

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

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

**Лабораторная работа №2
по курсу “Компьютерная графика”**

Основы 3D графики

Выполнил: В. Н. Шишин
Группа: М8О-310Б-23
Преподаватель: В. Д. Бахарев

Москва, 2025

Условие

В этой лабораторной работе вам предстоит реализовать “камеру” с возможностью ее перемещения и освещения по модели Блинн-Фонга с использованием множества типов источников света.

Вариант задания:

Матрица камеры рассчитывается с помощью *матрицы трансформации камеры* (положения и ориентации/поворота). Должны быть реализованы следующие компоненты освещения: рассеянное, направленное и *точечные источники света*. Точечные источники света должны терять свою интенсивность по закону обратных квадратов.

Добавьте еще один тип источников света. Если по варианту были точечные, то реализуйте еще и прожекторные.

Добавьте ещё один режим отображения камеры. Если по варианту был расчет матрицы вида при помощи матрицы трансформации модели камеры, то реализуйте еще и режим Look-At. Сделайте для этого UI элемент с возможностью сохранения состояния камеры до переключения режима.

Метод решения

Реализуется камера с двумя режимами (матрица трансформации и Look-At), освещение по модели Блинн-Фонга с компонентами ambient, diffuse и specular, реализованы направленный источник, набор точечных источников с затуханием по закону обратных квадратов и дополнительный тип — прожектор; предусмотрён UI для управления камерой и сохранения состояния при переключении режимов.

Математика:

- Камера:
 - Transform-режим: видовая матрица вычисляется как обратное аффинное преобразование камеры — смещение и повороты: $\text{view} = T(-\text{position}) \cdot R_z(-\text{rot.z}) \cdot R_y(-\text{rot.y}) \cdot R_x(-\text{rot.x})$.
 - Look-At-режим: по углам (pitch/yaw) вычисляется forward; далее $\text{right} = \text{normalize}(\text{cross}(\text{forward}, \text{up}))$, $\text{up} = \text{cross}(\text{right}, \text{forward})$.

Видовая матрица составляется из базиса (right, up, -forward) и сдвигов (dot).

- Проекция: стандартная перспективная матрица $P(\text{fov}, \text{aspect}, \text{near}, \text{far})$. В шейдер передаётся комбинированная матрица $\text{view_projection} = \text{view} \cdot \text{projection}$.
- Сохранение состояния: при переключении режимов UI сохраняет/восстанавливает позицию и поворот камеры.
- Освещение (Blinn-Phong):
 - Ambient: постоянная фоновая составляющая.
 - Diffuse: $I_{\text{diffuse}} = k_d \cdot \max(\mathbf{N} \cdot \mathbf{L}, 0) \cdot I_{\text{light}}$.
 - Specular (Blinn): $\mathbf{H} = \text{normalize}(\mathbf{L} + \mathbf{V})$; $I_{\text{specular}} = k_s \cdot \max(\mathbf{N} \cdot \mathbf{H}, 0)^{\text{shininess}} \cdot I_{\text{light}}$.
 - Направленный свет: фиксированное направление \mathbf{L} , $\text{attenuation} = 1$.
 - Точечные источники: $\mathbf{L} = \text{normalize}(\text{lightPos} - \mathbf{P})$; интенсивность убывает по закону $\approx 1 / \text{distance}^2$ (с ограничением радиусом для оптимизации).
 - Прожектор: как точечный свет с дополнительным множителем на основе угла между направлением прожектора и вектором к фрагменту (плавное затухание на краю конуса).

Архитектура передачи данных (Vulkan-объекты):

- Буферы:
 - Uniform Buffer (UBO) для данных сцены (view_projection , позиция камеры, ambient, параметры направленного света).
 - Dynamic Uniform / Model UBO для матриц моделей и параметров материалов.
 - Storage Buffer (SSBO) для массивов источников света ($\text{PointLight}[]$, $\text{SpotLight}[]$).
- Дескрипторы:
 - VkDescriptorPool / $\text{VkDescriptorSetLayout}$ / VkDescriptorSet : биндинги для scene UBO, model UBO (dynamic) и SSBOs (lights).
- Пайплайн и шейдеры:
 - Graphics pipeline с вершинным и фрагментным шейдерами, реализующими применение матриц и расчёт Blinn-Phong.
- Команды:
 - Обновление UBO/SSBO на CPU (host visible memory), $\text{vkUpdateDescriptorSets}$, затем vkCmdBindPipeline ,

vkCmdBindDescriptorSets, vkCmdBindVertexBuffers, vkCmdBindIndexBuffer и vkCmdDrawIndexed.

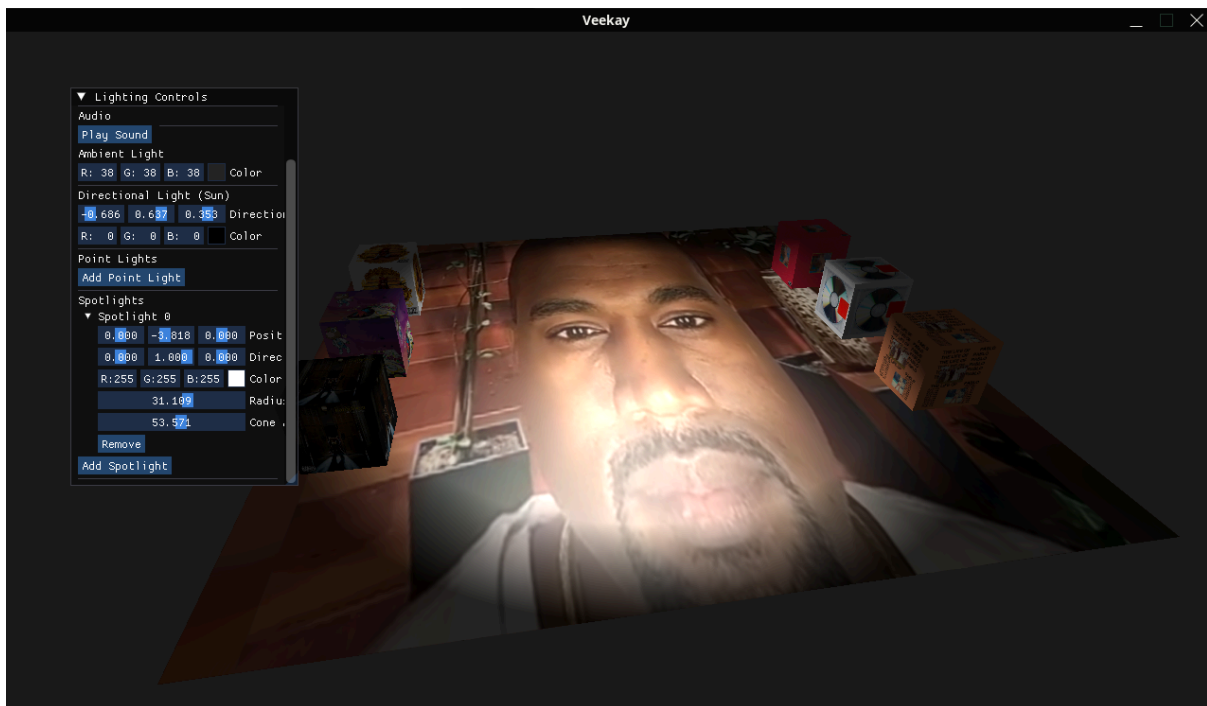
- UI:
 - Элементы ImGui для управления положением/поворотом камеры, переключения режима (Transform ↔ Look-At) и кнопки сохранения/восстановления состояния камеры; контролы для добавления/настройки точечных/прожекторных источников.

Результаты

Режим Transform: камера по матрице трансформации; направленный свет и несколько точечных источников с затуханием $\approx 1/r^2$.



Режим Look-At: камера, наведённая на цель; активный прожектор с видимым конусом и выраженной specular-компонентой.



Режим Look-At: освещение рассеянное (ambient light).



Выводы

Я научился(-ась) строить и переключать режимы камеры (Transform и Look-At) с сохранением состояния, формировать и комбинировать модельную, видовую и проекционную матрицы; реализовывать освещение по модели Блинн-Фонга (ambient, diffuse, specular) с направленными, точечными источниками (затухание $\approx 1/r^2$) и прожекторами; передавать

данные на GPU через UBO/SSBO и дескрипторы и управлять параметрами через UI для быстрой проверки визуальных эффектов.