МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

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

Лабораторная работа № 4 по курсу «Компьютерная графика»

Выполнил: Парфенов М.М.
Группа: М8О-301Б-22
Преподаватель: Филиппов Г.С.
Дата: 22.12.24
Оценка:

Москва, 2024

Лабораторная работа № 4

Тема: Освещение и работа с шейдерами
Задача: Постройте куб в 3D-пространстве. Реализуйте многоцветное освещение с использованием модели освещения Фонга (Phong Lighting), где объекты сцены имеют как диффузные, так и зеркальные компоненты освещения. Реализуйте возможность изменения цвета источника света в реальном времени. Дополнительно: Добавьте управление параметрами материала объекта (например, коэффициент отражения).
Вариант 10. Многоцветное освещение (Phong Lighting)

1 Решение

В этом проекте я реализовал рендеринг 3D-куба с использованием OpenGL ES 2.0 и модели освещения Фонга. Вот как я подошел к решению задачи:

1. **Инициализация и настройка контекста OpenGL:**
    - Я использовал GLFW для создания окна и контекста для работы с OpenGL ES 2.0. Для этого указал соответствующие параметры через `glfwWindowHint()`, чтобы создать контекст именно для OpenGL ES 2.0.
    - Включил тест глубины для правильного отображения объектов в 3D пространстве.
2. **Структура данных для куба:**
    - Для представления куба создал структуру `CubeVertex`, которая включает в себя позицию вершины, нормаль и цвет. Это позволило мне более гибко управлять атрибутами вершин.
    - Вершины куба были определены вручную для каждой из 6 граней, каждая грань состоит из 2 треугольников, всего 36 вершин.
3. **Шейдеры:**
    - Написал два шейдера: вершинный и фрагментный. Вершинный шейдер учитывает позицию вершин, их нормали и цвета, а также использует матрицы трансформации для проекции и освещения.
    - В фрагментном шейдере реализована модель освещения Фонга, которая учитывает амбиентное, диффузное и зеркальное освещение. Я добавил поддержку как для точечного света, так и для прожектора, с учетом углов отсечения для прожектора.
4. **Шейдерная программа:**
    - Я компилировал и линкуй шейдеры с помощью вспомогательных функций, таких как `compileShader()` и `createProgram()`. Это позволяло мне динамически загружать и применять шейдеры к объектам сцены.
5. **Настройка буферов и атрибутов:**
    - Для передачи данных о вершинах в OpenGL использовал буфер (VBO). Я создал VBO для хранения данных о позициях, нормалях и цветах каждой вершины, а затем настроил атрибуты шейдеров с помощью `glVertexAttribPointer()` и `glEnableVertexAttribArray()`.
6. **Освещение и материалы:**

- Я задал параметры освещения, такие как позиция источника света, его цвет и тип (точечный свет или прожектор). Для материала куба были установлены параметры амбиентного, диффузного и зеркального отражения, а также блеск материала (shininess).
- Эти значения передаются в шейдеры через uniform-переменные, что позволяет динамично изменять их в процессе рендеринга.

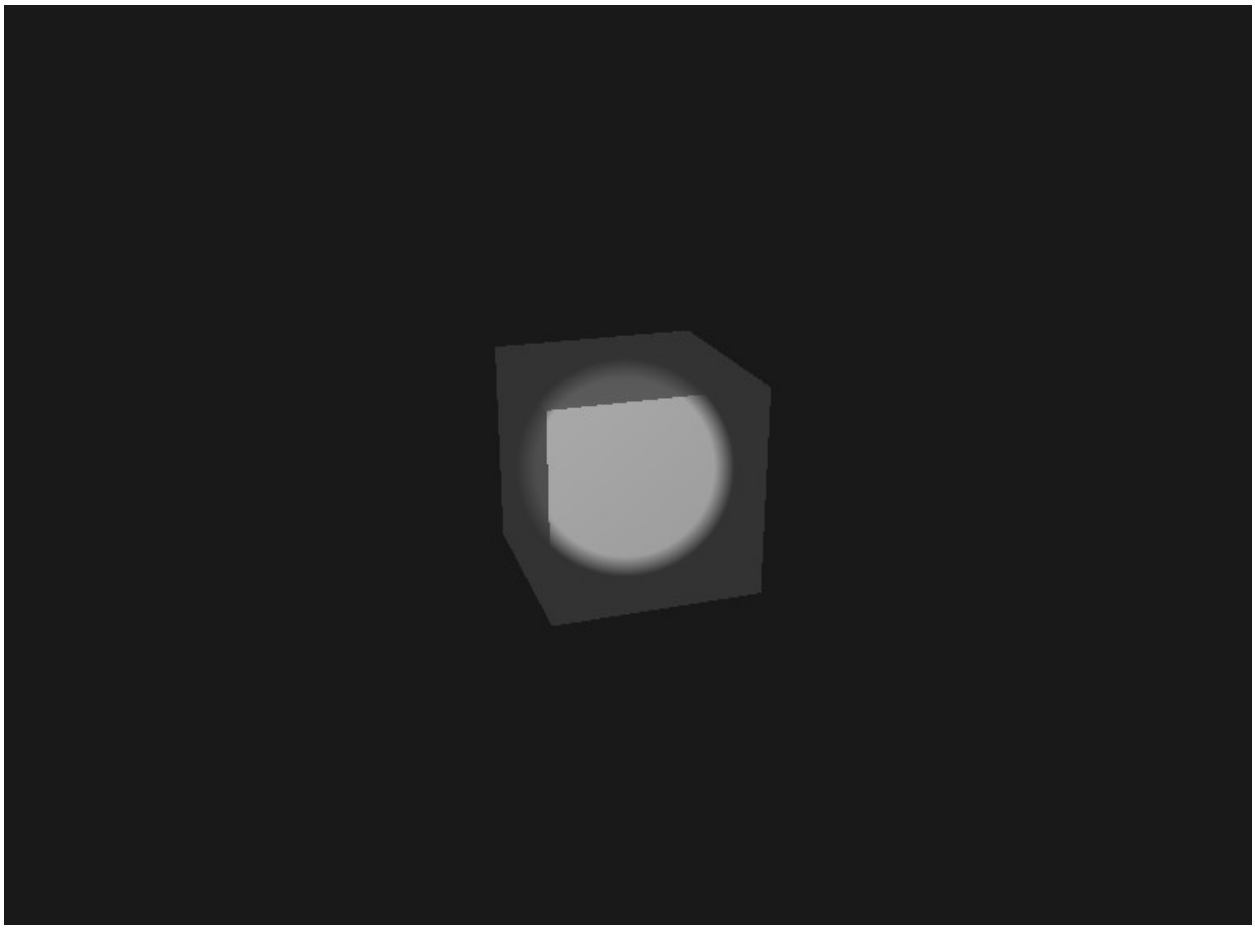
7. **Управление камерой:**

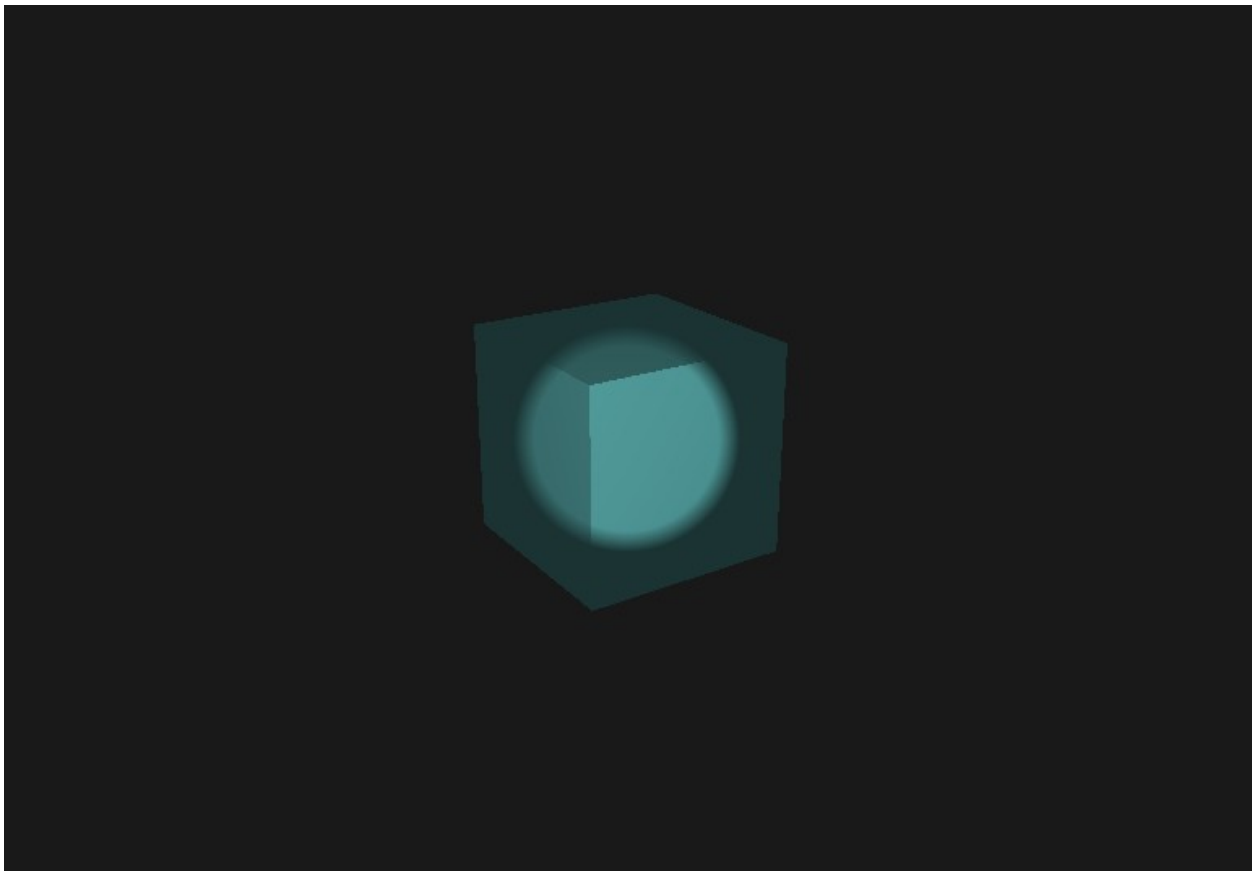
    - Камера вращается вокруг куба по окружности с заданной угловой скоростью и радиусом. Для этого я использую углы вращения и соответствующие матрицы преобразования для модели камеры.

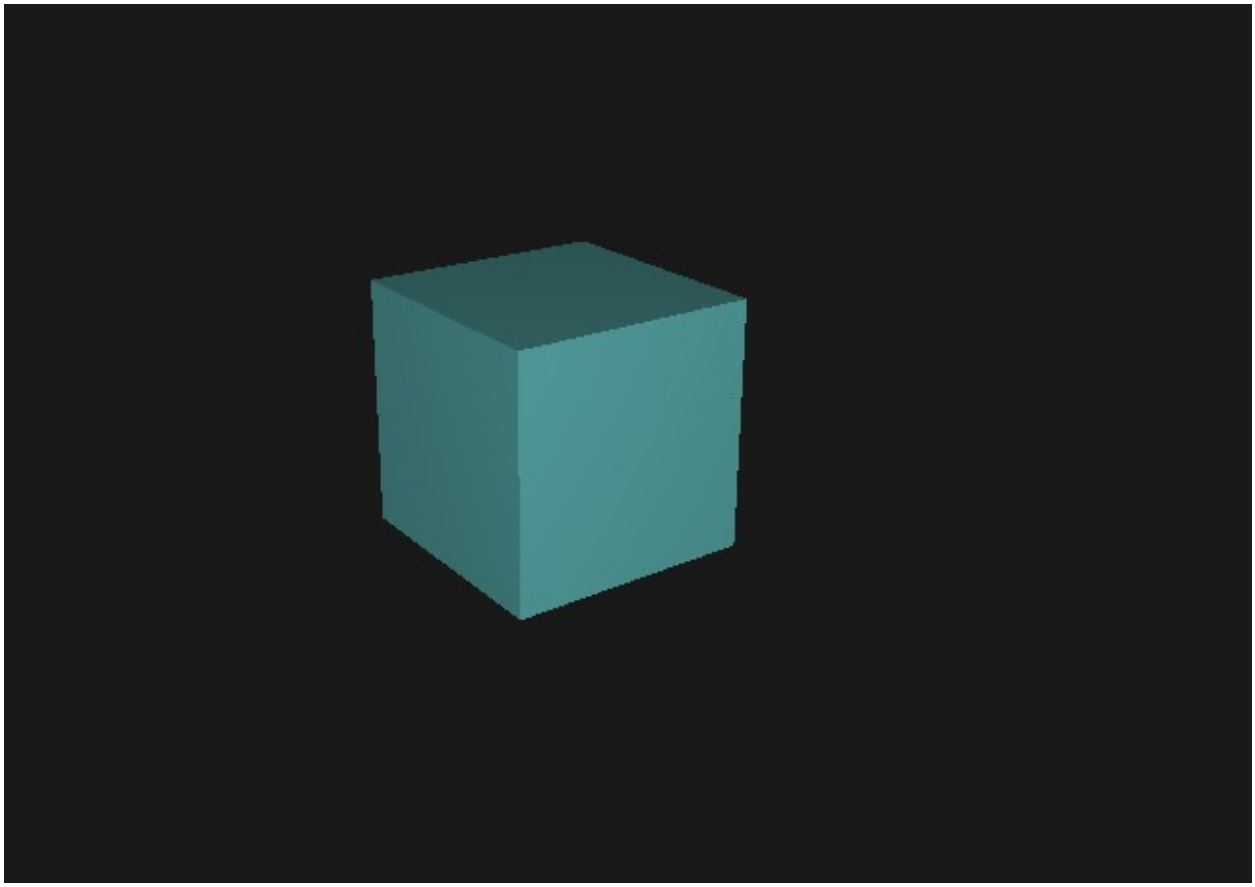
8. **Рендеринг:**

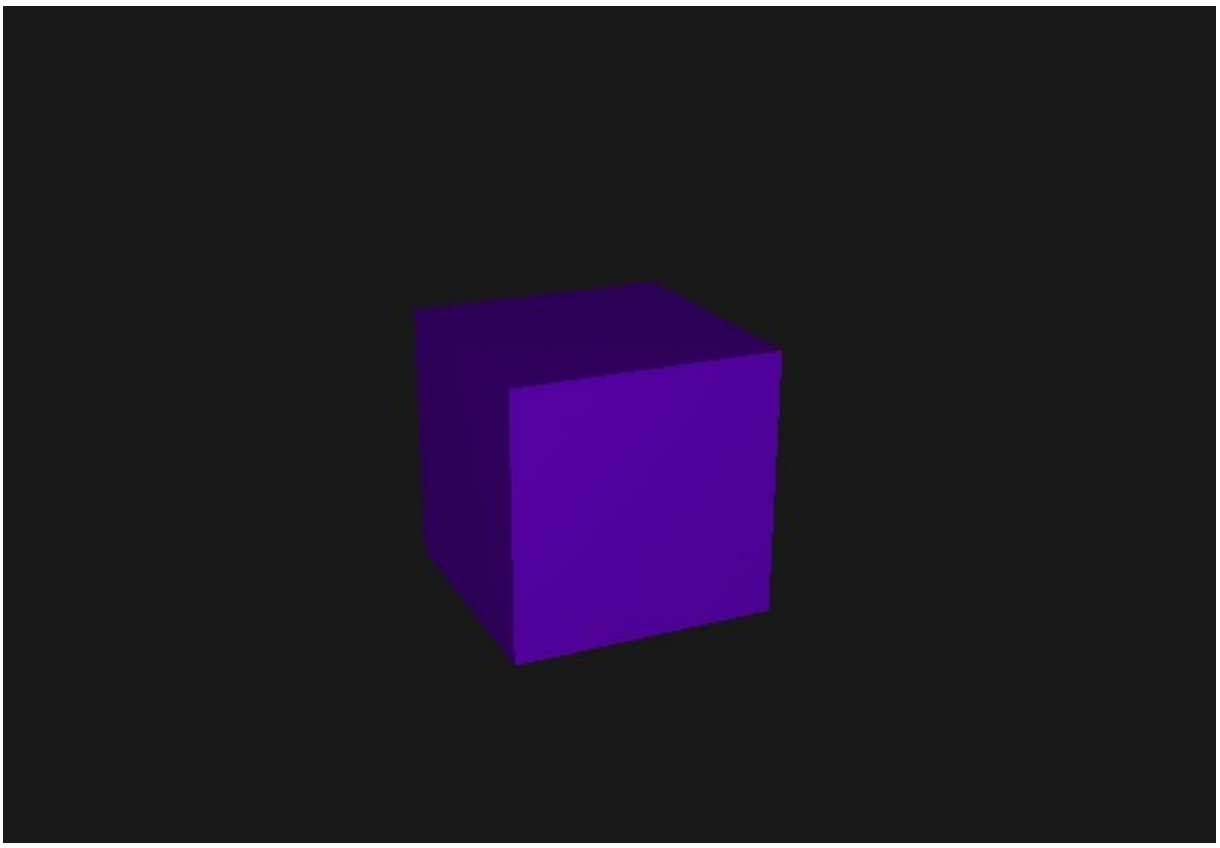
    - В главном цикле программы я обновляю время, рассчитываю углы вращения камеры, обрабатываю ввод с клавиатуры (например, для изменения типа света или остановки работы программы) и передаю необходимые значения в шейдеры.
    - После этого вызываю функцию для отрисовки куба с соответствующими трансформациями и освещением.

Результатом работы программы является 3D-куб, на который накладываются эффекты освещения Фонга, и который можно наблюдать с разных углов, вращая камеру вокруг объекта.









2 Вывод

В результате выполнения работы был реализован 3D-рендеринг куба с использованием освещения по модели Фонга с поддержкой двух типов освещения: точечного и прожектора. В процессе работы был использован OpenGL ES 2.0 для создания графического контекста и рендеринга, а также шейдеры для обработки освещения и материалов. Куб был оснащен нормалями и цветами для каждой вершины, что позволило эффективно отображать освещение и создать реалистичную картину. Камера вращалась вокруг куба, обеспечивая динамичное отображение сцены. Работа также продемонстрировала использование векторных математических операций через библиотеку GLM для обработки трансформаций и освещения.