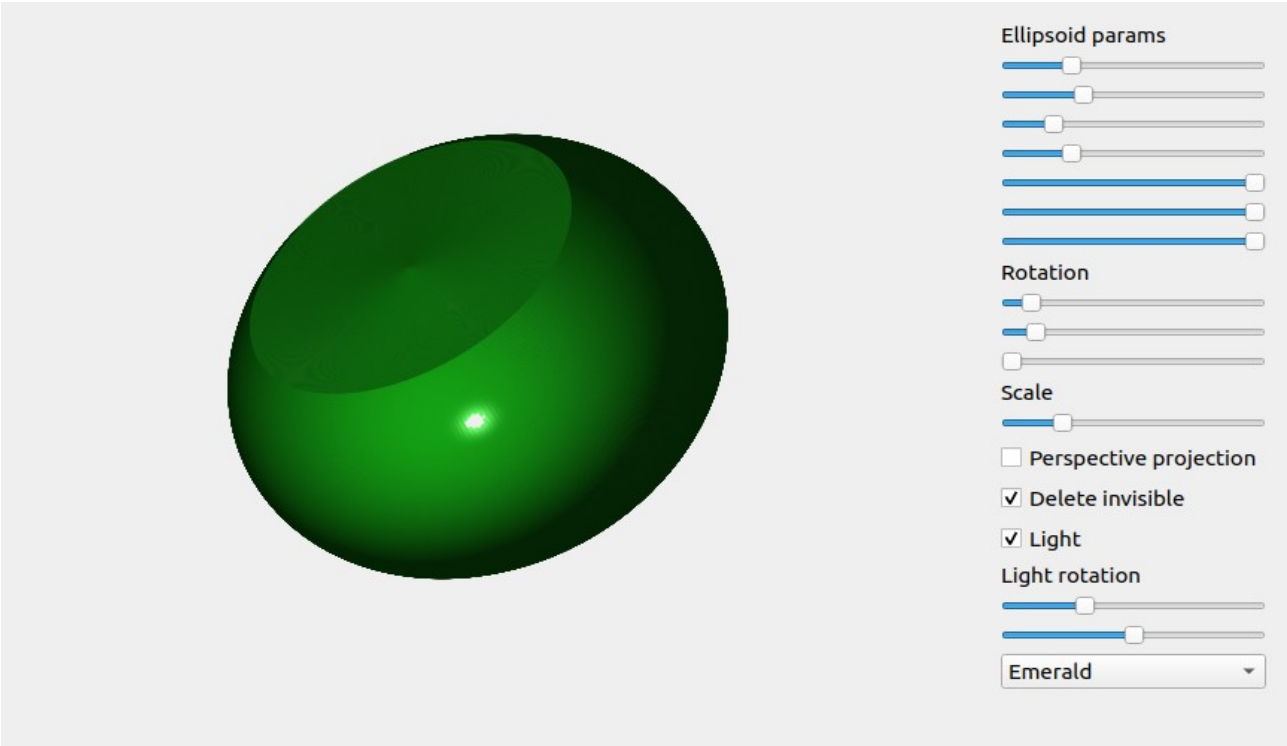
Вид при запуске:



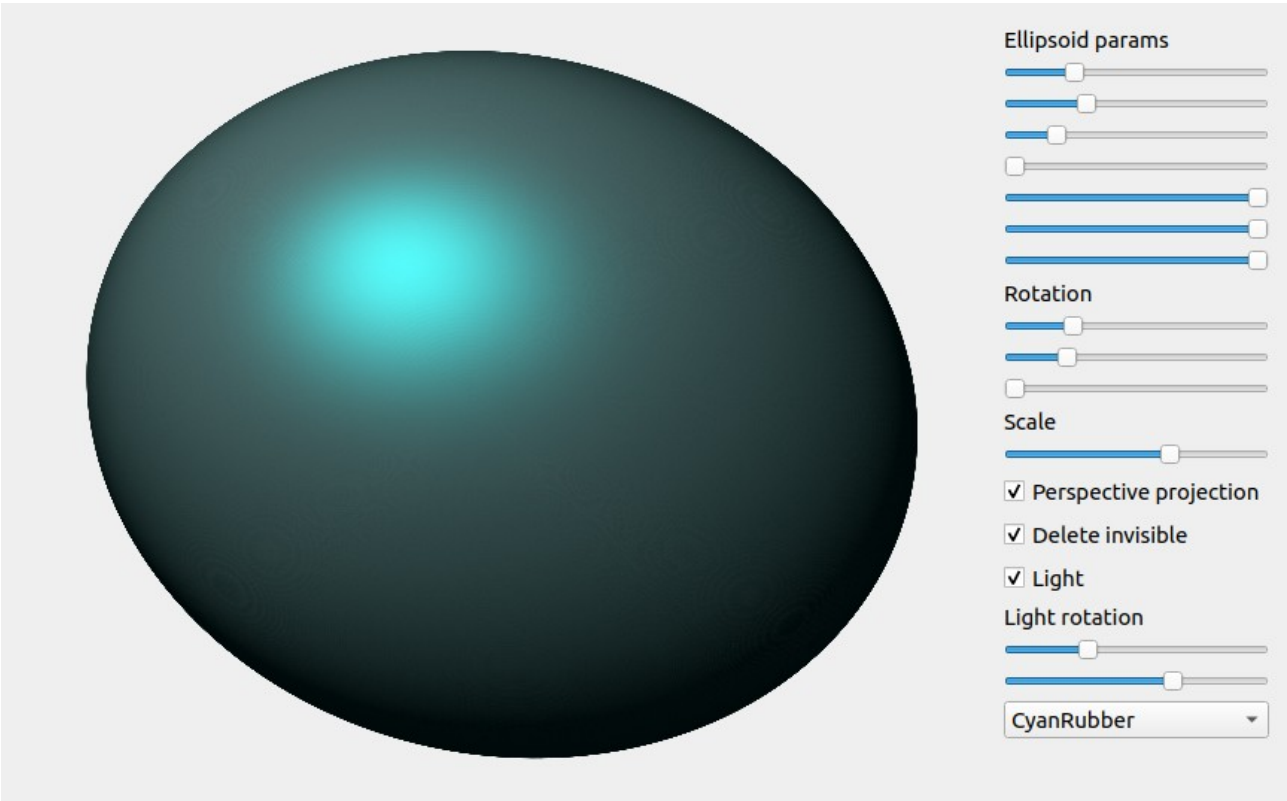
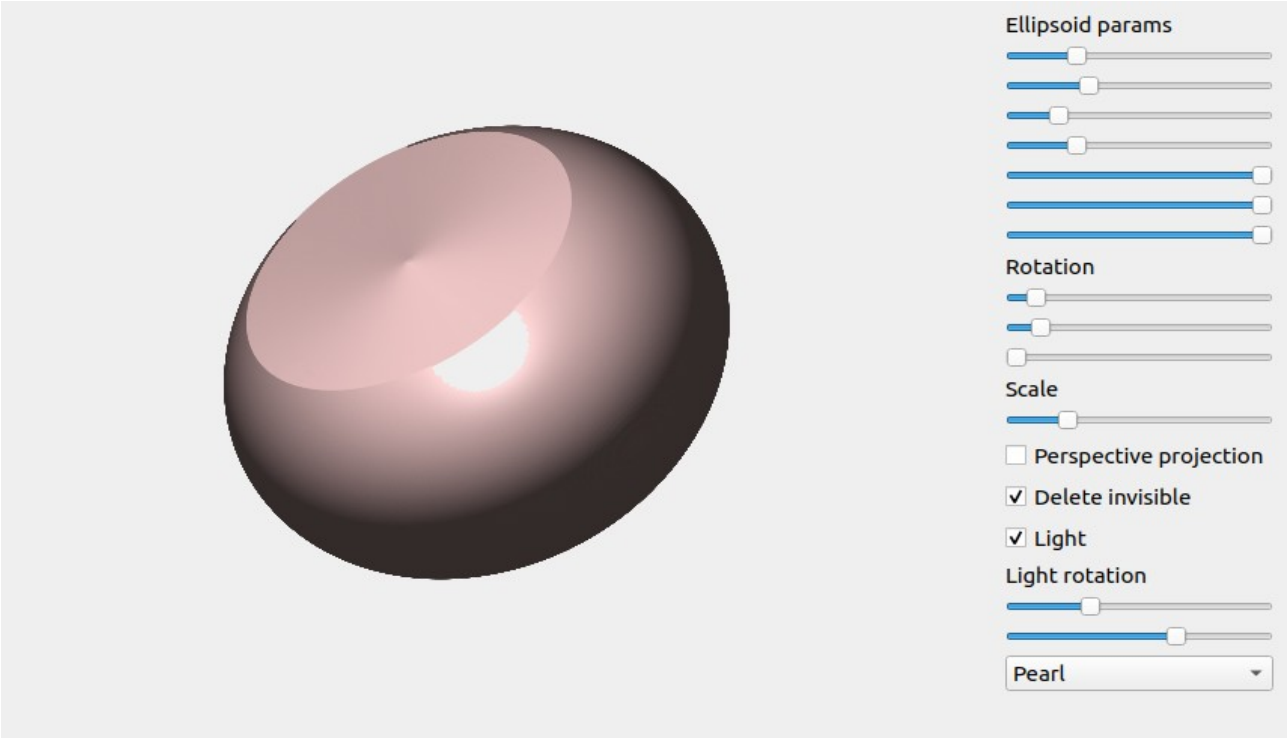
После изменения параметров фигуры и вращения:



Увеличение количества полигонов и срез верхнего слоя



Смена положения света и изменение материала:



Вывод

Геометрическое тело было аппроксимировано с помощью функций построения и заливки многоугольников конкретным цветом(данный функционал предоставлен объектом класса QPainter). Код для остальных преобразований и вычислений был написан вручную, что положительно влияет на понимание предмета.