

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу «Компьютерная графика»

Студент: М.Ю. Курносков  
Преподаватель:  
Группа: М8О-306Б-23  
Дата:  
Оценка:  
Подпись:

## Условие

В этой лабораторной работе вам предстоит загрузить изображение из файла, создать текстуру из изображения и, используя текстурные координаты, а также сэмплеры, наложить текстуру на объекты

## Метод решения

Загрузка текстур: Реализована загрузка изображений из файлов (форматы PNG/JPG) с использованием библиотеки `stb_image`. Обработка различных цветовых каналов (RGB/RGBA) и конвертация в линейное пространство.

Создание ресурсов Vulkan для текстуры:

`VkImage` с оптимальным форматом

(например, `VK_FORMAT_R8G8B8A8_SRGB`) и необходимыми usage flags (`VK_IMAGE_USAGE_SAMPLED_BIT`, `VK_IMAGE_USAGE_TRANSFER_DST_BIT`).

`VkDeviceMemory` для хранения текстуры на GPU.

`VkImageView` для доступа к изображению из шейдеров.

`VkSampler` с настройками фильтрации (линейная/ближайшая), адресации (`repeat`, `clamp-to-edge`) и мипмаппинга.

Копирование данных: Загрузка пиксельных данных в промежуточный staging buffer (`VkBuffer`), затем копирование через команды в командном буфере с барьерами памяти и layout переходами

(`VK_IMAGE_LAYOUT_UNDEFINED` → `VK_IMAGE_LAYOUT_TRANSFER_DST_OPTIMAL` → `VK_IMAGE_LAYOUT_SHADER_READ_ONLY_OPTIMAL`).

Текстурные координаты: Вершины моделей (куб, цилиндр) дополнены атрибутом 2D текстурных координат (UV). Для куба — стандартные координаты каждой грани, для цилиндра — проекция боковой поверхности.

Интеграция в шейдеры:

Вершинный шейдер передает UV-координаты во фрагментный шейдер.

Во фрагментном шейдере реализована выборка цвета текстуры с помощью `texture(sampler2D, uv)`.

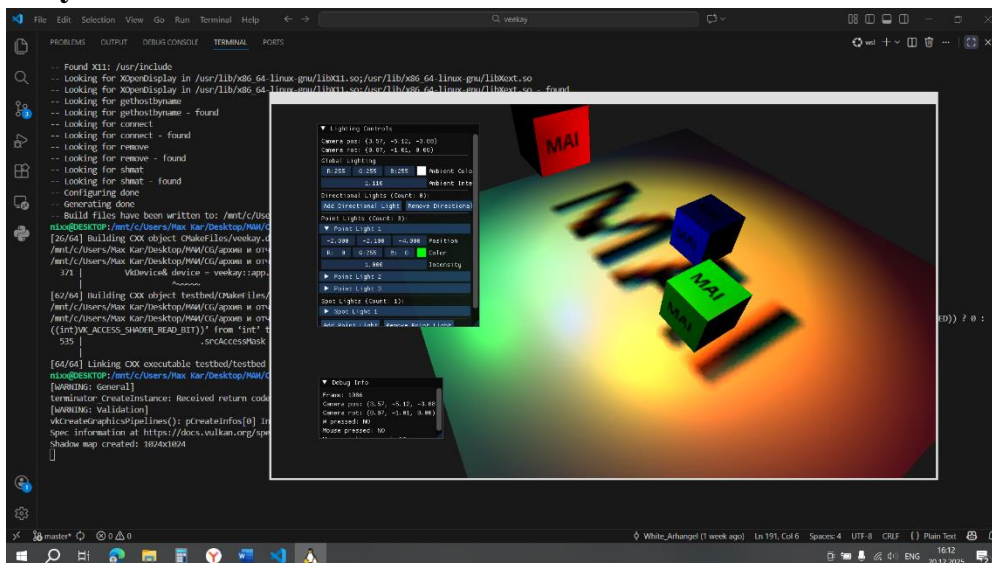
Текстурный цвет комбинируется с освещением Блинна-Фонга (модификация альбедо материала).

Дескрипторы и привязка: Создан дескрипторный набор (VkDescriptorSet) с комбинированным image sampler

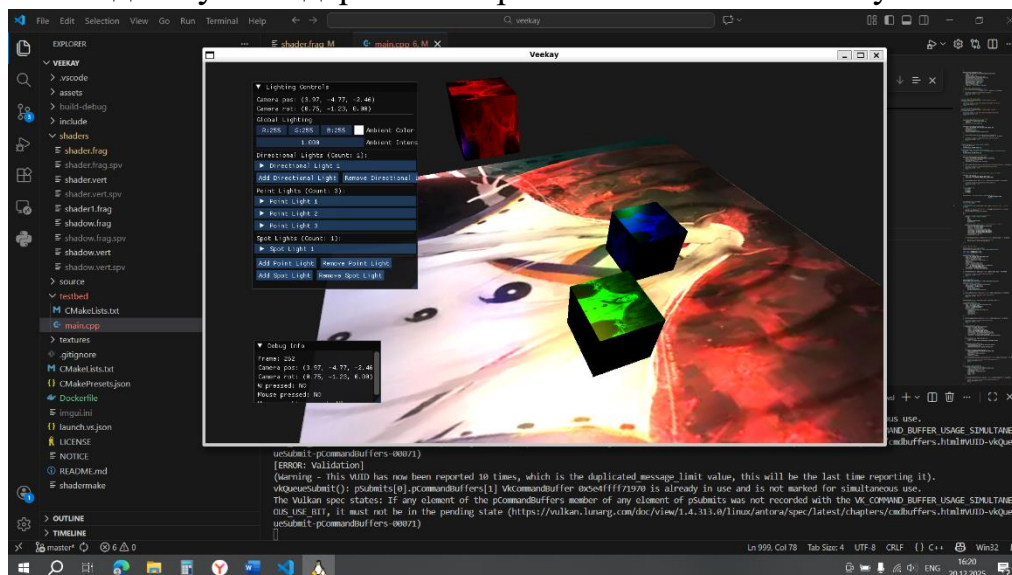
(VK\_DESCRIPTOR\_TYPE\_COMBINED\_IMAGE\_SAMPLER), привязанный к текстуре и семплеру. Дескриптор связывается с шейдером через layout.

Управление ресурсами: Освобождение памяти staging buffer после копирования, корректное уничтожение всех ресурсов Vulkan в конце работы программы.

## Результаты



Как видно кубы содержат изображение нашего института.



Как видно теперь кубы содержат текстуру изображения Ооски Кагуи.

## **Выводы**

В рамках лабораторной работы успешно реализована система работы с текстурами в Vulkan: от загрузки изображения с диска до наложения на 3D-объекты. Были освоены ключевые этапы: создание image ресурсов, копирование данных через staging buffer, настройка sampler'a для контроля качества текстурирования, передача текстурных координат и привязка дескрипторов в шейдерах. Результатом стала визуализация объектов с реалистичными текстурами, интегрированными в систему освещения, что демонстрирует полный цикл работы с текстурными ресурсами в современном графическом конвейере.