Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет: “Информационные технологии и прикладная математика”
Кафедра: 806 “Вычислительная математика и программирование”

Лабораторная работа №2.

Каркасная визуализация выпуклого многогранника.
Удаление невидимых линий.

Студент: Ефимов А. В.
Группа: М8О-307Б-18
Преподаватель: Филиппов Г. С.
Оценка: ____________
Дата: ____________
Подпись: ____________

Москва, 2020

1. Постановка задачи

Разработать формат представления многогранника и процедуру его каркасной отрисовки в ортографической изометрической проекциях. Обеспечить удаление невидимых линий и возможность пространственных поворотов имасштабирования многогранника. Обеспечить автоматическое центрирование и изменение размеров изображения приизменении размеров окна.

Вариант: 10 – гранная прямая правильная усеченная пирамида.

2. Решение задачи

Задача осложнена тремя пунктами:

a. Необходимо либо составить все точки, по которым можно построить фигуру, либо найти (составить) метод их подсчета;
b. Полученную фигуру необходимо вращать и масштабировать в пространстве, а также проектировать 3D координаты в доступные для инструмента рисования 2D координаты;
c. У сторон фигуры необходимо проверять видимости и избегать рисования невидимых сторон.

Пункт *(a)* решается более-менее просто. Двумя точками будут центры верхних и нижних точек. Задав радиусы верхних и нижних окружностей, их точки можно высчитать по формулам полярных координат:

$$\begin{cases} x_i = r \cdot \mathbf{cos}(\frac{2\pi}{N} \cdot i) \\ y_i = r \cdot \mathbf{sin}(\frac{2\pi}{N} \cdot i) \end{cases}$$

где i – номер вершины, r – радиус окружности, в которой выделяются основания пирамиды. Число вершин будет равно числу сторон N.

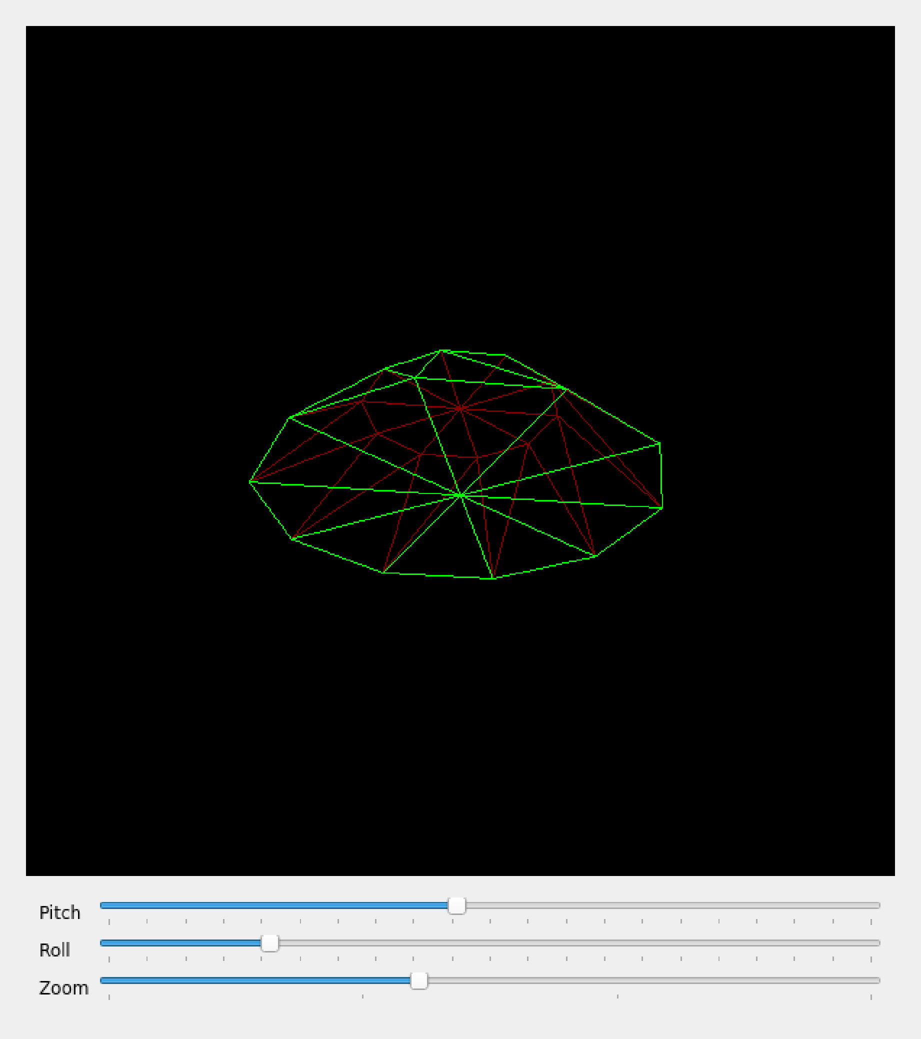
Для вращения и масштабирования фигуры достаточно для каждой точки применять матрицы преобразований из курса линейной алгебры.
Матрица проекции приводится не будет из-за специфичности в её имплементации для разных инструментов.

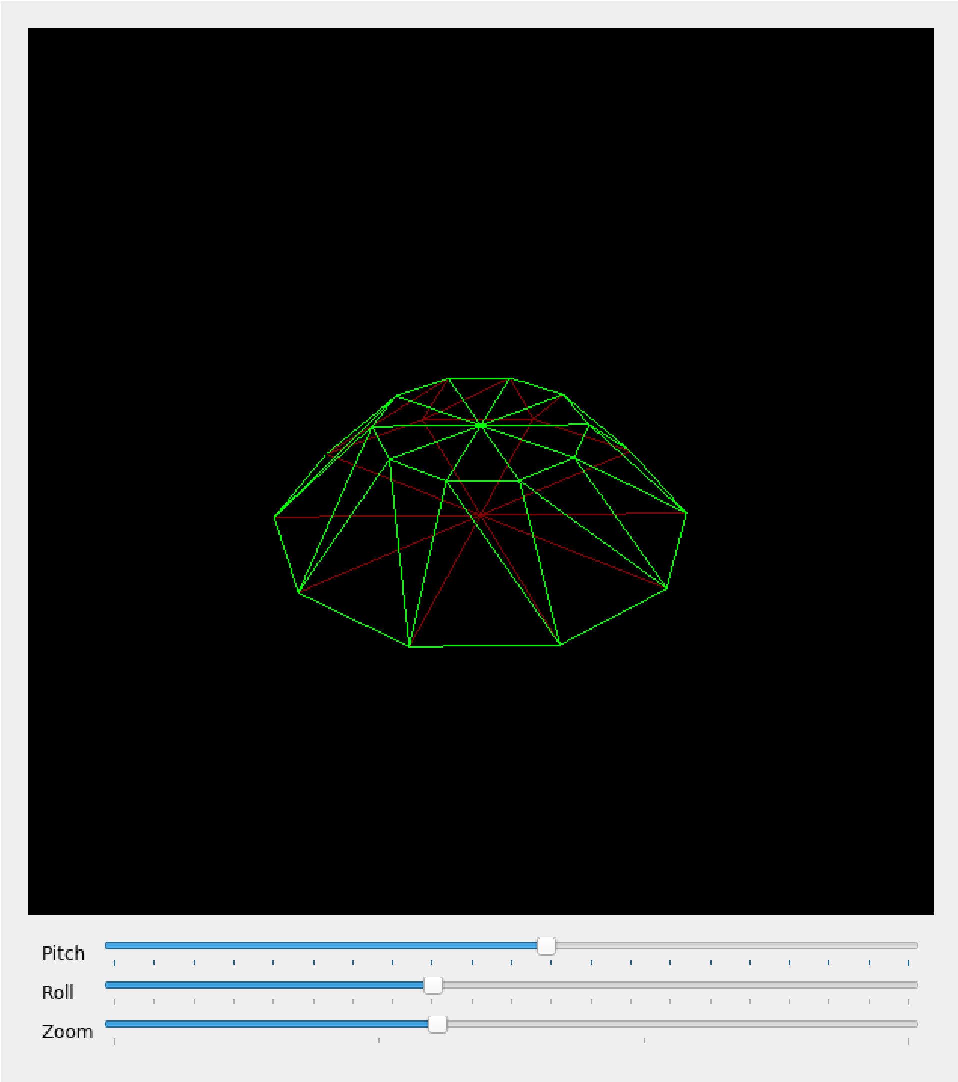
Пункт (c), спорно, является сложным из них. Для этого необходимо в пространстве определить направление вектора камеры. В OpenGL это будет простым вектором в направлении оси Z (если считается, что камера не поворачивается).
Если каждая из вершин (высчитанных в пункте (a), соединены так, что они образуют примитивную поверхность (например, треугольник), то через стороны этого примитива можно посчитать нормаль к нему.
Если нормаль исходит из поверхности, то посчитанный из определения скалярного произведения косинус должен быть меньше нуля (или больше, если входит в поверхность), чтобы поверхность можно было считать видимой.

3. Программа

Программа составлена на смеси Qt и OpenGL, причем никаких инструментов OpenGL, кроме самого рисования поверхностей (а также их рисования в виде линий glPolygonMode, что опционально) не использовано, все необходимые матрицы реализованы. Подсчет и проверка нормалей происходит классах поверхностей.

При запуске открывается отрисованная фигура и три ползуна, отвечающих за поворот и вращение фигуры, а также за приближение (управлять можно и мышкой). Красными линиями показаны невидимые линии, зелеными – видимые.





4. Выводы

На первый взгляд эта лабораторная может показаться сложной, но разбив ее на шаги, все становится предельно просто.
Для небольшого числа сторон метод подсчета нормалей будет достаточно, но для более сложных фигур, таких как реальные модели, нормали следует хранить вместе с вершинами и трансформировать соответственно, поскольку каждый такой подсчет может занимать со значительной сложностью.