

Лабораторные работы. Вариант 0.8 – Черновой

Лабораторная работа 1: Основы 2D-графики и трансформаций

Цель лабораторной работы:

В данной лабораторной работе вам предстоит научиться работать с графическим API для отрисовки 2D-примитивов, освоить основные 2D-трансформации (перемещение, масштабирование, поворот) и изучить алгоритмы построения 2D-кривых.

Требования:

Вы должны использовать C++ (OpenGL + SFML) или C# (OpenTK).

Программа должна работать в реальном времени, обновляя изображение в цикле.

Визуальный результат необходимо продемонстрировать на экране с возможностью управления через интерфейс.

Вариант 1. Отрисовка и трансформация линии

Инициализируйте окно с помощью графического API.

Нарисуйте линию, задав координаты её концов.

Реализуйте возможность перемещения, масштабирования и поворота линии с использованием клавиатуры.

При перемещении и вращении линия должна корректно обновляться на экране.

Дополнительно: Реализуйте отображение текущих координат и угла поворота линии в углу экрана.

Вариант 2. Отрисовка прямоугольника с трансформациями

Создайте программу, которая отрисовывает прямоугольник.

Реализуйте возможность изменения положения, угла поворота и масштабирования прямоугольника.

Управляйте трансформациями с помощью клавиатуры.

Дополнительно: Добавьте возможность изменять цвет прямоугольника в зависимости от направления трансформации.

Вариант 3. Отрисовка многоугольника

Создайте многоугольник с произвольным числом сторон (например, пятиугольник).

Реализуйте его перемещение по экрану с помощью клавиатуры.

Добавьте возможность масштабирования и вращения многоугольника вокруг его центра.

Дополнительно: Реализуйте замкнутую траекторию движения многоугольника по экрану.

Вариант 4. Построение кривой Безье

Реализуйте алгоритм построения квадратичной кривой Безье по трем контрольным точкам.

Контрольные точки должны быть видимы и управляемы пользователем через интерфейс.

Программа должна динамически обновлять кривую при перемещении контрольных точек.

Дополнительно: Реализуйте сглаживание кривой для более плавного отображения.

Вариант 5. Построение кубической кривой Безье

Постройте кубическую кривую Безье, используя четыре контрольные точки.

Контрольные точки должны быть отображены на экране и иметь возможность перемещения.

Кривая должна обновляться в реальном времени при изменении положения любой контрольной точки.

Дополнительно: Реализуйте анимацию, где кривая плавно изменяет свою форму в зависимости от времени.

Вариант 6. Реализация круговой трансформации

Нарисуйте круг с фиксированным радиусом.

Реализуйте возможность перемещения круга по экрану, масштабирования и вращения вокруг центра.

Управление должно осуществляться с помощью клавиатуры и мыши.

Дополнительно: Добавьте визуализацию пути, по которому перемещается центр круга.

Вариант 7. Отрисовка многоугольника с интерполяцией

Постройте многоугольник с 6-ю вершинами.

Реализуйте перемещение всех вершин одновременно с помощью матричных операций.

Добавьте интерполяцию между начальной и конечной позицией многоугольника для плавного движения.

Дополнительно: Реализуйте циклическую анимацию, при которой многоугольник меняет свою форму и возвращается в исходное состояние.

Вариант 8. Алгоритм Брезенхэма для окружности

Реализуйте алгоритм Брезенхэма для отрисовки окружности с заданным радиусом.

Окружность должна быть отрисована с центром в любой точке экрана.

Добавьте возможность динамического изменения радиуса окружности с помощью интерфейса.

Дополнительно: Реализуйте анимацию пульсации окружности (увеличение и уменьшение радиуса со временем).

Вариант 9. Отрисовка эллипса и его трансформации

Реализуйте алгоритм отрисовки эллипса по заданным полуосям.

Добавьте возможность изменения масштаба и поворота эллипса вокруг его центра.

Управление должно осуществляться с помощью клавиатуры.

Дополнительно: Добавьте динамическое изменение полуосей эллипса в зависимости от времени.

Вариант 10. Реализация алгоритма отсечения линии (Cohen-Sutherland)

Реализуйте алгоритм отсечения линии с использованием алгоритма Козна-Сазерленда.

Нарисуйте окно для отсечения и несколько линий, пересекающих это окно.

Реализуйте динамическое отсечение линий при изменении положения окна.

Дополнительно: Реализуйте возможность перемещения и масштабирования окна отсечения.

Вариант 11. Анимация по траектории кривой Безье

Реализуйте анимацию движения объекта (например, круга) по траектории кривой Безье.

Анимация должна быть плавной, объект должен перемещаться по кривой с равномерной скоростью.

Добавьте управление скоростью анимации через интерфейс.

Дополнительно: Реализуйте изменение размера объекта в зависимости от его положения на кривой.

Вариант 12. Построение ломаной кривой (Poly-Line)

Реализуйте отрисовку ломаной кривой через несколько контрольных точек, которые можно перемещать.

Обеспечьте динамическое обновление кривой при перемещении контрольных точек.

Добавьте возможность добавления и удаления точек через интерфейс.

Дополнительно: Реализуйте возможность изменения формы кривой со временем, где точки автоматически перемещаются

Лабораторная работа 2: Основы 3D-графики и проекция

Цель лабораторной работы:

В этой лабораторной работе вы познакомитесь с основами 3D-графики: построением простых 3D-объектов, проекцией на 2D-плоскость, а также научитесь работать с матрицами перспективы и ортогографической проекции.

Требования:

Вы должны использовать OpenGL + SFML или OpenTK + C#.

Программа должна работать в реальном времени, с возможностью динамической смены проекции и трансформаций объектов.

Все объекты должны корректно отрисовываться с учетом проекции и иметь возможность взаимодействия с пользователем.

Вариант 1. Построение куба с ортогографической проекцией

Реализуйте отрисовку куба в 3D-пространстве.

Примените ортогографическую проекцию для отображения куба на экране.

Добавьте возможность вращения куба вокруг осей X и Y с помощью клавиатуры.

Дополнительно: Реализуйте смену видов (сверху, сбоку, спереди) с использованием разных матриц проекций.

Вариант 2. Построение пирамиды с перспективной проекцией

Постройте 3D-пирамиду (с квадратным основанием).

Примените перспективную проекцию для отображения пирамиды.

Реализуйте вращение пирамиды вокруг всех осей с помощью клавиш управления.

Дополнительно: Добавьте динамическое изменение угла обзора (field of view) и наблюдайте, как это влияет на проекцию.

Вариант 3. Построение цилиндра с управлением камерой

Постройте цилиндр в 3D-пространстве.

Примените перспективную проекцию для его отображения.

Реализуйте перемещение камеры вокруг цилиндра по круговой траектории с возможностью изменения радиуса этой траектории.

Дополнительно: Добавьте управление высотой камеры для осмотра цилиндра сверху и снизу.

Вариант 4. Реализация ортографической и перспективной проекции для куба

Постройте куб в 3D-пространстве.

Реализуйте переключение между ортографической и перспективной проекциями в реальном времени.

Обеспечьте плавный переход между проекциями.

Дополнительно: Добавьте возможность изменения параметров каждой проекции (например, угол обзора для перспективной и размер проекционного объема для ортографической).

Вариант 5. Построение сферического объекта с возможностью масштабирования

Постройте сферу в 3D-пространстве.

Реализуйте возможность масштабирования сферы (увеличение и уменьшение радиуса).

Примените перспективную проекцию для корректного отображения.

Дополнительно: Добавьте возможность перемещения камеры вокруг сферы с изменением расстояния до неё.

Вариант 6. Построение куба с нормальными для каждой грани

Постройте 3D-куб.

Для каждой грани куба рассчитайте и визуализируйте нормали.

Используйте перспективную проекцию для отображения куба на экране.

Дополнительно: Реализуйте вращение куба вокруг осей, так чтобы нормали менялись в зависимости от поворота.

Вариант 7. Построение 3D-сцены с несколькими объектами

Постройте сцену, содержащую куб, пирамиду и сферу.

Используйте перспективную проекцию для отображения сцены.

Реализуйте возможность перемещения камеры по сцене с помощью клавиатуры.

Дополнительно: Реализуйте возможность приближения и удаления камеры от объектов с плавным изменением угла обзора.

Вариант 8. Построение сцены с несколькими источниками света (без освещения)

Постройте 3D-сцену с несколькими объектами: куб, пирамида и цилиндр.

Примените перспективную проекцию для их отображения.

Визуализируйте несколько источников света как простые сферы (без расчетов освещения(источники света визуализируются как простые объекты (например, сферы или точки), но никакие реальные эффекты освещения сцены не вычисляются)).

Дополнительно: Добавьте возможность перемещения этих "источников света" по сцене.

Вариант 9. Построение объекта с текстурой в 3D-пространстве

Постройте объект (например, куб или пирамиду) и примените к нему текстуру.

Используйте перспективную проекцию для отображения.

Реализуйте вращение объекта и изменение его текстурных координат.

Дополнительно: Добавьте возможность переключения текстур в реальном времени.

Вариант 10. Анимация вращения объекта с ортографической проекцией

Постройте куб или другой простой объект.

Реализуйте его вращение вокруг всех осей в ортографической проекции.

Добавьте плавную анимацию вращения и возможность регулирования скорости анимации.

Дополнительно: Добавьте переключение между ортографической и перспективной проекцией во время анимации.

Вариант 11. Реализация теней для простого объекта

Постройте 3D-куб или пирамиду.

Используя перспективную проекцию, отобразите объект на экране.

Реализуйте примитивное затенение объекта с учетом его положения относительно "источника света" (без шейдеров).

Дополнительно: Реализуйте несколько источников света, влияющих на затенение.

Вариант 12. Построение сцены с двухточечной перспективой

Постройте несколько объектов (куб, пирамиду, цилиндр) и разместите их на сцене.

Реализуйте двухточечную перспективу (с двумя точками схода).

Добавьте возможность перемещения объектов на сцене и наблюдения за изменением их положения относительно точек схода.

Дополнительно: Реализуйте возможность перемещения самих точек схода.

Лабораторная работа 3: Камера и базовые 3D-трансформации

Цель лабораторной работы:

В этой лабораторной работе вы научитесь работать с камерой в 3D-пространстве, управлять её положением и направлением, а также освоите базовые трансформации (перемещение, поворот и масштабирование) объектов в 3D.

Требования:

Вы должны использовать OpenGL + SFML или OpenTK + C#.

Программа должна работать в реальном времени, с возможностью динамической смены положения камеры и трансформаций объектов. **Управление камеры должно быть назначено на клавиатуру или мышь**

Все объекты должны корректно отрисовываться с учетом положения камеры и примененных трансформаций.

Вариант 1. Камера: вращение вокруг объекта

Постройте куб в 3D-пространстве.

Реализуйте камеру, которая вращается вокруг куба по круговой траектории.

Добавьте возможность увеличения и уменьшения радиуса вращения камеры с помощью клавиатуры.

Дополнительно: Реализуйте изменение угла обзора (field of view) для камеры и визуализируйте результат.

Вариант 2. Камера: свободное перемещение в 3D-пространстве

Постройте несколько простых 3D-объектов (кубы, пирамиды, сферы).

Реализуйте камеру, которой можно свободно управлять в 3D-пространстве (перемещение вперед, назад, влево, вправо, вверх, вниз).

Управление камерой должно осуществляться с помощью клавиатуры и мыши.

Дополнительно: Реализуйте "режим полета", когда камера может двигаться свободно в любом направлении.

Вариант 3. Камера: орбитальная камера с управлением мышью

Постройте 3D-сферу.

Реализуйте орбитальную камеру, которая вращается вокруг сферы, но управляется мышью (направление взгляда камеры изменяется перемещением мыши).

Добавьте возможность увеличения и уменьшения расстояния от камеры до сферы.

Дополнительно: Реализуйте ограничение угла наклона камеры, чтобы избежать некорректных положений (например, полное переворачивание камеры).

Вариант 4. Камера: управление через look-at матрицу

Постройте куб в 3D-пространстве.

Реализуйте камеру с использованием матрицы вида (look-at), которая всегда смотрит на куб.

Реализуйте возможность перемещения камеры по круговой траектории вокруг куба, сохраняя взгляд направленным на объект.

Дополнительно: Добавьте возможность изменения высоты камеры, чтобы наблюдать объект с разных углов.

Вариант 5. Масштабирование объекта с учетом позиции камеры

Постройте пирамиду в 3D-пространстве.

Реализуйте возможность масштабирования пирамиды в реальном времени.

При изменении масштаба пирамиды камера должна оставаться неподвижной, но сцена должна корректно отрисовываться.

Дополнительно: Реализуйте ограничение минимального и максимального масштаба объекта.

Вариант 6. Поворот объекта вокруг нескольких осей

Постройте куб.

Реализуйте вращение куба вокруг осей X, Y и Z.

Вращение должно управляться с клавиатуры, при этом оси должны быть независимыми друг от друга.

Дополнительно: Реализуйте плавную анимацию вращения с изменяемой скоростью.

Вариант 7. Комбинация трансформаций (перемещение, масштабирование, поворот)

Постройте несколько объектов (например, куб и пирамиду) в разных местах сцены.

Реализуйте для каждого объекта возможность перемещения, масштабирования и поворота.

Управление должно осуществляться через клавиатуру, при этом трансформации должны применяться последовательно.

Дополнительно: Реализуйте интерфейс для переключения между объектами, чтобы трансформировать их по отдельности.

Вариант 8. Реализация матриц трансформаций

Постройте один 3D-объект (например, сферу).

Реализуйте базовые трансформации (перемещение, масштабирование, поворот) с использованием матричных операций.

Обеспечьте возможность изменения порядка применения матриц (например, сначала поворот, затем перемещение или наоборот).

Дополнительно: Добавьте визуализацию матриц трансформаций на экране в виде текстовой информации.

Вариант 9. Камера: перемещение по заданной траектории

Постройте 3D-сцену с несколькими объектами.

Реализуйте камеру, которая перемещается по заданной траектории (например, по спирали или по окружности).

Обеспечьте плавное движение камеры с возможностью изменения скорости.

Дополнительно: Реализуйте возможность переключения между несколькими траекториями движения камеры.

Вариант 10. Вращение объекта относительно камеры

Постройте куб в 3D-пространстве.

Реализуйте возможность вращения куба относительно положения камеры (то есть объект всегда вращается вокруг точки, на которую смотрит камера).

Управляйте направлением вращения с помощью клавиатуры.

Дополнительно: Реализуйте возможность переключения между вращением вокруг центра объекта и вращением относительно камеры.

Вариант 11. Камера: плавный переход между двумя позициями

Постройте 3D-сцену с несколькими объектами.

Реализуйте возможность плавного перемещения камеры между двумя фиксированными позициями.

Обеспечьте управление скоростью перехода с помощью интерфейса (клавиши или ползунок).

Дополнительно: Реализуйте несколько точек перехода и возможность переключаться между ними.

Вариант 12. Анимация движения объекта с фиксированной камерой

Постройте куб и реализуйте его движение по круговой траектории.

Камера должна оставаться неподвижной, но постоянно следить за кубом.

Обеспечьте возможность изменения скорости движения объекта по траектории.

Дополнительно: Добавьте возможность изменения формы траектории (например, эллиптической).

Лабораторная работа 4: Освещение и работа с шейдерами

Цель лабораторной работы:

В этой лабораторной работе вы научитесь работать с освещением в 3D-пространстве, используя различные типы источников света, и освоите основы написания шейдеров. Вы реализуете освещение объектов в сцене с использованием простейших моделей освещения и настроите эффекты при помощи вершинных и фрагментных шейдеров.

Требования:

Вы должны использовать OpenGL + SFML или OpenTK + C#.

Программа должна корректно отображать освещение с учетом типов источников света, используя написанные вами шейдеры.

В каждом варианте задания должны быть задействованы как минимум один тип освещения (например, направленное освещение, точечный источник света или прожектор).

Вариант 1. Направленное освещение с диффузным компонентом

Постройте куб в 3D-пространстве.

Реализуйте направленное освещение (Directional Light), которое влияет на куб, используя диффузную модель освещения (Lambertian Lighting).

Используйте вершинные и фрагментные шейдеры для расчета освещения на объекте.

Дополнительно: Реализуйте возможность изменения направления источника света в реальном времени.

Вариант 2. Точечный источник света и зеркальные блики

Постройте сферу в 3D-пространстве.

Реализуйте точечный источник света (Point Light), используя модель Фонга для освещения.

Включите зеркальные блики (specular highlights) для более реалистичного отражения света на поверхности объекта.

Дополнительно: Добавьте управление положением источника света с помощью клавиатуры.

Вариант 3. Прожектор с ограниченным углом освещения

Постройте пирамиду.

Реализуйте прожектор (Spotlight), который освещает пирамиду с ограниченным углом света.

Используйте вершинные и фрагментные шейдеры для реализации градиентного затухания света на границах прожектора.

Дополнительно: Реализуйте возможность изменения угла прожектора и его направления в реальном времени.

Вариант 4. Комбинация направленного и точечного освещения

Постройте сцену с несколькими объектами (например, куб, сферу и пирамиду).

Реализуйте направленный источник света и точечный источник света, которые одновременно влияют на все объекты сцены.

Используйте вершинные и фрагментные шейдеры для расчета взаимодействия обоих источников света с объектами.

Дополнительно: Реализуйте возможность динамически переключать типы источников света и изменять их параметры.

Вариант 5. Плоское затенение (Flat Shading)

Постройте куб в 3D-пространстве.

Реализуйте плоское затенение (Flat Shading), при котором каждый полигон объекта имеет один цвет, основанный на нормали грани.

Используйте только вершинный шейдер для выполнения освещения.

Дополнительно: Реализуйте возможность переключения между плоским и гладким затенением (Gouraud Shading).

Вариант 6. Гладкое затенение (Gouraud Shading)

Постройте сферу.

Реализуйте гладкое затенение (Gouraud Shading), при котором освещение рассчитывается на вершинах объекта и затем интерполируется по его поверхности.

Используйте как вершинный, так и фрагментный шейдер для реализации этого эффекта.

Дополнительно: Реализуйте динамическое переключение между Gouraud Shading и Flat Shading для сравнения эффектов.

Вариант 7. Использование нормальных карт для создания детализации

Постройте куб и примените к нему текстуру.

Реализуйте нормальные карты (normal mapping) для добавления детализации на поверхности куба без увеличения количества полигонов.

Используйте фрагментный шейдер для расчета освещения с учетом нормальной карты.

Дополнительно: Добавьте возможность переключения между нормальной картой и стандартным затенением для сравнения результатов.

Вариант 8. Бликовое освещение с контролируемой интенсивностью

Постройте пирамиду в 3D-пространстве.

Реализуйте бликовое освещение (specular lighting) с возможностью управления интенсивностью блика.

Включите параметры контроля материала объекта (например, сила зеркального блика, specular power) через пользовательский интерфейс.

Дополнительно: Добавьте возможность изменения положения источника света и наблюдайте за изменением бликов на поверхности объекта.

Вариант 9. Реализация мягких теней от направленного источника света

Постройте сцену с кубом и сферой.

Реализуйте направленный источник света, который создает мягкие тени (soft shadows) от объектов на сцене.

Используйте фрагментные шейдеры для расчета и сглаживания теней.

Дополнительно: Реализуйте переключение между мягкими и жесткими тенями (hard shadows).

Вариант 10. Многоцветное освещение (Phong Lighting)

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 11. Освещение нескольких объектов с использованием шейдеров

Постройте сцену с несколькими объектами: куб, сферу и пирамиду.

Реализуйте освещение сцены с несколькими источниками света (например, один направленный и один точечный источник).

Используйте шейдеры для расчета освещения на каждом объекте с учетом их индивидуальных нормалей и положения относительно света.

Дополнительно: Реализуйте возможность динамического включения и отключения каждого источника света.

Вариант 12. Прожектор с реализацией физически корректного затухания света

Постройте цилиндр.

Реализуйте прожектор (Spotlight) с физически корректным затуханием света в зависимости от расстояния до объекта.

Используйте фрагментные шейдеры для расчета уровня освещения в зависимости от угла падения и расстояния до прожектора.

Дополнительно: Добавьте возможность изменения коэффициентов затухания для демонстрации различий в освещении.

Лабораторная работа 5: Трассировка лучей (Ray Tracing)

Цель лабораторной работы:

В этой лабораторной работе вы научитесь работать с техникой трассировки лучей для создания реалистичной 3D-графики. Вы реализуете алгоритм Ray Tracing, который позволяет рассчитывать физически корректные отражения, преломления, тени и свет в сцене. Лабораторная работа подводит к пониманию основ рендеринга, работающего с лучами света, а также к созданию реалистичных сцен.

Требования:

Реализуйте алгоритм трассировки лучей для отрисовки простой сцены, используя минимальный набор примитивов (сферы, плоскости и т.д.).

Реализуйте базовые эффекты: отражения, тени и освещение.

Трассировка должна быть реализована как на CPU, так и с возможной оптимизацией на GPU (опционально).

Программа должна корректно отображать сцены в зависимости от выбранного задания.

Вариант 1. Базовая трассировка лучей с отражениями

Постройте сцену с двумя сферами и одной плоскостью, представляющей пол.

Реализуйте трассировку лучей для рендеринга сцены, где свет отражается от поверхности сфер.

Включите зеркальные отражения (mirror reflections) для сфер и пола.

Дополнительно: Реализуйте возможность изменения коэффициента отражения для разных материалов (например, у пола и сфер).

Вариант 2. Трассировка лучей с мягкими тенями

Постройте сцену с одной сферой и одной плоскостью (пол).

Реализуйте направленный источник света, который отбрасывает тени на объект.

Реализуйте мягкие тени (soft shadows) с помощью распределенной трассировки лучей.

Дополнительно: Реализуйте возможность изменения размера источника света, чтобы контролировать степень мягкости теней.

Вариант 3. Преломления и прозрачные объекты

Постройте сцену с двумя сферами — одна из них должна быть прозрачной.

Реализуйте эффект преломления (refraction) для прозрачной сферы с использованием закона Снелля.

Включите управление коэффициентом преломления для настройки уровня преломления света внутри объекта.

Дополнительно: Реализуйте возможность изменения показателя преломления для создания различных материалов (например, стекло, вода).

Вариант 4. Множественные отражения и преломления

Постройте сцену с тремя сферами и одной плоскостью.

Реализуйте трассировку лучей, чтобы каждый объект мог как отражать, так и преломлять свет.

Включите обработку множественных отражений и преломлений, где лучи света могут многократно взаимодействовать с объектами сцены.

Дополнительно: Реализуйте контроль глубины трассировки для оптимизации производительности.

Вариант 5. Реалистичное глобальное освещение

Постройте сцену с кубом и сферой.

Реализуйте глобальное освещение (Global Illumination) с помощью трассировки лучей.

Обеспечьте взаимодействие объектов со светом, отскакивающим от других поверхностей (интерфлексии света).

Дополнительно: Реализуйте управление количеством "прямых" и "косвенных" лучей для точной настройки уровня освещения.

Вариант 6. Поверхностное затенение (Phong Shading)

Постройте сцену с одной сферой и плоскостью.

Реализуйте трассировку лучей с поддержкой модели затенения Фонга (Phong Shading), которая учитывает диффузное, зеркальное и окружающее освещение.

Используйте нормали для точного расчета освещения на поверхности объектов.

Дополнительно: Реализуйте возможность изменения параметров модели освещения Фонга для демонстрации различных эффектов.

Вариант 7. Отражения и текстурированные поверхности

Постройте сцену с двумя текстурированными плоскостями (стена и пол) и одной сферой.

Реализуйте трассировку лучей с поддержкой текстурирования объектов.

Включите отражения на сфере и учтите отраженные текстуры на её поверхности.

Дополнительно: Реализуйте управление зеркальностью поверхности сферы для изменения интенсивности отражений.

Вариант 8. Трассировка лучей с распределенным освещением

Постройте сцену с тремя сферами и одной плоскостью (пол).

Реализуйте распределённое освещение (Distributed Lighting) с трассировкой лучей для создания более реалистичных эффектов, таких как блёстки или шероховатости на поверхностях.

Обеспечьте случайное распределение лучей для симуляции эффектов шероховатых поверхностей.

Дополнительно: Реализуйте управление количеством случайных выборок лучей для настройки уровня детализации освещения.

Вариант 9. Трассировка лучей с каустиками (Caustics)

Постройте сцену с прозрачной сферой и плоскостью.

Реализуйте каустики — световые эффекты, создаваемые преломляющимся светом, проходящим через прозрачные объекты.

Включите трассировку лучей для корректного рендеринга световых паттернов на плоскости, возникающих из-за преломления и отражения света.

Дополнительно: Реализуйте управление параметрами прозрачности и коэффициентами преломления для настройки эффекта каустики.

Вариант 10. Путь света с участием объёмных эффектов (Volumetric Light)

Постройте сцену с двумя сферами и одной плоскостью.

Реализуйте трассировку лучей с поддержкой объёмных эффектов (например, свет, рассеивающийся в тумане или дыме).

Обеспечьте симуляцию рассеяния и поглощения света внутри объёмных объектов.

Дополнительно: Реализуйте возможность изменения плотности и цвета объёмных объектов для демонстрации различных эффектов.

Вариант 11. Эффект глубины резкости (Depth of Field)

Постройте сцену с тремя сферами, расположенными на разном расстоянии от камеры.

Реализуйте трассировку лучей с эффектом глубины резкости (Depth of Field), чтобы объекты на переднем плане были в фокусе, а объекты на заднем плане – размыты.

Настройте фокусное расстояние и глубину поля для создания реалистичных эффектов камеры.

Дополнительно: Реализуйте управление параметрами камеры для изменения фокуса в реальном времени.

Вариант 12. Многопроходная трассировка лучей с отражением и преломлением

Постройте сцену с несколькими объектами (кубы и сферы).

Реализуйте многопроходную трассировку лучей, чтобы комбинировать эффекты отражений и преломлений для каждого объекта.

Включите множественные источники света для более сложной сцены, где каждый объект взаимодействует с отражениями и преломлениями от других объектов.

Дополнительно: Реализуйте возможность настройки глубины трассировки для управления качеством изображения.

Лабораторная работа 6: Разработка простого игрового движка

Цель лабораторной работы:

В этой лабораторной работе вы будете разрабатывать основу для собственного игрового движка. Это завершающая работа, в которой вы объедините все изученные техники работы с 2D и 3D графикой, освещением, шейдерами и трассировкой лучей. В результате выполнения лабораторной работы у вас должен получиться простой движок, который может отрисовывать сцены в реальном времени, с возможностью работы с камерой, объектами и освещением.

Основные задачи:

1. Создать базовую архитектуру игрового движка с поддержкой 3D-сцен.
2. Реализовать систему рендеринга с использованием изученных подходов: рендеринг через растризацию и трассировку лучей.
3. Настроить камеру и систему управления ею.
4. Реализовать поддержку базовых игровых объектов и взаимодействие с ними.
5. Обеспечить работу с несколькими типами источников света.
6. Оптимизировать производительность движка (по возможности).

Этап 1. Архитектура игрового движка

1. Инициализация системы:

Создайте основную структуру движка. Включите в нее следующие компоненты:

- Рендеринг: модуль, отвечающий за отрисовку сцен.
- Физика (опционально): простой модуль для расчета столкновений объектов (например, сферы и плоскости).
- Игровая логика: управление взаимодействием объектов сцены и правил игры.

- Загрузчик файлов формата OBJ для добавления их на сцену
2. Менеджер сцены:

Разработайте систему для управления игровыми сценами. Сцена должна содержать:

- Объекты (3D-модели).
- Источники света.
- Камеру.

Этап 2. Рендеринг сцены

1. Рендеринг через растризацию:

Реализуйте базовый алгоритм отрисовки сцены с использованием растризации. Включите поддержку:

- Основных 3D-примитивов (сферы, кубы, плоскости и т.д.).
- Затенения (Flat или Gouraud shading).
- Простого освещения: направленный источник света и точечные источники света.

2. Трассировка лучей (Ray Tracing):

Добавьте возможность рендеринга сцены через трассировку лучей. Это может быть отдельный режим движка. Обеспечьте:

- Поддержку отражений и теней.
- Простые материалы (зеркальные и матовые).
- Реализацию хотя бы одного эффекта: преломления или каустики (опционально).

3. Шейдеры:

Напишите простые вершинные и фрагментные шейдеры для объектов сцены. Реализуйте:

- Диффузное и зеркальное освещение.
- Управление цветами материалов через шейдеры.

Этап 3. Работа с камерой

1. Базовое управление камерой:

Реализуйте систему управления камерой, которая позволяет:

- Перемещаться по сцене (WASD или стрелки).
- Изменять угол обзора (вращение камеры мышью).
- Масштабировать изображение (например, через колесо мыши).

2. Проекции:

Реализуйте возможность переключения между ортогографической и перспективной проекциями для камеры.

Этап 4. Работа с объектами и освещением

1. Игровые объекты:

Реализуйте возможность добавления объектов в сцену через код или файл конфигурации. Объекты должны иметь:

- Позицию, вращение, масштаб.
- Взаимодействие с источниками света (освещаться).

2. Освещение:

Настройте работу нескольких типов источников света:

- Направленный свет (Directional Light): влияет на всю сцену.
- Точечный свет (Point Light): создаёт затухающее освещение, которое зависит от расстояния до объекта.
- Прожектор (Spotlight): освещает только определенную область сцены.

3. Тени:

Добавьте поддержку теней для объектов сцены. Это могут быть простые тени для начала, например, только от одного источника света (опционально).

Этап 5. Оптимизация и улучшение производительности

1. Уменьшение количества рендеринговых вызовов:

Постарайтесь уменьшить количество вызовов отрисовки для повышения производительности движка.

2. Ограничение глубины трассировки лучей:

Если вы используете трассировку лучей, реализуйте ограничение глубины лучей для ускорения работы.

3. Профилирование производительности:

Реализуйте базовое профилирование кадров (FPS) и времени рендеринга для отслеживания производительности движка.

Этап 6. Разработка простого интерфейса

1. Игровой интерфейс:

Реализуйте базовый интерфейс (HUD), который показывает информацию о текущем состоянии игры, например, FPS или количество объектов в сцене.

2. Меню настроек:

Сделайте меню, где можно будет изменять параметры графики (например, качество теней, включение/отключение трассировки лучей).

Финальная задача: создание мини-игры

Используя созданный вами игровой движок, реализуйте простую игру. Она должна включать:

- Сцену с несколькими объектами (например, движущиеся или статичные препятствия).
- Управление игроком (например, от первого лица с возможностью перемещения камеры).
- Взаимодействие с объектами (например, сбор предметов или избегание препятствий).