

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

**Лабораторная работа №1
по курсу “Компьютерная графика”**

Основы 3D графики

Выполнил: А. Н. Гуляев
Группа: М8О-311Б-23
Преподаватель: В. Д. Бахарев

Москва, 2026

Условие

Целью работы является знакомство с 3D-графикой, а именно с построением простых 3D-объектов, проекцией на 2D-плоскость, матрицами перспективы и ортографической проекции, а также аффинными преобразованиями.

Выполнен вариант 5: сгенерировать тор (~200 вершин, двойной цикл). Цвет меняется через uniform/push constant по времени (например, $\sin(\text{time})$ для RGB-компонент). Перспективная проекция. Анимация: тор вращается вокруг своей оси. Добавить UI элемент для изменения угла обзора и скорости вращения

Метод решения

Для реализации задания использован Vulkan API с библиотеками GLFW (управление окном/вводом) и ImGui (интерфейс пользователя). Основная логика рендеринга построена на матрицах преобразований Model-View-Projection, которые применяются в вершинном шейдере для проекции 3D-координат на 2D-экран.

Тор

Тор сгенерирован параметрически с использованием торических координат. Для дискретизации поверхности использованы два цикла по углам:

- u — азимутальный угол по большому кругу (полный оборот вокруг оси Z)
- v — азимутальный угол по малому кругу (полный оборот вокруг трубки)

Координаты вершин вычисляются по формулам:

$$x = (R + r \cdot \cos(v)) \cdot \cos(u)$$

$$y = (R + r \cdot \cos(v)) \cdot \sin(u)$$

$$z = r \cdot \sin(v)$$

где R — радиус большого круга (major radius), r — радиус трубки (minor radius). Цвет вершин устанавливается в белый (будет модифицироваться временем в шейдере). Для аппроксимации гладкой поверхности используется 20 сегментов по большому кругу и 10 сегментов по малому кругу. Тор состоит из 20 колец по 10 вершин в каждом, итого 200 вершин. Индексы треугольников генерируются для каждого квадрата сетки (два треугольника на каждый квад), всего 1200 индексов (400 треугольников).

Вершины и индексы треугольников записаны в VkBuffer (vertex/index buffer).

Матрицы преобразований

Матрица модели (Model Matrix) описывает положение, ориентацию и размер объекта в мировом пространстве. Она строится как композиция элементарных аффинных преобразований:

$$M = S \cdot R_x \cdot R_y \cdot R_z \cdot T$$

где:

- T — матрица перемещения (Translation): сдвигает объект на вектор pos
- R_x, R_y, R_z — матрицы поворота вокруг осей X, Y, Z на углы rot_x, rot_y, rot_z в радианах
- S — матрица масштабирования: $S = \text{diag}(scale_x, scale_y, scale_z)$, где $scale_y = 1.0 + \sin(time)$ создаёт пульсацию по Y .

Порядок перемножения справа налево обусловлен тем, что матрицы применяются сначала к локальным координатам вершин, затем результат трансформируется дальше: сначала поворот (R), потом перемещение (T), затем масштаб (S). Это стандарт для right-handed систем координат в графике.

Матрица вида (View Matrix) преобразует мировые координаты в камерную систему.

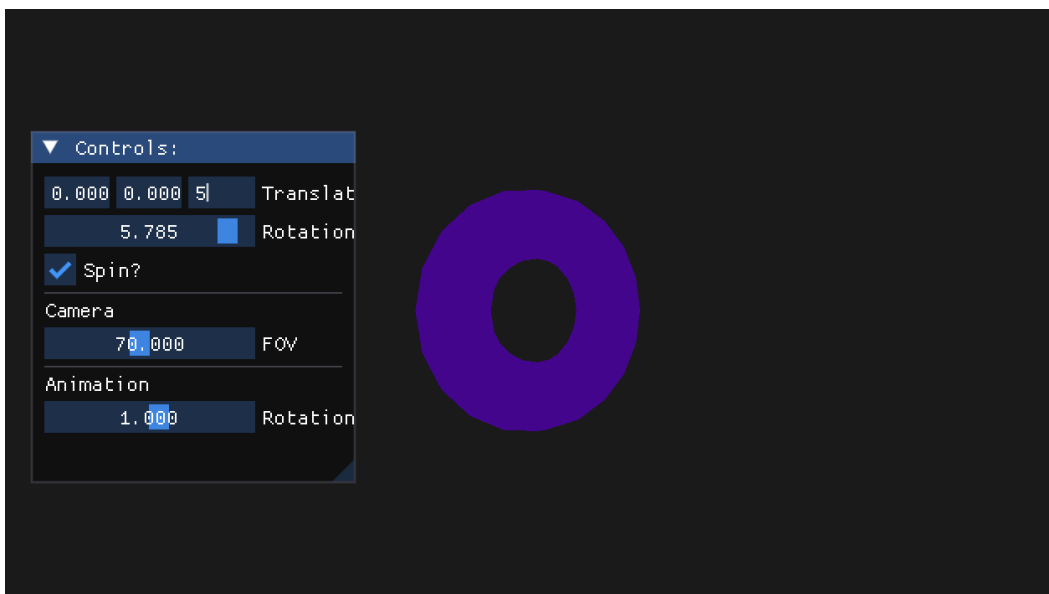
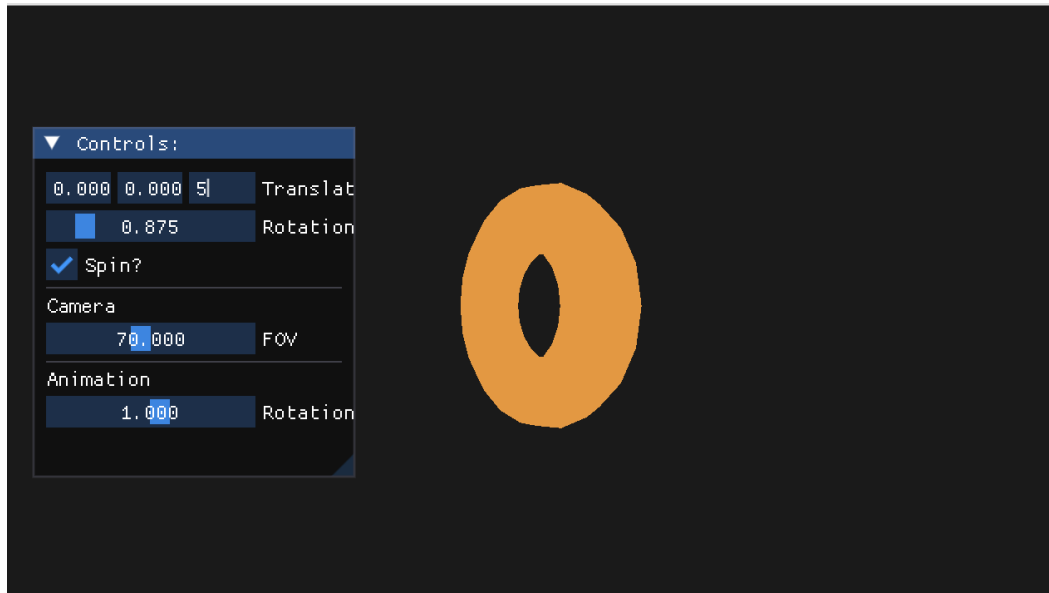
Итоговая матрица имеет вид: $MVP = P \cdot V \cdot M$.

Анимация и UI

Анимация пульсации реализована через глобальное время $time$ (обновляется в $update()$), передаваемое в $uniform$. Масштаб: $scale_y = 1.0 + \sin(time)$.

Результаты

На рисунках ниже приведен вращающийся тор, меняющий цвет в зависимости от времени:



Выводы

Я научился создавать простой 3D-объект (сфера), строить матрицы модели, вида и проекции и настраивать Vulkan pipeline. Понял принцип анимации масштабирования сферы по синусоиде и интерактивного управления камерой. Понял различие перспективной проекции от ортогографической.