

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

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

**Лабораторная работа №3
по курсу “Компьютерная графика”**

Текстурирование

Выполнил: К. А. Арусланов

Группа: М8О-303Б-23

Преподаватель: В. Д. Бахарев

Москва, 2025

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.

Базовое условие на 3:

- Загрузить изображение из файла (желательно, чтобы ширина и высота изображения имели значение степени двойки, сами знаете зачем)
- Создать текстуру из изображения (использовать `veekay::graphics::Texture`)
- Создать объект сэмплера и задать разумные параметры сэмплирования
- Записать в набор дескрипторов новую привязку (дескриптор – изображение+сэмплер), чтобы его увидел фрагментный шейдер
- Вершины объектов должны содержать текстурные координаты
- Фрагментный шейдер должен сэмплировать текстуру, используя текстурные координаты, которые были переданы из вершинного шейдера

Базовое условие на 4:

Условие то же, что и у базового условия на оценку 3, но расширяется следующими требованиями:

- Реализовать использование разных текстур (`VkImageView`) и сэмплеров (`VkSampler`) моделями на сцене (с помощью использования разных наборов дескрипторов для каждой текстуры или набора текстур, то есть “материала”)

Дополнительные задания:

1. Сделайте нетривиальное сэмплирование текстуры в шейдере (можно модулировать входящие текстурные координаты какой-либо сложной функцией в шейдере; также можно сэмплировать много раз и смешивать цвета множества текселей по нетривиальной схеме)
2. Реализуйте две (или больше) дополнительные текстуры, описывающие материал, например `specular` текстуру (для определенных участков, где будут видны блики) и `emissive` текстуру (для участков, которые будут игнорировать просчет освещения и затенения, они будут светиться в темноте)

Метод решения

Была реализована система текстурирования объектов в Vulkan. На первом этапе изображение загружалось из файла на CPU (с помощью сторонней библиотеки загрузки изображений), после чего на его основе создавался объект `veekay::graphics::Texture`. Для корректной работы использовались изображения с размерами, кратными степени двойки, что упрощало использование `mip`-уровней и повышало совместимость с GPU. Далее для текстуры создавался объект сэмплера (`VkSampler`) с заданными параметрами фильтрации и адресации, такими как линейная фильтрация и режим повторения (REPEAT).

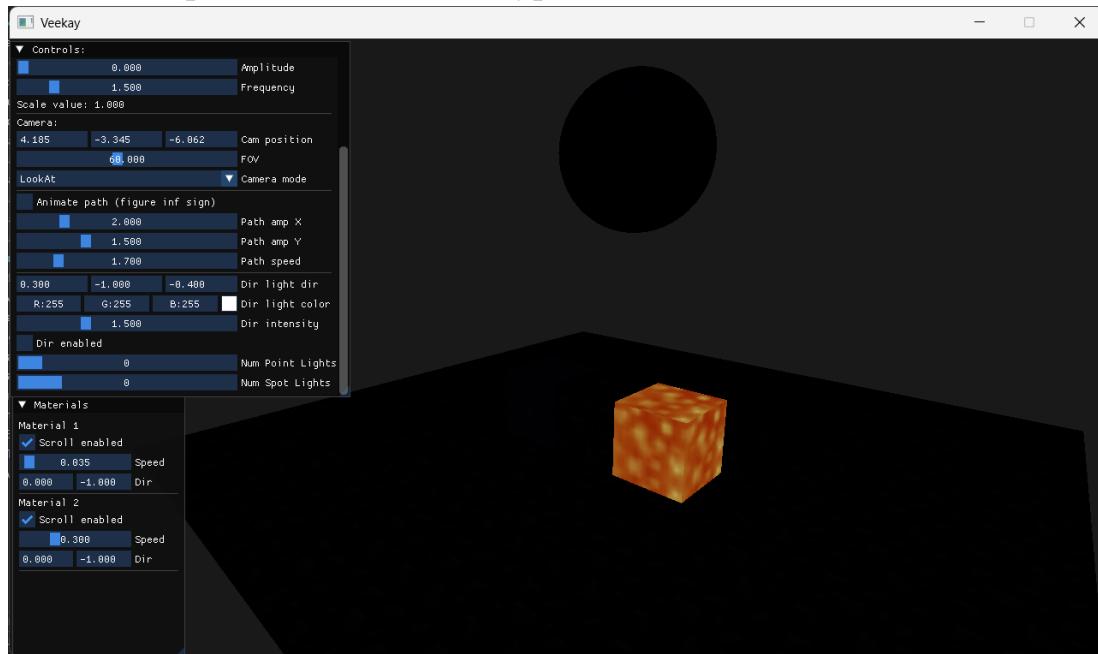
Затем текстура и сэмплер были записаны в набор дескрипторов в виде комбинированного дескриптора `image + sampler`, доступного фрагментному шейдеру. Геометрия всех объектов сцены была расширена текстурными координатами, которые передавались из вершинного шейдера во фрагментный. Во фрагментном шейдере выполнялось сэмплирование текстуры с использованием полученных координат, что позволило корректно наложить изображение на поверхность объектов.

Для выполнения требований на оценку «4» была введена концепция материала: каждая модель использовала собственный набор дескрипторов с индивидуальными `VkImageView` и `VkSampler` (для обсидиана включен фильтр NEAREST, для остальных использовался фильтр LINEAR). Таким образом на сцене одновременно отображались разные материалы — обсидиан, пол с повторяющейся текстурой, лава и вода. Переключение материалов происходило на уровне привязки дескрипторных наборов без изменения шейдерного кода, что соответствует правильной архитектуре Vulkan.

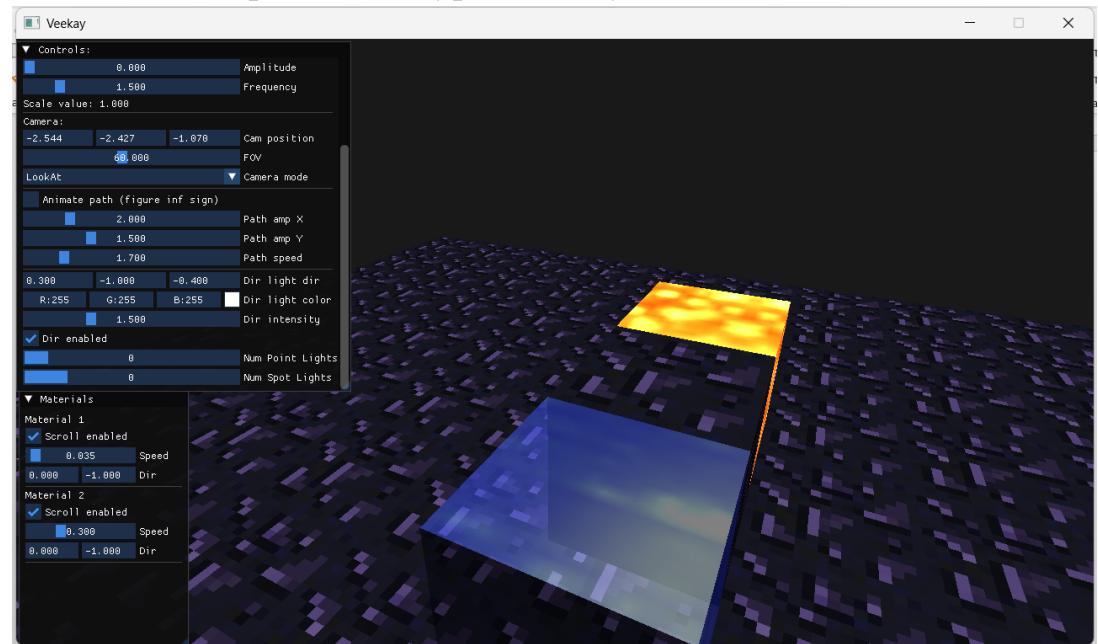
В рамках дополнительных заданий было реализовано нетривиальное сэмплирование текстур: для воды применялось искажение текстурных координат во времени, создающее эффект текущей поверхности, а также использовалась прозрачность. Также для воды дополнительно применялась specular-текстура, а для лавы emissive-текстура, позволяющая участкам материала светиться независимо от освещения сцены. Для пола использовалось масштабирование текстурных координат, что позволило добиться эффекта повторяющейся (тайловой) текстуры без увеличения геометрической сложности сцены.

Результаты

Демонстрация emissive-текстуры, свечение лавы в темноте



Демонстрация specular-текстуры воды, матовость текстуры обсидиана и повторения текстуры на полу



Выводы

В ходе выполнения лабораторной работы были приобретены практические навыки создания текстур в Vulkan, а также настройки

сэмплеров и работы с дескрипторными наборами. Я научился передавать текстурные координаты через графический конвейер и использовать их для сэмплирования текстур во фрагментном шейдере. Кроме того, был освоен подход к реализации материалов с использованием нескольких текстур (albedo, specular, emissive), а также методы нетривиального сэмплирования, включая анимацию, прозрачность и повторение текстур, что позволило создать более выразительные и визуально сложные материалы на сцене.