

3D-RENDERER С НУЛЯ

Тип проекта: Программный

Выполнил: Бонич Дмитрий, БПМИ 213

Научный руководитель: Трушин Дмитрий Витальевич, ФКН ВШЭ

3D рендеринг – процесс выдающий двумерное изображение по некоторому описанию трехмерного объекта.

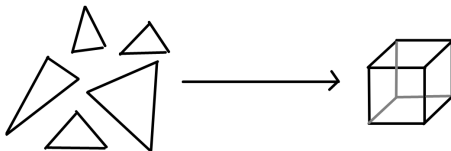
Цель работы и задачи

Цель: Написать программу способную осуществлять 3D-рендеринг.

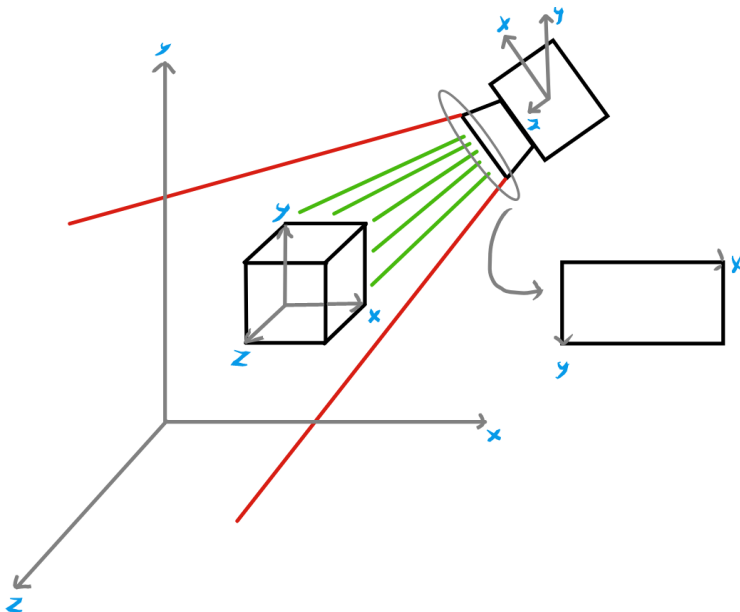
Задачи:

- Отрисовка 3d-объектов
- Загрузка моделей из .obj формата
- Освещение

Любой 3d-объект будем аппроксимировать треугольниками.

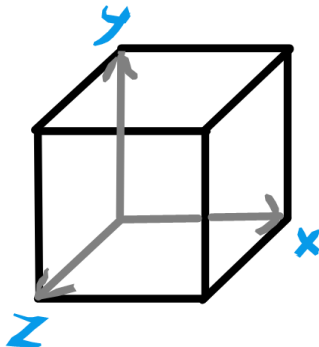


Значит теперь мы хотим научиться рендерить 1 треугольник.



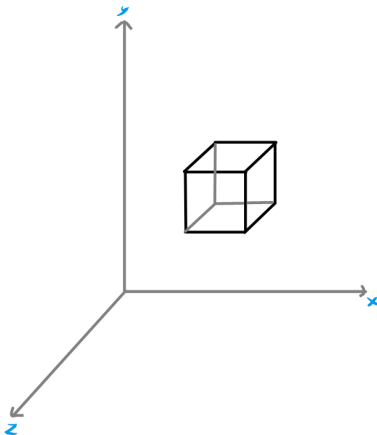
Local space

Local space – персональное для каждого объекта пространство в котором нам изначально приходят координаты объекта (например из .obj файла).



World space

В реальности у нас как правило несколько объектов входящих в сцену. Их расположение относительно друг друга задается в world space системе координат.



Преобразования координат

Используем 4 координаты для задания точки: $\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$.

Это позволяет нам делать аффинные преобразования с помощью матриц. Если $w = 1$, то можно сделать сдвиг следующим образом:

$$\begin{pmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + \Delta x \\ y + \Delta y \\ z + \Delta z \\ 1 \end{pmatrix}$$

Помимо сдвигов можно делать обычные линейные операции: повороты, растяжения.

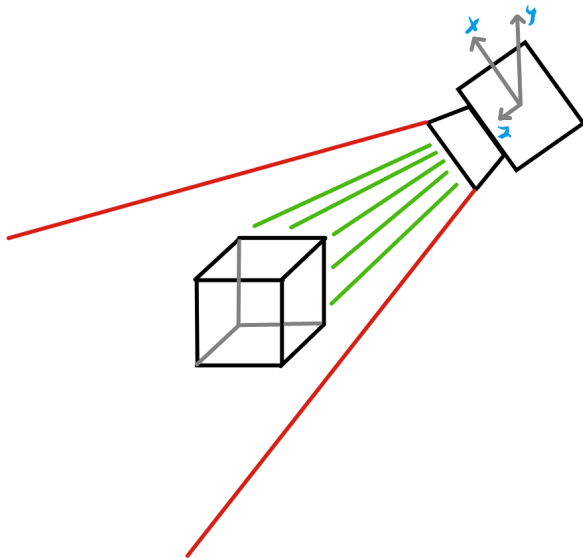
Хотим получить расположение объектов относительно камеры.
Это сдвиг + ортогональная замена базиса.

$$M_{view} = \begin{pmatrix} r_x & r_y & r_z & 0 \\ u_x & u_y & u_z & 0 \\ -d_x & -d_y & -d_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

p – позиция камеры

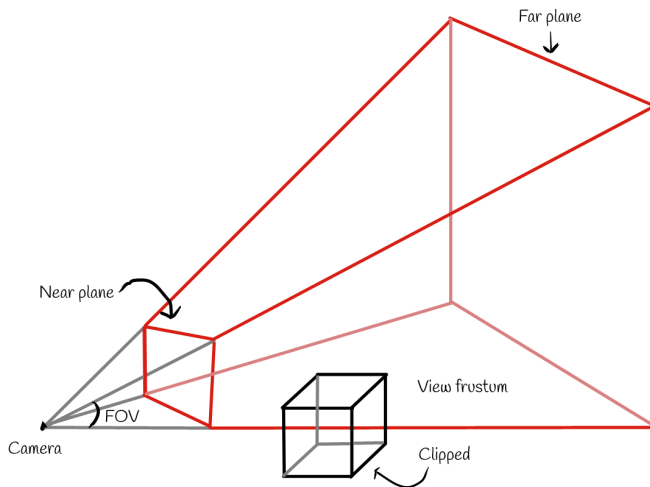
r, u, d – право, верх и направление камеры.

View space



Clip space

Перспективная проекция + обрезка(clipping)

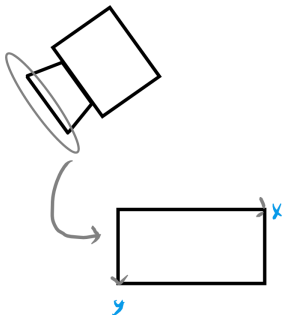


Screen space

Переходим к пикселям. Целые координаты означают центр пикселя.

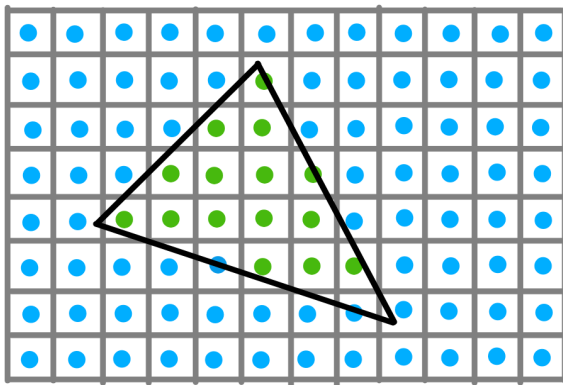
$$M_{screen} = \begin{pmatrix} w' & 0 & 0 & w' \\ 0 & -h' & 0 & h' \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

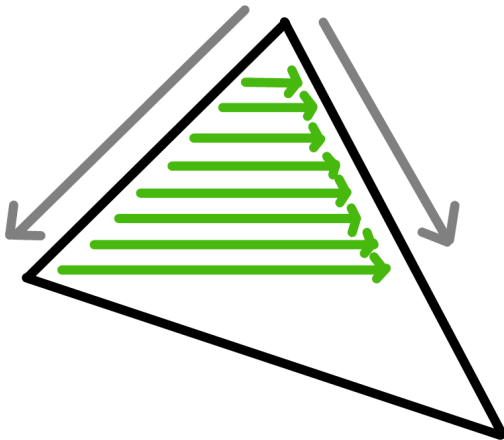
w' , h' – половинки ширины и высоты окна.

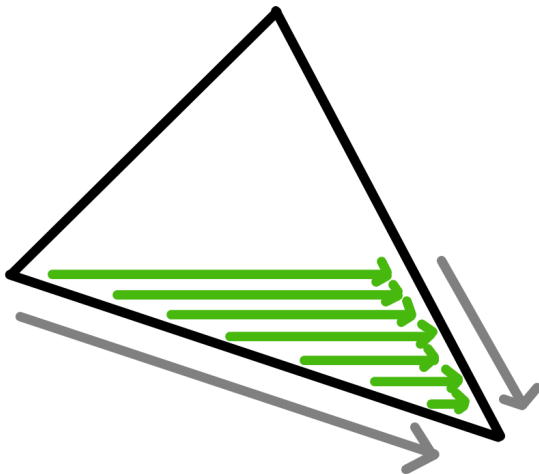


Растеризация

Хотим понять какие пиксели лежат внутри треугольника и их нужно закрасить, а какие нет.



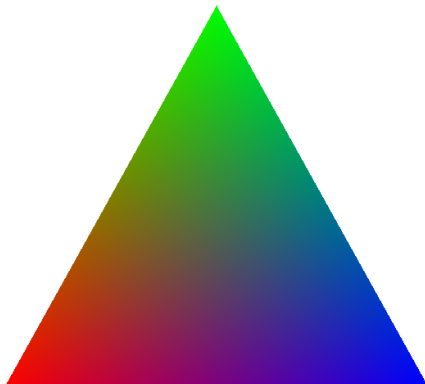




Сохраняем глубину записанного пикселя в буфер и перезаписываем цвет только если глубина уменьшилась.

Линейная интерполяция

Достаточно посчитать градиент линейной функции от координат x и y .



Линейная интерполяция

Проблема: зависимость интерполянта от screen space координат не линейная.

Решение: для интерполянта f будем интерполировать значение $\frac{f}{z}$, а также будем интерполировать $\frac{1}{z}$.

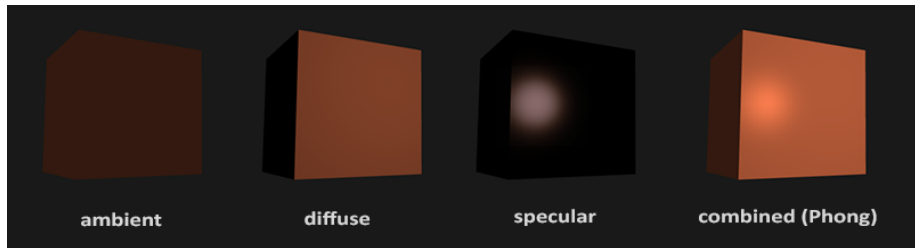
Точечный свет

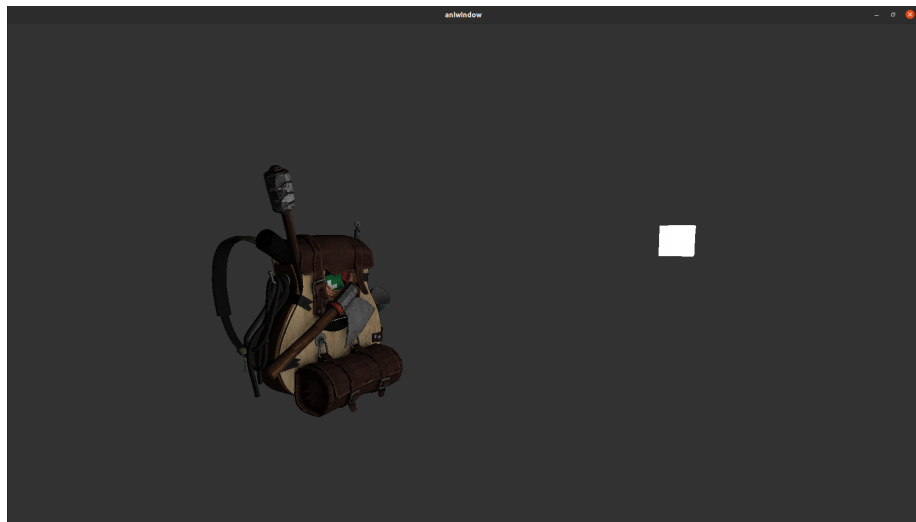
В качестве источника света используется точечный свет.



Phong Shading Model

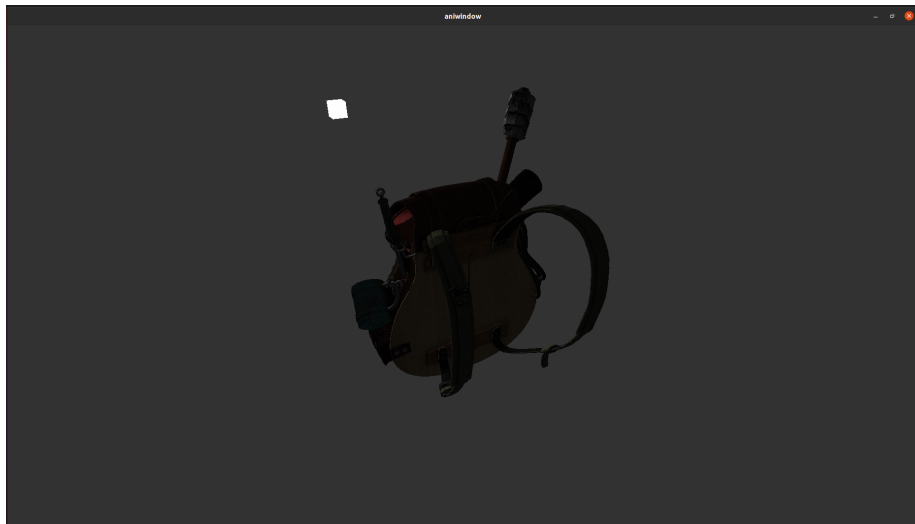
Ambient + diffuse + specular







Результаты



Спасибо за внимание!