Построение трехмерной сцены объектов

Студент:

- Фролов Евгений ИУ7-55Б

Научный руководитель:

- Романова Т.Н.

Москва 2021

Цель работы - разработать программу для построения реалистического изображения из трехмерных геометрических объектов

Реализованное ПО предоставляет возможность:

а) выбора и добавления в сцену трехмерных объектов;

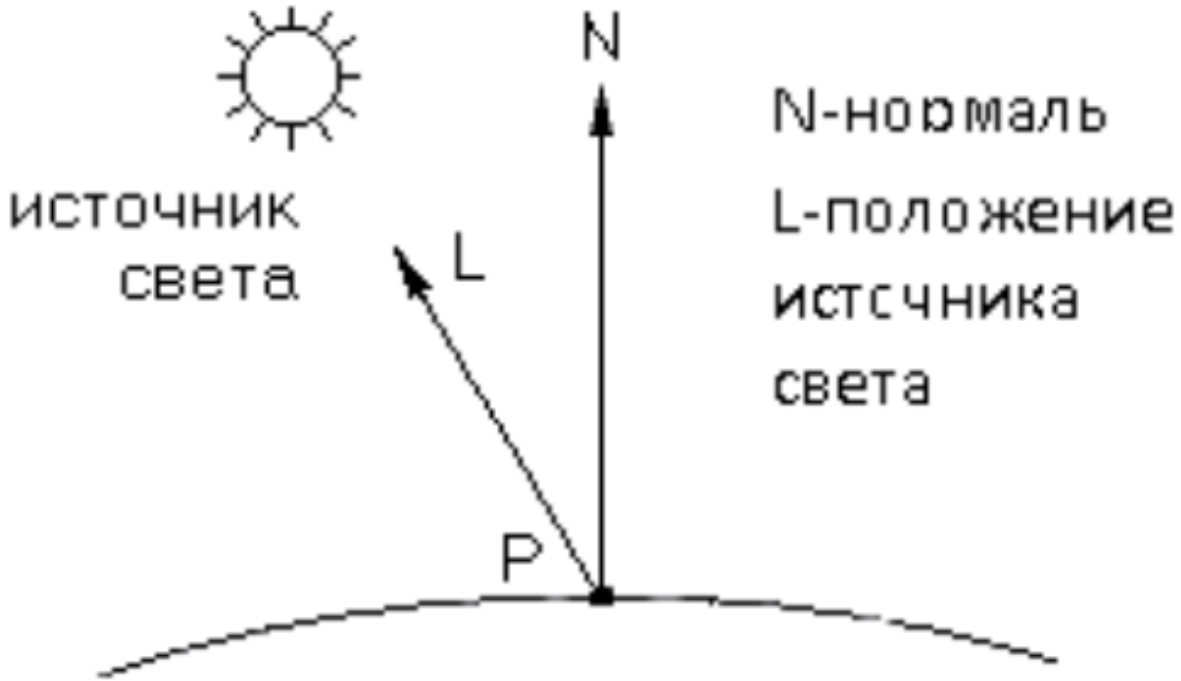
б) перемещения, поворота и масштабирования объектов;

в) изменение текстуры объекта, его цвет, свойств поверхности.

Формализация сцены

- Трехмерные геометрические объекты;
- источник света;
- отражение фигур;
- текстуры фигуры.

Простой метод освещения



В простом методе освещения интенсивность рассчитывается по закону Ламберта:

$$I = I_0 * \cos(\alpha), \text{ где}$$

- I – результирующая интенсивность света в точке
- I_0 – интенсивность источника
- α – угол между нормалью к поверхности и вектором направления света

Алгоритм Z-буфера

1. Всем элементам буфера кадра присвоить фоновое значение

2. Инициализировать Z буфер минимальными значениями глубины

3. Выполнить растровую развертку каждого многоугольника сцены:

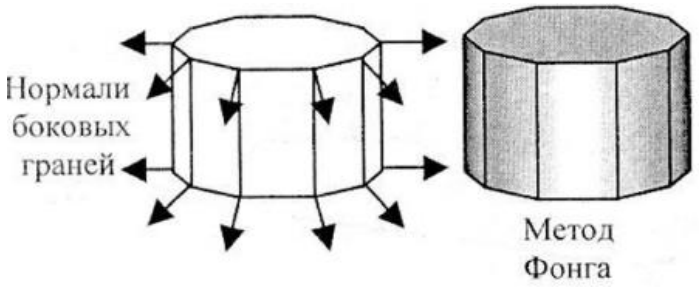
    a. Для каждого пикселя, связанного с многоугольником вычислить его глубину z(x, y)

    b. Сравнить глубину пискселя со значением, хранимым в Z буфере.

        Если z(x, y) > zбуф(x, y), то zбуф(x,y) = z(x,y), цвет(x, y) = цветПикселя.

4. Отобразить результат

Процесс закраски по методу Фонга



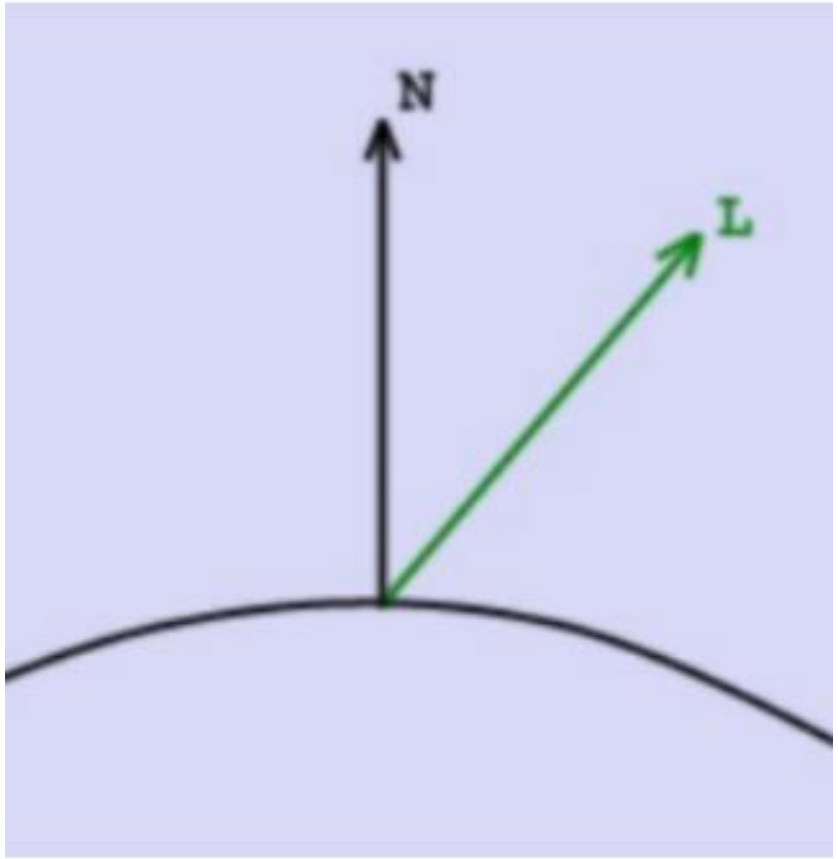
1. Определяются нормали к граням.
2. По нормалям к граням определяются нормали в вершинах.
3. В каждой точке закрашиваемой грани определяется интерполированный вектор нормали.
4. По направлению векторов нормали определяется цвет точек грани.

Модель Ламберта

Модель Ламберта моделирует идеальное диффузное освещение. Свет при попадании на поверхность рассеивается равномерно во все стороны.

Учитывается:

- ориентация поверхности (нормаль N)
- направление на источник света (вектор L).



Выбор языка программирования и среды разработки

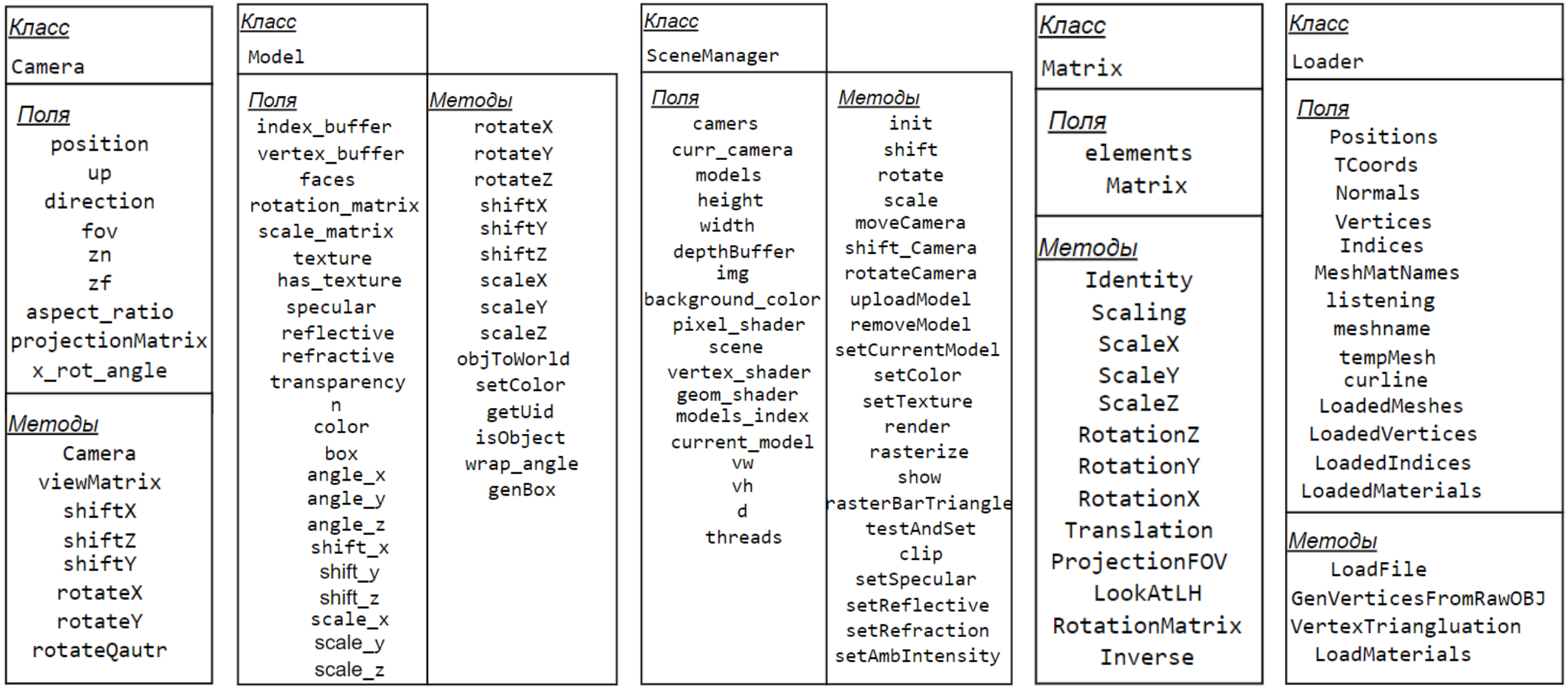
В качестве языка программирования был выбран язык C++:

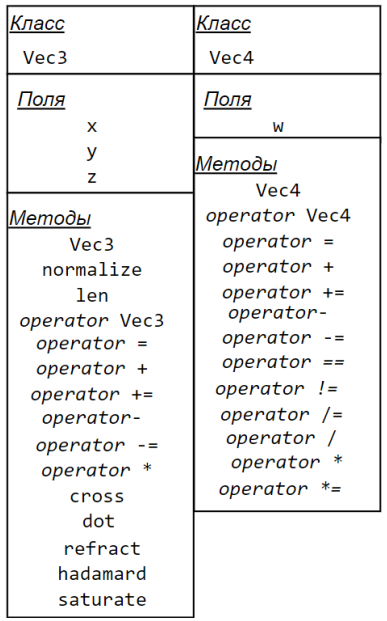
- т.к. работал с этим языком во время курса по предмету ООП.

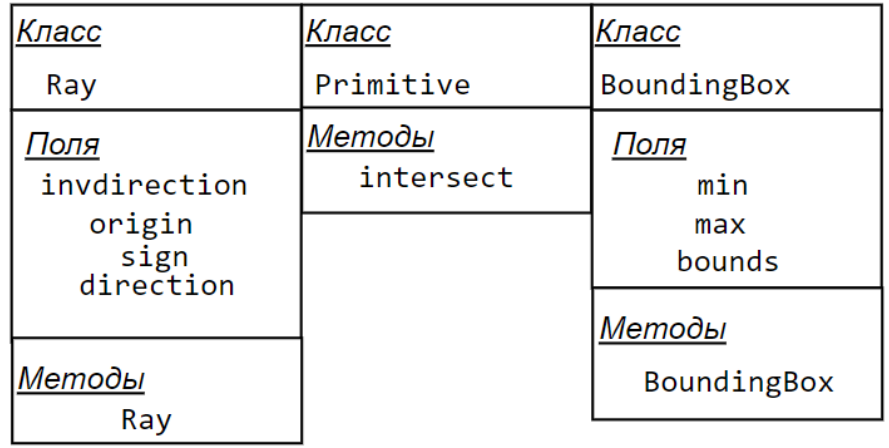
В качестве среды разработки была выбрана QtCreator:

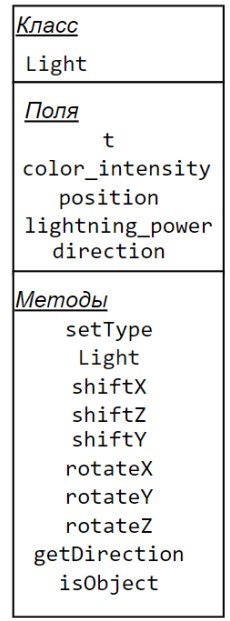
- бесплатный для студентов;
- удобства отладки и написания кода;
- удобное создание интерфейса.

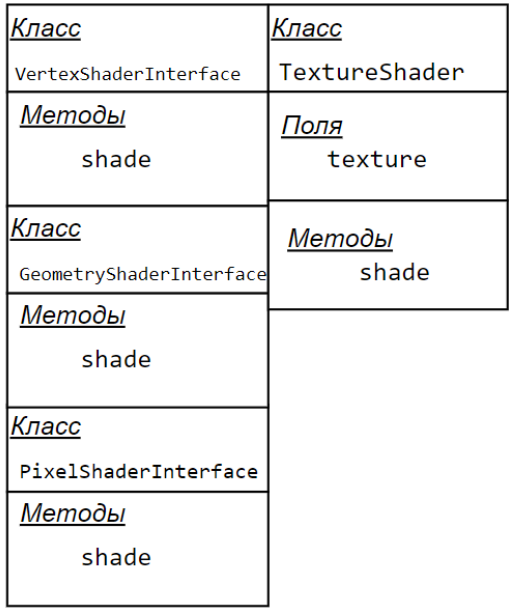
Структура и состав классов

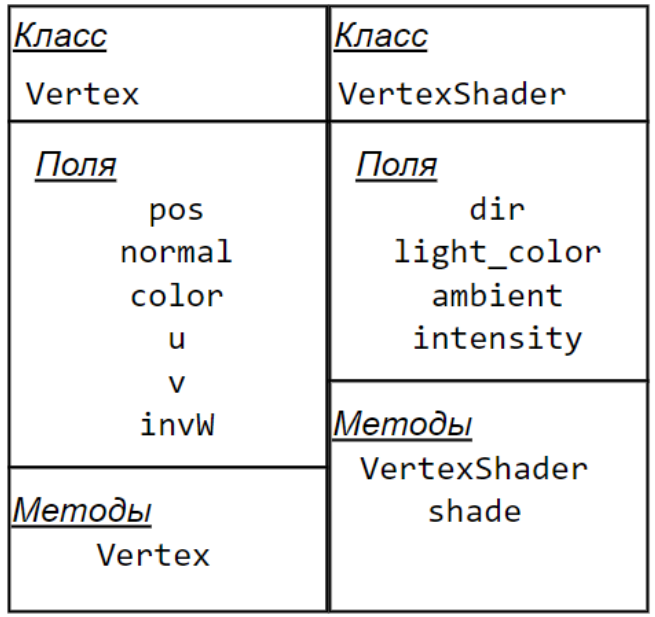


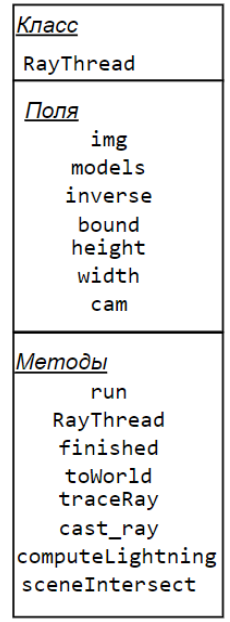

Класс
Vec3
Поля
x
y
z
Методы
Vec3
normalize
len
operator Vec3
operator =
operator +
operator +=
operator-
operator -=
operator *
cross
dot
refract
hadamard
saturate
Класс
Vec4
Поля
w
Методы
Vec4
operator Vec4
operator =
operator +
operator +=
operator-
operator -=
operator ==
operator !=
operator /=
operator /
operator *
operator *=


Класс
Ray
Поля
invdirection
origin
sign
direction
Методы
Ray
Класс
Primitive
Методы
intersect
Класс
BoundingBox
Поля
min
max
bounds
Методы
BoundingBox

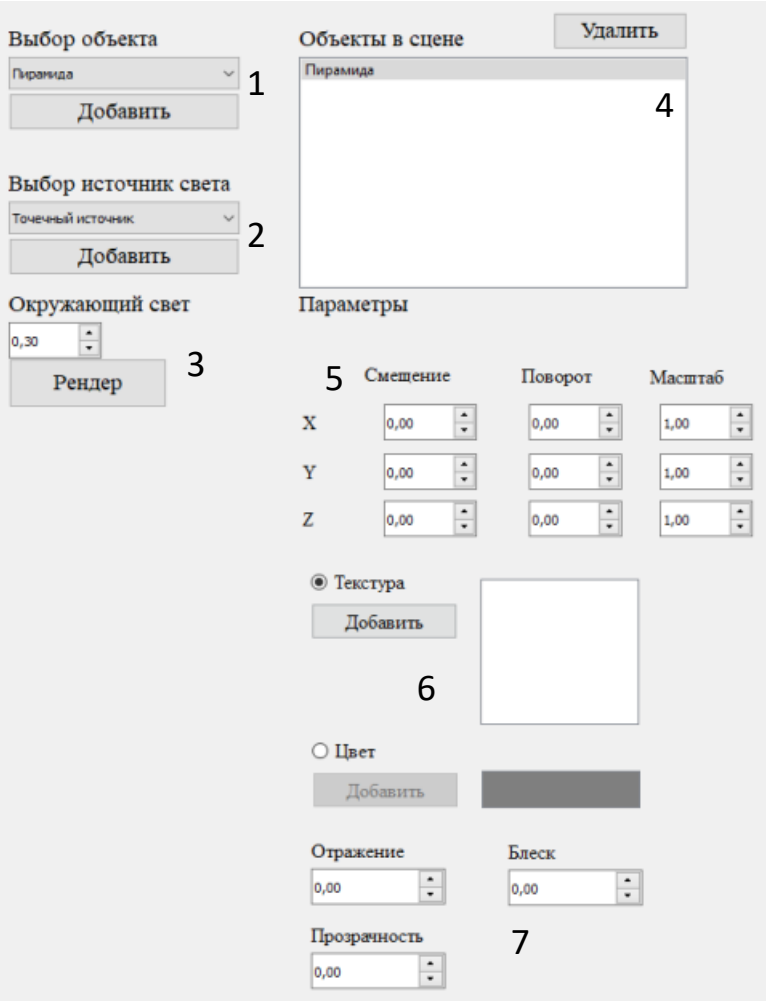

Класс
Light
Поля
t
color_intensity
position
lightning_power
direction
Методы
setType
Light
shiftX
shiftZ
shiftY
rotateX
rotateY
rotateZ
getDirection
isObject


Класс
VertexShaderInterface
Методы
shade
Класс
GeometryShaderInterface
Методы
shade
Класс
PixelShaderInterface
Методы
shade
Класс
TextureShader
Поля
texture
Методы
shade


Класс
Vertex
Поля
pos
normal
color
u
v
invW
Методы
Vertex
Класс
VertexShader
Поля
dir
light_color
ambient
intensity
Методы
VertexShader
shade


Класс
RayThread
Поля
img
models
inverse
bound
height
width
cam
Методы
run
RayThread
finished
toWorld
traceRay
cast_ray
computeLightning
sceneIntersect

Интерфейс программы



1. Выбор объекта - добавление объектов в сцену

2. Выбор источника света

3. Изменение параметра окружающего света

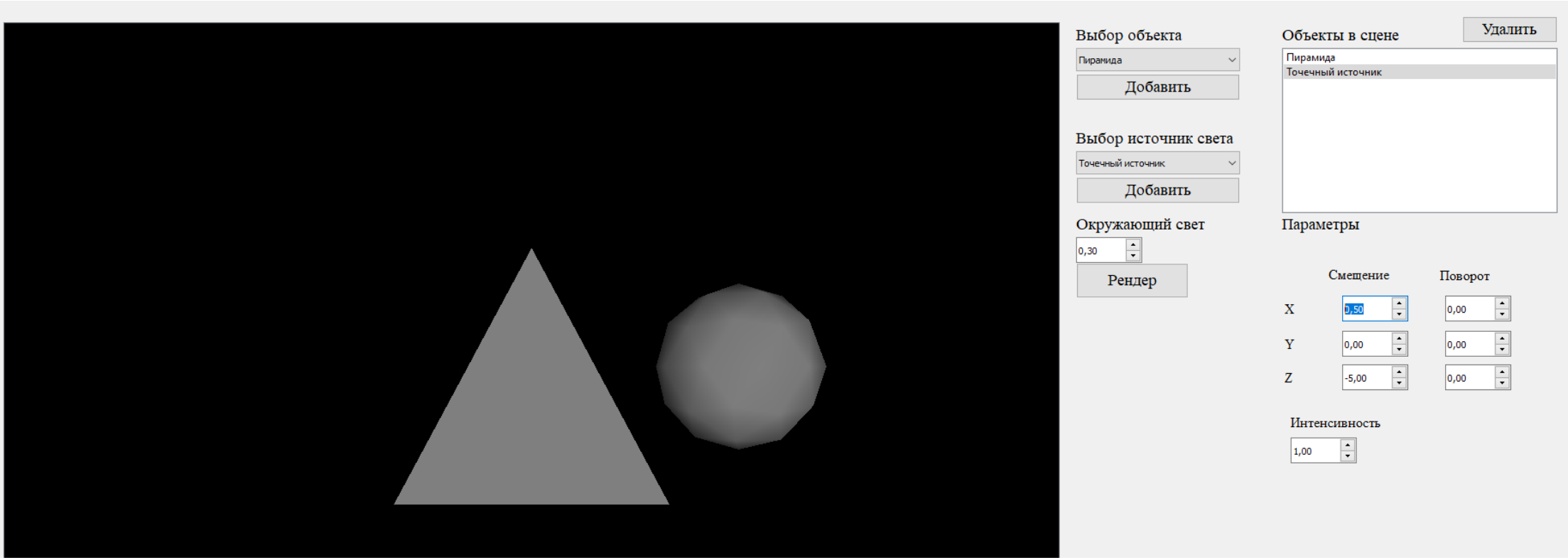
4. Просмотр объектов сцены

5. Изменение положения объекта

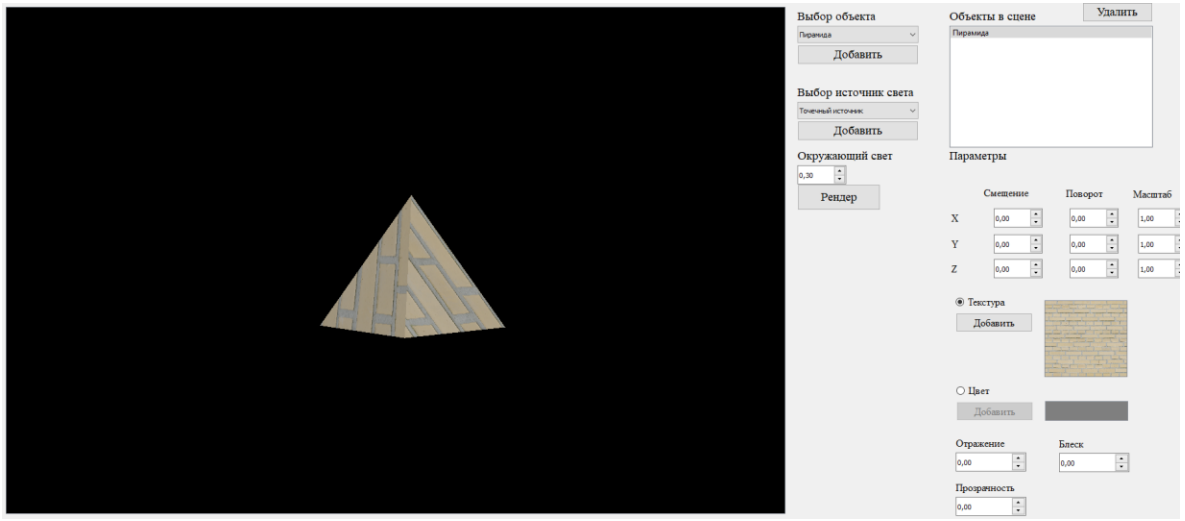
6. Добавление текстуры и изменение цвета объекта

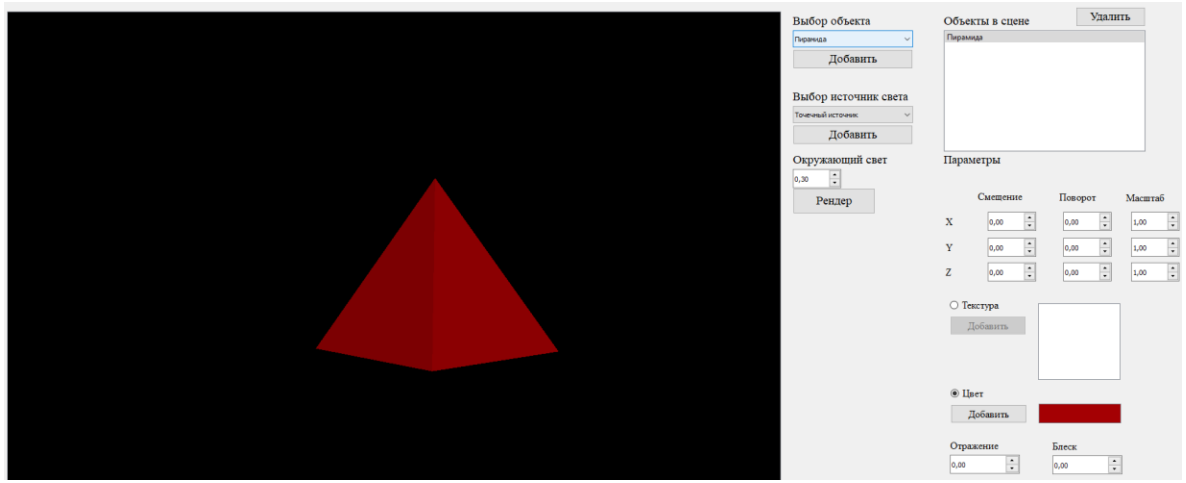
7. Изменение параметров отражения, блеска и прозрачности

На сцену добавили пирамиду и источник света

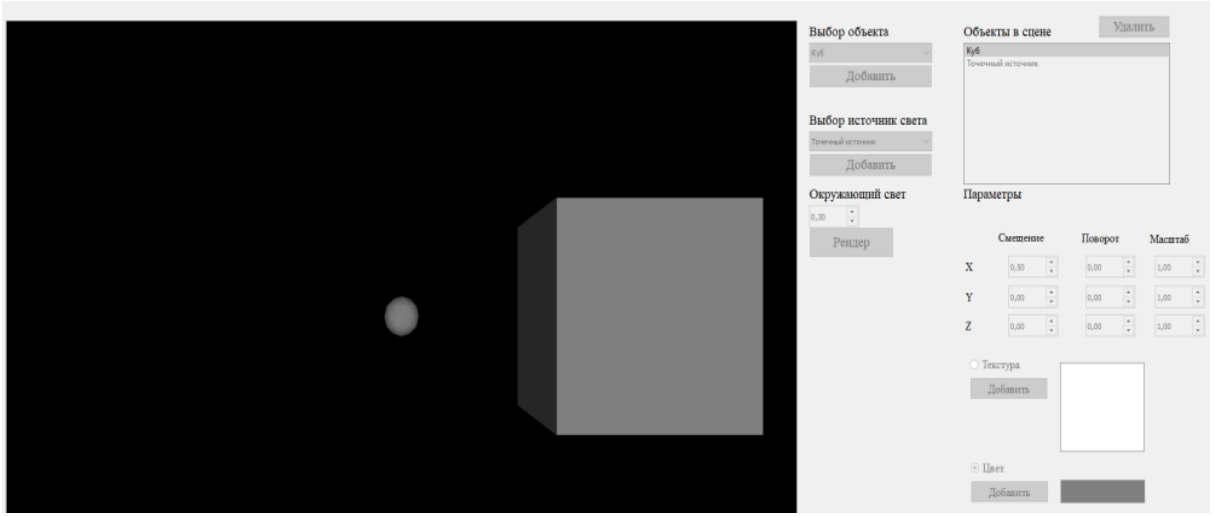


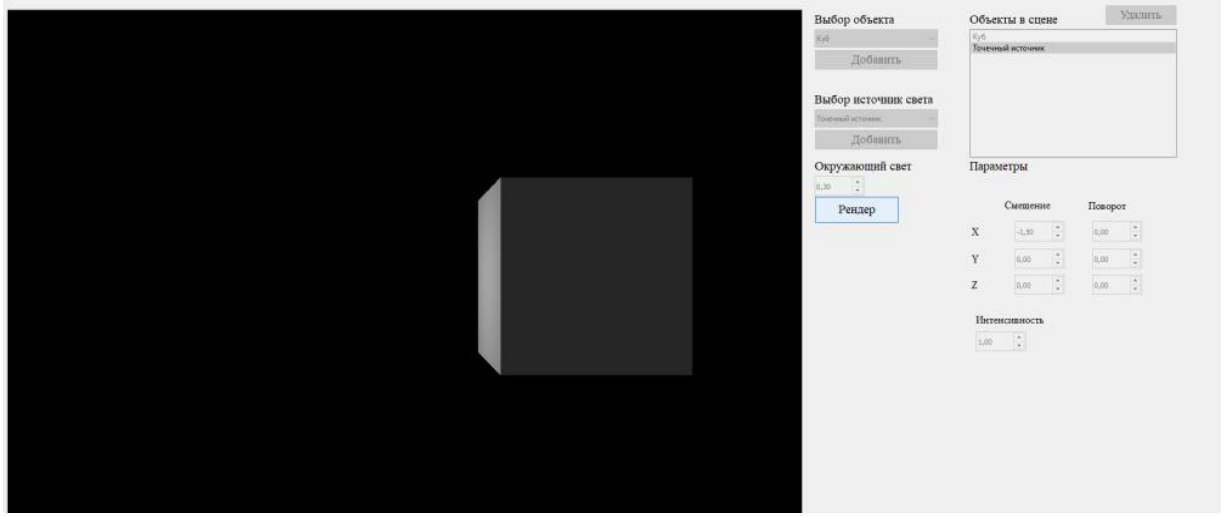
Наложение текстуры и цвета





Результат рендеринга





Построение реалистического изображения