

# Курсовая работа, 4 семестр

[Общая формулировка](#)

[Основные этапы](#)

[Этап 1, простейший](#)

[Этап 2, точечное освещение](#)

[Этап 3, "гном"](#)

[Этап 4, зеркало](#)

[Этап 5, прозрачное стекло](#)

[Этап 6, сложное освещение](#)

[Этап 7, арифметика объектов](#)

[Этап 8, текстуры](#)

[Бонусные баллы](#)

[Многоголовый](#)

[Самый быстрый](#)

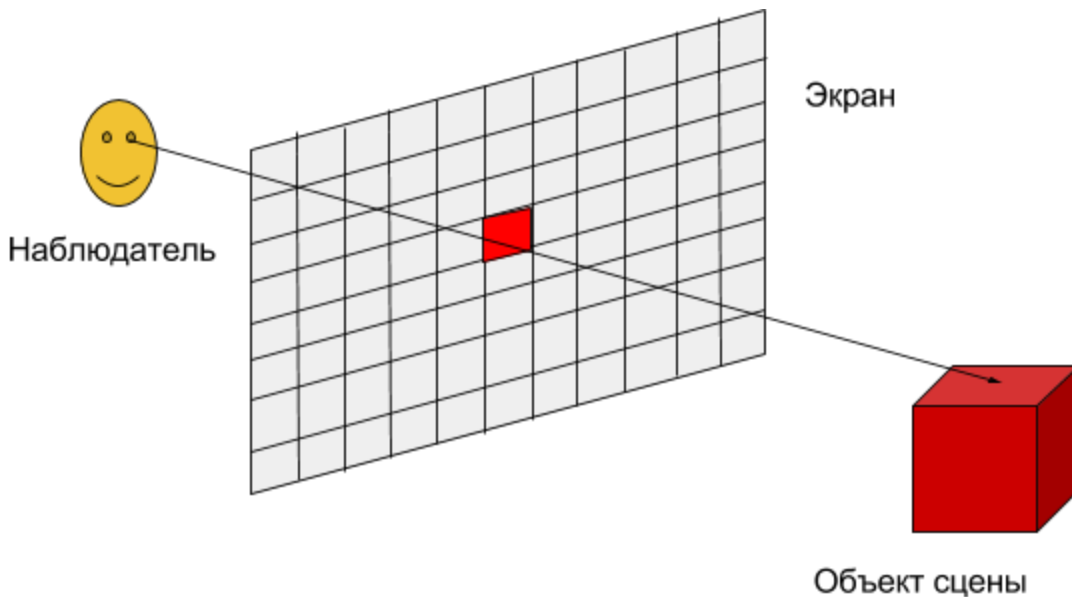
[Самый компактный](#)

[Красивый](#)

[Дружелюбный](#)

## Общая формулировка

При визуализации трехмерных сцен на компьютере часто используется техника ray-tracing. Её суть заключается в следующем:



Из “глаза” виртуального наблюдателя пускается луч в каждый пиксель отображаемой картинке. Ищется объект, лежащий на продолжении этого луча, пиксель заполняется цветом этого объекта.

Задача заключается в написании утилиты, способной по списку:

1. Координат объектов и их атрибутов
2. Координат углов виртуального экрана и количества пикселей по вертикали и горизонтали
3. Координат “глаза” наблюдателя

построить и отобразить соответствующую картинку. За решение каждого этапа начисляются баллы.

Разрешается использовать только библиотеки STL и Boost. Запрещено использовать библиотеки обработки 3d графики. Из функций рисования на окне/фрейме/etc разрешено использовать только функцию очистки области и установки цвета точки.

## Основные этапы

### Этап 1, простейший

Программа должна выводить картинку без учета освещенности, из-за этого рассмотреть грани объектов будет невозможно. Входные данные берутся из произвольного формата. Должны быть поддержаны следующие примитивы:

1. Сфера
2. Произвольный треугольник
3. Произвольный плоский четырехугольник

Объекты должны поддерживать атрибут “цвет”. Для реализации примитива “четыреугольник” запрещено использовать его представление в виде списка треугольников.

Для треугольника и четырехугольника необходимо иметь возможность задавать нормаль, выделяющую “лицевую” и “изнаночную” стороны объекта.

### Этап 2, точечное освещение

Добавить сущность “точечный источник света” с атрибутом “интенсивность”. Источник света должен влиять на цвет отображаемой точки согласно [законам физики](#):

1. Интенсивность света обратно пропорциональна квадрату расстояния
2. Интенсивность света прямо пропорциональна косинусу угла нормали к поверхности и луча до источника

Не забывайте, что между источником света и точкой на объекте могут лежать другие объекты, создающие эффект тени.

*Примечание.*

Можно добавить “фоновое” освещение, дающее равномерную подсветку всех объектов с одинаковой интенсивностью.

### Этап 3, “гном”

<http://lurkmo.re/%D0%93%D0%BD%D0%BE%D0%BC> Скорость отрисовки одного пикселя гнома с бутылочкой не должна линейно зависеть от количества полигонов (треугольников) в этой бутылочке, если бутылочка не может быть отрисована с данного ракурса, т.е. положения экрана и наблюдателя. Другими словами, необходимо уметь эффективно отсекал невидимые области..

### Этап 4, зеркало

Добавить к объектам атрибут “отражающая способность”, принимающий значение от 0 до 100. Если отражающая способность больше 0, то цвет точки объекта должен складываться с цветом отражения в нём. Использовать формулу для alpha-blend:

$$\text{DstColor}_{\text{channel}} = X_{\text{channel}} * \alpha + Y_{\text{channel}} * (1 - \alpha), \text{ где}$$

DstColor - результирующий цвет смешивания,

X, Y - цвет точек, соответственно, отражения и объекта

channel - компонента цвета точки - {red | green | blue},

alpha - отражающая способность, приведенная к интервалу 0..1

### Этап 5, прозрачное стекло

Добавить к объектам атрибут “преломляющая способность”, принимающий значение от 0 до  $\infty$ . Если значение больше 0, то использовать его как показатель  $\frac{n_1}{n_2}$  в [законе](#)

[Снеллиуса](#), при этом  $n_1$  описывает коэффициент преломления среды, лежащей в лицевой части объекта,  $n_2$  - в изнаночной (см. Общую формулировку).

*Примечание 1.*

Значения 0..1 описывают попадание луча из воздушной среды, лежащей в лицевом полупространстве, в среду с коэффициентом, большим 1, лежащей в изнаночном полупространстве.

*Примечание 2.*

Вводить понятие объема или среды, в которой распространяет луч, не обязательно. Достаточно рассчитывать преломление луча при прохождении через поверхность объекта.

*Примечание 3.*

Преломляющая способность может быть задана для объекта совместно с отражающей способностью.

## Этап 6, сложное освещение

Каждый объект, имеющий ненулевую отражательную способность, должен выступать в качестве источника вторичного освещения. Т.е. красный шар должен добавлять красный оттенок на лежащую поблизости плоскость.

Глубина рекурсии - единица, т.е. источники вторичного освещения не должны порождать источники "третичного" освещения.

## Этап 7, арифметика объектов

Добавить возможность объединять, пересекать и вычитать объемные объекты (параллелограмм и сфера).

## Этап 8, текстуры

Добавить возможность указать для плоских объектов файл, содержащий изображение, накладываемое на плоскость объекта.

## Бонусные баллы

Дополнительные подзадачи, добавляющие итоговые баллы.

### Многоголовый

Выполнять отрисовку в многопоточном режиме.

### Самый быстрый

Выполнять отрисовку как минимум на 10% быстрее остальных.

### Самый компактный

Использовать максимум 90% байт оперативной памяти от используемой памяти любого другого студента.

## Красивый

Реализовать алгоритм anti-aliasing для сглаживания изображения

## Дружелюбный

Реализовать импорт данных напрямую из файлов описываемого ниже формата.

### Формат rt.

Файл состоит из секций, содержащих данные одного типа. Все ключевые слова формата состоят из одного слова (образованного, возможно, двумя английскими). Для описания числовых параметров используются числа с плавающей точкой, при этом десятичным разделителем является точка. Для комментариев используется символ “#”. Файл хранится в кодировке UTF-8.

Секция viewport описывает положение глаза наблюдателя и координаты экрана. Экран можно считать прямоугольным, параллельность его границ осям координат предполагать не следует.

```
# Описание экрана
viewport
    origin x y z          # Положение глаза наблюдателя
    topleft x y z         # Верхний-левый угол экрана
    bottomleft x y z      # Нижний-левый угол экрана
    topright x y z        # Верхний-правый угол экрана
endviewport
```

Далее идет список используемых материалов. Он задает соответствие основных параметров некоторому имени, которое в дальнейшем используется при описании геометрии. Каждый отдельный материал заключается между сроками entry и endentry. Ограничение на имя совпадает с таковым для идентификаторов C++. Цвета задаются в виде тройки RGB с диапазонами компонент 0..255.

```
# Описание материалов
materials
    entry
        name <cpp-like-id>    # Имя
        color <RGB>           # Цвет поверхности
        # Коэффициент прозрачности (1-непрозрачный)
        alpha <0..1>
        reflect <0..1>         # Коэффициент отражения
        refract <0..+∞>        # Коэффициент преломления
    endentry
    ...
```

```
endmaterials
```

Описание источников света. Используются только точечные источники света. Также в данном узле описываются нормировочные коэффициенты - источник свет заданной мощности обеспечивает такую освещенность ближайшей точки плоскости на заданном расстоянии, что её цвет, отображаемый на экране, в точности соответствует цвету материала (не светлее и не темнее).

```
# Описание освещения сцены
```

```
lights
```

```
    # Описание нормировочных коэффициентов
```

```
    reference
```

```
        # Мощность эталонного источника
```

```
        power p
```

```
        # Расстояние от него до эталонной плоскости
```

```
        distance d
```

```
    endreference
```

```
    # Описание одного источника света
```

```
    point
```

```
        # Координаты
```

```
        coords x y z
```

```
        # Мощность
```

```
        power p
```

```
    endpoint
```

```
    ...
```

```
endlights
```

Описание геометрии. Каждый из примитивов содержит ссылку на материал, из которого он состоит. При указании вершин 2d примитивов используется обход против часовой стрелки, если смотреть на примитив со стороны лицевой поверхности.

```
# Описание примитивов
```

```
geometry
```

```
    sphere
```

```
        coords x y z
```

```
        radius r
```

```
        material <id>
```

```
    endsphere
```

```
    ...
```

```
    triangle
```

```
        vertex x y z
```

```
        vertex x y z
```

```
        vertex x y z
```

```
        material <id>
```

```
    endtriangle
```

```
    ...
```

```
    quadrangle
      vertex x y z
      vertex x y z
      vertex x y z
      vertex x y z
      material <id>
    endquadrangle
  ...
endgeometry
```