

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: К.А. Ефременко
Группа: М8О-312Б-23
Преподаватель: В. Д. Бахарев

Москва, 2025

Условие

Цель работы

Реализовать систему камеры с возможностью её перемещения и освещение по модели Блинна-Фонга с использованием множества типов источников света. Освоить работу с Vulkan API для создания реалистичного освещения сцены в реальном времени.

Выполненный вариант

В данной работе реализован комбинированный подход, включающий элементы из нескольких вариантов задания:

Основной функционал (варианты 2 и 4):

- Матрица камеры рассчитывается с использованием **матрицы Look-At** с возможностью переключения на режим Transform (положение и ориентация)
- Реализованы все типы источников света:
 - **Рассеянное освещение** (ambient light)
 - **Направленное освещение** (directional light)
 - **Точечные источники света** (point lights) с затуханием по закону обратных квадратов
 - **Прожекторные источники света** (spot lights) с затуханием и гладкими краями
- Управление камерой с помощью клавиатуры и мыши без использования UI-элементов

Дополнительные задания

Выполнены оба дополнительных задания:

1. **Реализованы все типы источников света:** Помимо точечных источников добавлены прожекторные источники с плавными краями и затуханием интенсивности
2. **Переключение режимов камеры:** Реализовано переключение между режимами Transform и Look-At с сохранением состояния камеры при переключении

Дополнительные улучшения

- Система shadow mapping с PCF (Percentage Closer Filtering) для мягких теней
- Процедурные текстуры (specular и emissive maps)
- Продвинутая текстурная выборка с искажением UV-координат и мультисэмплингом
- Полноценная сцена с полом и несколькими объектами с разными материалами

Метод решения

Математические основы

1. Система камеры

Режим Look-At:

Матрица вида для режима Look-At строится на основе трёх векторов: позиция наблюдателя (eye), точка, на которую смотрит камера (target), и вектор "вверх" (up). Алгоритм построения:

```
// Вычисление базисных векторов камеры zaxis = normalize(eye - target) // Направление "назад"
xaxis = normalize(cross(up, zaxis)) // Направление "вправо" yaxis = cross(zaxis, xaxis) // Направление "вверх"
// Матрица вида View = [xaxis.x yaxis.x zaxis.x 0] [1 0 0 eye.x]
[xaxis.y yaxis.y zaxis.y 0] × [0 1 0 -eye.y] [xaxis.z yaxis.z zaxis.z 0] [0 0 1 -eye.z] [0 0 0 1] [0 0 0 1]
```

Режим Transform:

В этом режиме матрица вида строится путём инверсии матрицы модели камеры:

View = inverse(Rotation_Y × Rotation_X × Rotation_Z × Translation) **Матрица**

перспективной проекции:

P = perspective(fov, aspect, near, far) где
fov = 60°, near = 0.01, far = 100.0

2. Модель освещения Блинна-Фонга

Модель Блинна-Фонга представляет собой улучшенную версию модели Фонга, которая использует полувектор (halfway vector) для более эффективного расчёта бликов:

```
// Общая формула освещения L_total = L_ambient + Σ(L_diffuse + L_specular) //
Компоненты: L_ambient = ambient_color × ambient_intensity × albedo // Диффузная
составляющая (рассеянное отражение) L_diffuse = light_color × light_intensity × max(N · L,
0) × albedo // Блик (зеркальное отражение) - модель Блинна H = normalize(L + V) //
Полувектор между направлением на свет и на камеру L_specular = light_color ×
light_intensity × pow(max(N · H, 0), shininess) × specular_color где: N - нормаль
поверхности L - направление на источник света V - направление на камеру H -
полувектор (halfway vector)
```

Преимущество модели Блинна перед классической моделью Фонга: полувектор H меняется медленнее при движении камеры, что даёт более стабильные результаты и лучше подходит для аппаратной реализации.

3. Типы источников света

Рассеянный свет (Ambient Light):

L_ambient = ambient.color × ambient.intensity × material.albedo

Обеспечивает базовое освещение всей сцены, имитируя многократно отражённый свет.

Направленный свет (Directional Light):

```
L = normalize(-light.direction) L_dir = calculateBlinnPhong(L, N, V, light.color, light.intensity)
```

Имитирует удалённый источник света (например, солнце), лучи которого параллельны.

Точечный свет (Point Light):

```
L = normalize(light.position - fragment_position) distance = length(light.position - fragment_position) // Затухание по закону обратных квадратов attenuation = 1.0 / (constant + linear * distance + quadratic * distance^2) L_point = calculateBlinnPhong(L, N, V, light.color, light.intensity) * attenuation
```

Источник света, излучающий свет во все стороны с затуханием по расстоянию.

Прожекторный свет (Spot Light):

```
L = normalize(light.position - fragment_position) distance = length(light.position - fragment_position) attenuation = 1.0 / (constant + linear * distance + quadratic * distance^2) // Угловое затухание с плавными краями theta = dot(L, normalize(-light.direction)) epsilon = inner_cutoff - outer_cutoff intensity = clamp((theta - outer_cutoff) / epsilon, 0.0, 1.0) L_spot = calculateBlinnPhong(...) * attenuation * intensity
```

Направленный конус света с мягкими краями, имитирующий реальный прожектор.

4. Shadow Mapping

Для создания реалистичных теней используется техника shadow mapping:

1. **Первый проход (shadow pass):** Рендеринг сцены с точки зрения источника света в depth buffer (карту глубины)
2. **Второй проход (lighting pass):** При расчёте освещения для каждого фрагмента проверяется, находится ли он в тени, сравнивая его глубину с сохранённой в shadow map
3. **PCF (Percentage Closer Filtering):** Для сглаживания краёв теней выполняется выборка из 9 соседних текстелей shadow map и усреднение результата

Архитектура Vulkan

Структуры данных

Вершина (Vertex):

```
struct Vertex { vec3 position; // Позиция в пространстве модели vec3 normal; // Нормаль для расчёта освещения vec2 uv; // Текстурные координаты }; Uniform буферы:
```

```
// Данные сцены (обновляются каждый кадр) struct SceneUniforms { mat4 view_projection; // Матрица вид-проекция vec3 camera_position; // Позиция камеры в мировых координатах mat4 light_view_projection; // Матрица для shadow mapping }; // Данные модели (для каждого объекта) struct ModelUniforms { mat4 model; // Матрица
```

трансформации модели vec3 albedo_color; // Базовый цвет (диффузный) vec3 specular_color; // Цвет бликов float shininess; // Параметр блеска (8-256) }; **Storage буфер для источников света:**

```
struct LightBuffer { AmbientLight ambient_light; uint directional_count; // Количество направленных источников uint point_count; // Количество точечных источников uint spot_count; // Количество прожекторов DirectionalLight directional_lights[8]; PointLight point_lights[32]; // До 32 точечных источников SpotLight spot_lights[32]; // До 32 прожекторов };
```

Использование shader storage buffer (SSBO) вместо uniform buffer позволяет хранить динамические массивы источников света и эффективно управлять ими в реальном времени.

Pipeline конфигурация

Этап	Описание
Vertex Shader	Трансформация вершин, вычисление мировых координат и нормалей
Rasterization	Интерполяция атрибутов, back-face culling
Fragment Shader	Расчёт освещения по модели Блинна-Фонга, shadow mapping, текстурирование
Depth Testing	Z-буферизация для корректного отображения перекрывающихся объектов

Descriptor Sets

Организация ресурсов для шейдеров:

- **Binding 0:** Scene Uniforms (uniform buffer)
- **Binding 1:** Model Uniforms (dynamic uniform buffer)
- **Binding 2:** Light Buffer (storage buffer)
- **Binding 3:** Albedo Texture (sampler2D)
- **Binding 4:** Specular Texture (sampler2D)
- **Binding 5:** Emissive Texture (sampler2D)
- **Binding 6:** Shadow Map (sampler2D)

Управление камерой

Управление мышью:

- ЛКМ + перемещение мыши: поворот камеры (изменение yaw и pitch)
- Горизонтальное движение мыши → вращение вокруг оси Y (yaw)
- Вертикальное движение мыши → вращение вокруг оси X (pitch)

Управление клавиатурой:

- **W/S:** движение вперёд/назад вдоль направления взгляда
- **A/D:** движение влево/вправо (стрейф)

- **Space/Shift:** движение вверх/вниз по мировой оси Y

В режиме Look-At камера вращается вокруг целевой точки по орбите, сохраняя постоянное расстояние.

Результаты

Демонстрация работы программы

Скриншот 1

Рис. 1. Базовая сцена с тремя кубами на плоскости. Видна панель управления "Lighting & Camera Controls" с возможностью переключения режимов камеры (Transform Mode / LookAt Mode). Камера находится в режиме Transform Mode. Демонстрируется базовое освещение сцены без детального отображения источников света.

Скриншот 2

Рис. 2. Та же сцена с раскрытой панелью освещения. Видны настройки Ambient Light (рассеянный свет) с синеватым оттенком (R:51, G:51, B:77) и интенсивностью 2.0. Также показаны свёрнутые разделы для Directional Light (направленный свет), Point Lights (точечные источники) и Spot Light (прожектор). Камера находится под углом, демонстрируя глубину сцены.

Скриншот 3

Рис. 3. Демонстрация работы направленного света (Directional Light). В UI видны параметры: Direction (-0.200, 1.000, -0.300), Color (белый), Intensity 2.0. Сцена ярко освещена, видны чёткие блики на кубах благодаря модели Блинна-Фонга. Пол имеет выраженную текстуру с причудливыми узорами. Демонстрируется сильное освещение сцены сверху.

Скриншот 4

Рис. 4. Работа точечных источников света (Point Lights). Открыта панель "Point Light 0" с красноватым оттенком (R:255, G:153, B:153), интенсивностью 3.0, и параметрами затухания (Linear и Quadratic для реализации закона обратных квадратов). На сцене видно, как красный свет окрашивает пол вокруг своей позиции (-1.000, -1.258, 0.000), создавая круговой градиент затухания. Видны также зоны влияния других источников света.

Скриншот 5

Рис. 5. Общий вид сцены с разных ракурсов, демонстрирующий работу всех типов освещения одновременно. Видны блики на кубах с разными материалами (красный, зелёный, синий), интерполяция освещения на гранях, мягкие тени. Пол освещён комбинацией точечных источников, создающих сложную картину освещения с перекрывающимися зонами влияния разных цветов. Видны также эмиссивные участки на полу (светящиеся линии сетки).

Функциональные возможности

Реализованная программа предоставляет следующие возможности:

Управление камерой

- **Два режима камеры:** Transform Mode (позиция + вращение) и LookAt Mode (глаз + цель + вверх)
- **Свободное перемещение:** WASD для движения, мышь для поворота
- **Плавное управление:** камера реагирует на ввод в реальном времени без задержек
- **Сохранение состояния:** при переключении режимов состояние камеры сохраняется

Система освещения

- **Ambient Light:** настройка цвета и интенсивности глобального освещения
- **Directional Light:** управление направлением, цветом и интенсивностью
- **Point Lights (2 шт.):** позиция, цвет, интенсивность, параметры затухания (linear, quadratic)
- **Spot Light:** позиция, направление, цвет, интенсивность, углы конуса (inner/outer angle) для гладких краёв

Материалы объектов

- **Плоскость (пол):** текстурированная поверхность с процедурными паттернами
- **Красный куб:** высокий shininess (64), яркие блики
- **Зелёный куб:** средний shininess (32)
- **Синий куб:** очень высокий shininess (128), металлический вид

Технические характеристики

Параметр	Значение
Графический API	Vulkan 1.0+
Язык шейдеров	GLSL 4.50 (компилируется в SPIR-V)
Модель освещения	Blinn-Phong с нормальными на вершинах
Максимум направленных источников	8
Максимум точечных источников	32
Максимум прожекторов	32
Разрешение shadow map	2048 × 2048
Shadow filtering	PCF 3×3 (9 samples)
Количество вершин куба	24 (с правильными нормальными для каждой грани)
Количество индексов куба	36 (12 треугольников)

Выводы

В ходе выполнения лабораторной работы были успешно освоены продвинутое техники компьютерной графики и получены глубокие знания о работе современных графических API. Реализованы все требования базового задания и выполнены оба дополнительных задания.

Приобретённые знания и навыки:

- **Системы камер:** Освоил два принципиально разных подхода к построению матрицы вида — через матрицу трансформации и через функцию Look-At. Понял математику построения ортонормированного базиса камеры
 - **Модель освещения Блинна-Фонга:** Изучил и реализовал классическую модель освещения с разделением на ambient, diffuse и specular компоненты. Понял преимущество использования полувектора перед вектором отражения в модели Фонга
 - **Типы источников света:** Реализовал четыре типа источников: – Ambient — глобальное рассеянное освещение
 - Directional — параллельные лучи (имитация солнца)
 - Point — всенаправленное излучение с затуханием по расстоянию
 - Spot — конический луч с гладкими краями
 - **Физически корректное затухание:** Применил закон обратных квадратов для точечных и прожекторных источников с настраиваемыми параметрами constant, linear и quadratic
 - **Shader Storage Buffers (SSBO):** Научился использовать SSBO для передачи динамических массивов данных в шейдеры, что позволяет эффективно управлять множеством источников света
 - **Shadow Mapping:** Реализовал технику создания теней:
 - Двухпроходный рендеринг (shadow pass + lighting pass)
 - PCF (Percentage Closer Filtering) для сглаживания краёв теней
 - Настройка bias для предотвращения артефактов (shadow acne)
 - **Продвинутая работа с текстурами:** Создал процедурные текстуры (specular и emissive), реализовал искажение UV-координат и мультисэмплинг для улучшения детализации
 - **Descriptor Sets в Vulkan:** Освоил организацию ресурсов через descriptor sets с различными типами биндингов (uniform buffer, dynamic uniform buffer, storage buffer, combined image sampler)
 - **Интерактивное управление:** Реализовал intuitive управление камерой через мышь и клавиатуру, а также полный контроль над параметрами освещения через ImGui
 - **Оптимизация рендеринга:** Использовал back-face culling, depth testing, dynamic uniform buffers для эффективного рендеринга множества объектов
- Технические достижения:**

- Построена полноценная система освещения с поддержкой до 72 источников света одновременно (8 directional + 32 point + 32 spot)
- Реализована плавная интерполяция освещения между вершинами благодаря передаче нормалей
- Создана визуально привлекательная сцена с реалистичным освещением и материалами
- Достигнута высокая производительность при рендеринге сложной сцены с множественными источниками света
- Корректная работа с нормальными: вычисление normal matrix для учёта nonuniform scaling
- Реализация гладких краёв прожекторов через linear interpolation между inner и outer cutoff углами

Понимание графического конвейера:

Работа дала глубокое понимание того, как данные проходят через графический конвейер — от вершинного буфера через трансформации в vertex shader, интерполяцию атрибутов в rasterizer, до финального расчёта цвета пикселя в fragment shader с учётом освещения, теней и текстур.

Особенно ценным оказался опыт отладки освещения: понимание того, как каждая компонента модели Блинна-Фонга влияет на итоговый результат, как правильно нормализовать векторы, как избежать артефактов при расчёте specular компоненты.

Практическое применение:

Полученные знания являются фундаментальными для разработки игровых движков, систем рендеринга и любых приложений, требующих реалистичного отображения 3D-графики. Модель Блинна-Фонга, несмотря на свою относительную простоту, до сих пор широко используется в реальном времени графике благодаря оптимальному балансу между качеством и производительностью.