

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Компьютерная графика»

Студент: М.Ю.Курносов

Преподаватель:

Группа: М8О-306Б-23

Дата:

Оценка:

Подпись:

## **Условие**

Задача: Требуется используя Vulkan, GLFW и ImGUI, создать программу. Программа должна работать в реальном времени, с возможностью динамической смены проекции и трансформаций объектов. Все объекты должны корректно отрисовываться с учетом проекции и иметь возможность взаимодействия с пользователем.

Вы должны сделать задание своего варианта полностью по условию на базовую оценку 4. Любой невыполненный пункт из варианта снижает оценку на пол-балла. Чтобы повысить оценку, нужно сделать два дополнительных задания из списка ниже после вариантов. Каждое выполненное дополнительное задание даёт пол-балла.

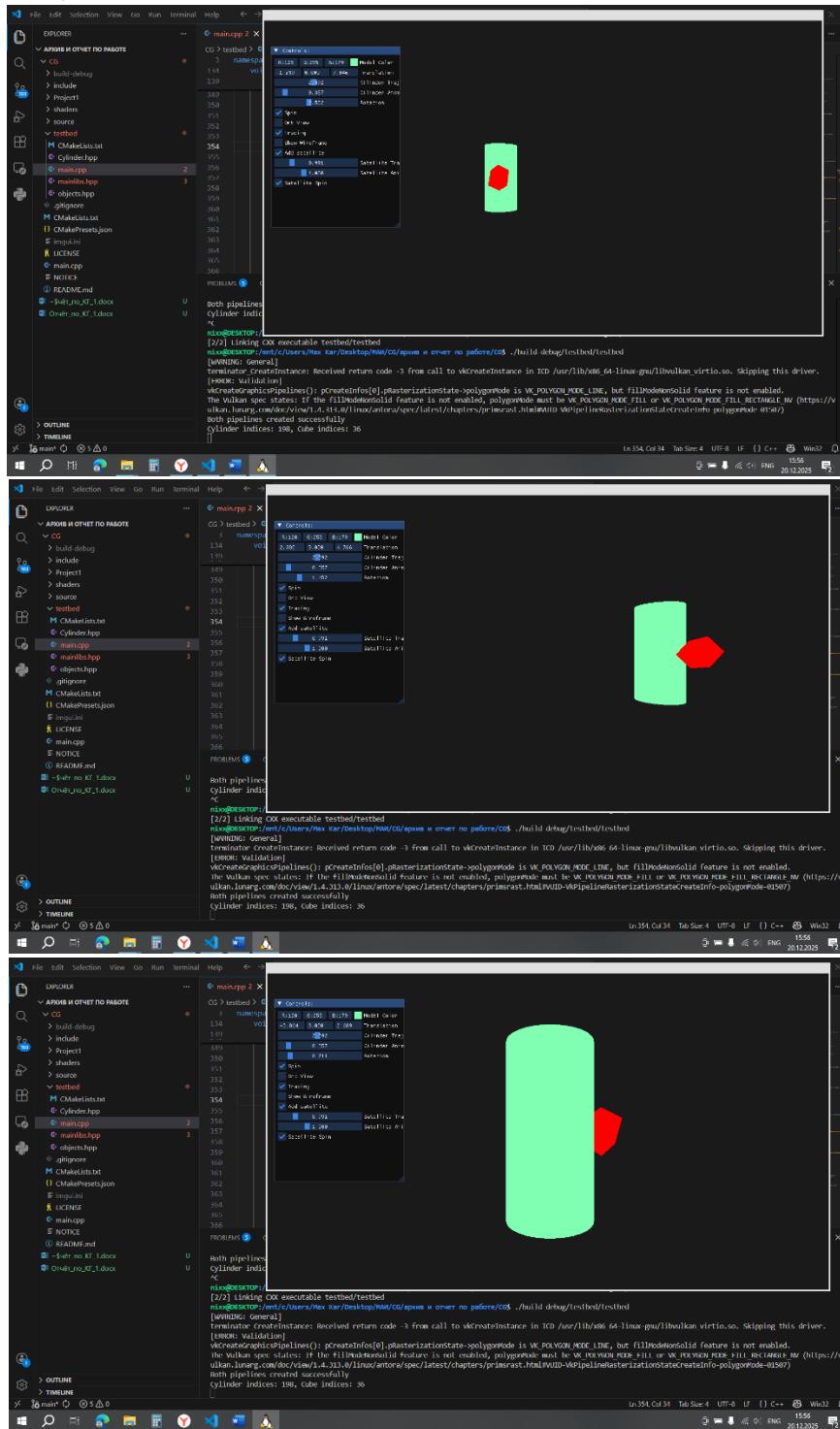
## **Метод решения**

Методы решения задачи анимации цилиндра и куба в Vulkan

1. Управление данными: Отдельные VkBuffer для вершин/индексов цилиндра и куба. Uniform буферы (UBO) для передачи матриц и времени на GPU.
2. Анимация преобразованиями: Цилиндр: Матрица модели с вращением вокруг своей оси: Куб: Матрица модели с орбитальным движением: позиция вычисляется как, где центр орбиты — позиция цилиндра.
3. Шейдеры: Вершинный шейдер умножает координаты на матрицы преобразований. Фрагментный шейдер рассчитывает цвет.
4. Генерация мешей: Цилиндр и куб сгенерированы алгоритмически на CPU с нормалями для освещения.

Итог: Комбинация матричных преобразований на CPU, передаваемых через UBO, и статической обработки в шейдерах на GPU позволила реализовать независимую анимацию объектов.

## Результаты



## **Выводы**

В результате работы была успешно реализована динамическая 3D-сцена на Vulkan, в которой цилиндр совершает вращение вокруг своей оси, а куб — орбитальное движение вокруг цилиндра. Задача решена за счёт комбинирования CPU-расчётов матричных преобразований (вращения и орбитального перемещения) и эффективной передачи этих данных на GPU через uniform-буфера. Использование единого конвейера рендеринга с отдельными буферами вершин для каждого объекта позволили создать визуально целостную и производительную анимацию, демонстрирующую базовые принципы работы с трансформациями, синхронизацией данных и управлением ресурсами в современном графическом API.