

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)

Факультет программной инженерии и компьютерной техники

Наименование дисциплины: **Алгоритмы Компьютерной Графики**

О Т Ч Е Т

по лабораторной работе 4

Выполнил:

Маргиев Давид, Р3369

Преподаватель:

Андреев Артем Станиславович

Дата: **06.12.2025**

г. Санкт-Петербург
2025

Задание 4:

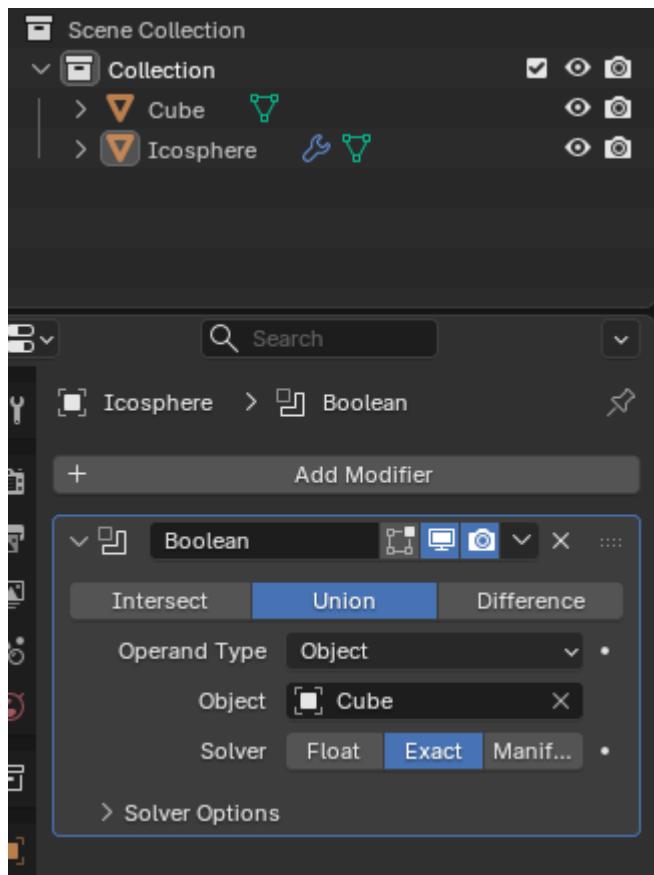
В любом 3d редакторе выполнить модель, используя средства CSG – constructive solid geometry. Получить примеры высоко и низко полигональной моделей. Научиться экспортовать файл. Находить точки на модели и в файле. Открывать этот файл в FX Composer или Render Monkey Задавать материалы (генерировать материалы из прототипа Wizard), ставить свет, найти где управлять FoV камеры.

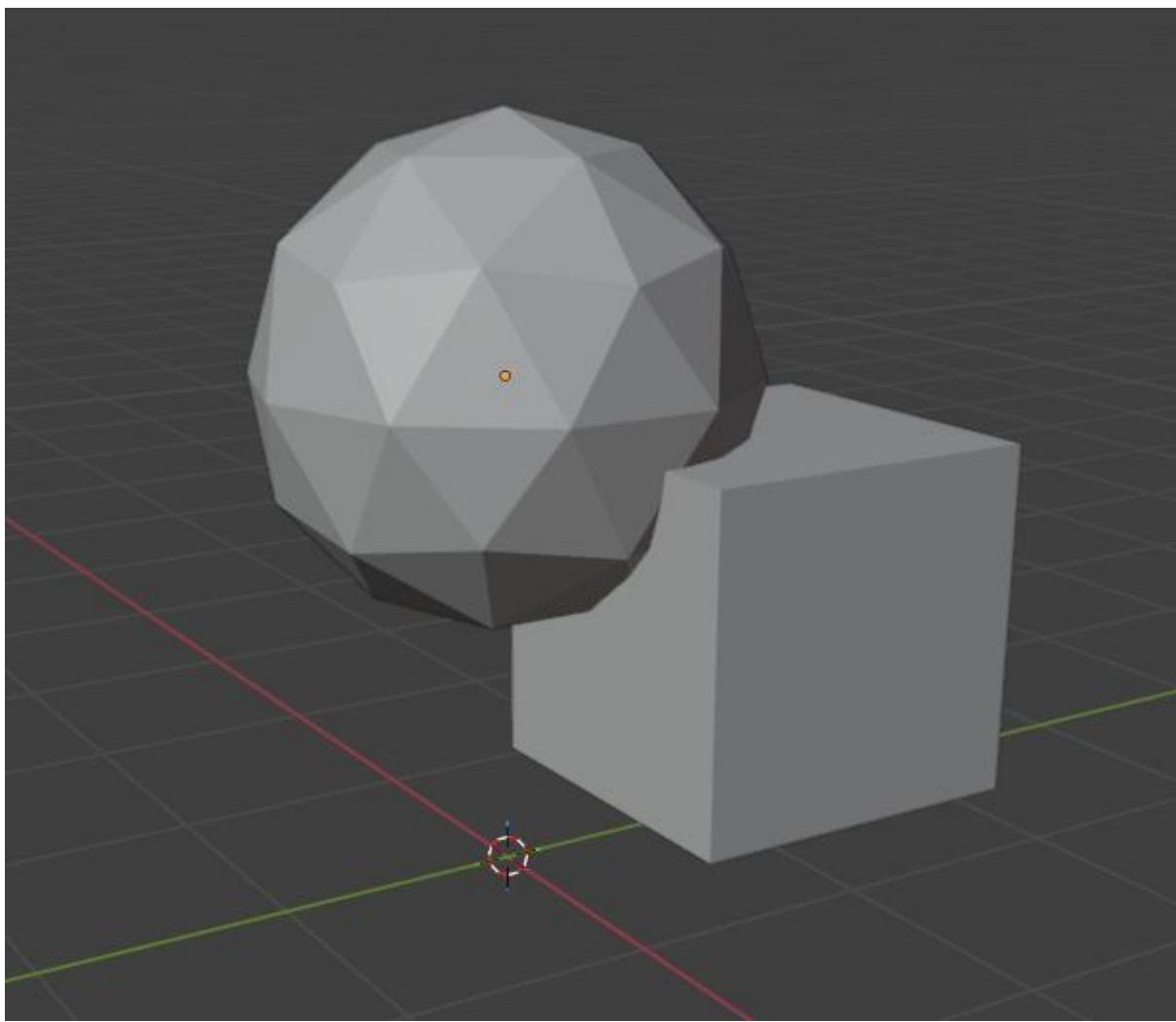
Выполнение:

Для выполнения данного задания я решил использовать blender, он предоставляет все необходимые для этого инструменты.

В blender есть инструмент-модификатор boolean, который применяется к объектам и позволяет использовать режимы union, difference и intersect, что соответствует принципам constructive solid geometry.

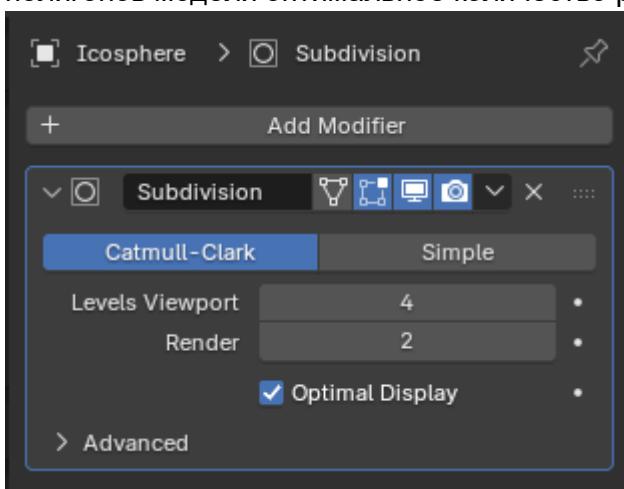
Я сделал простую модель, используя икосферу, куб и модификатор boolean в режиме union, чтобы объединить сетки:



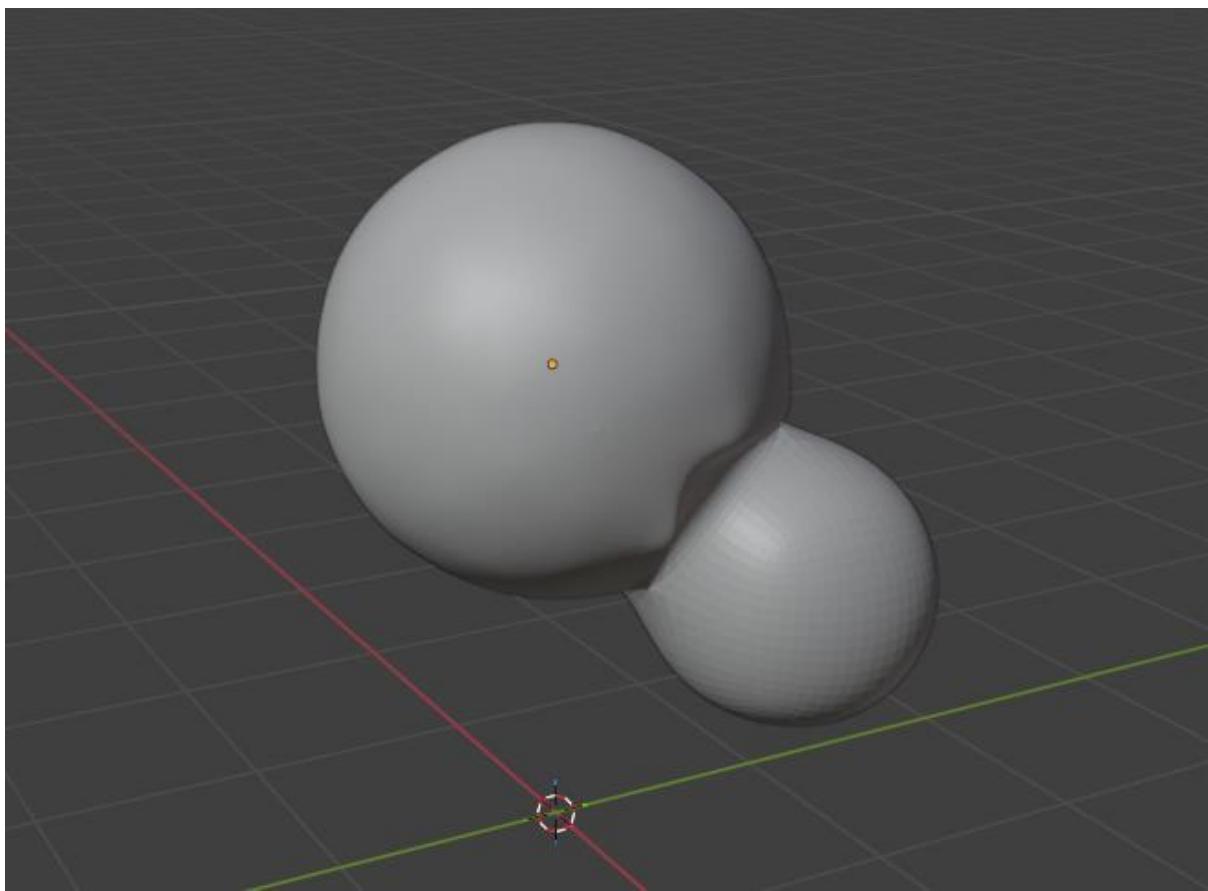


Получилась такая примитивная модель. Можно считать ее low-poly (по современным стандартам), так как blender считает всего 88 полигонов.

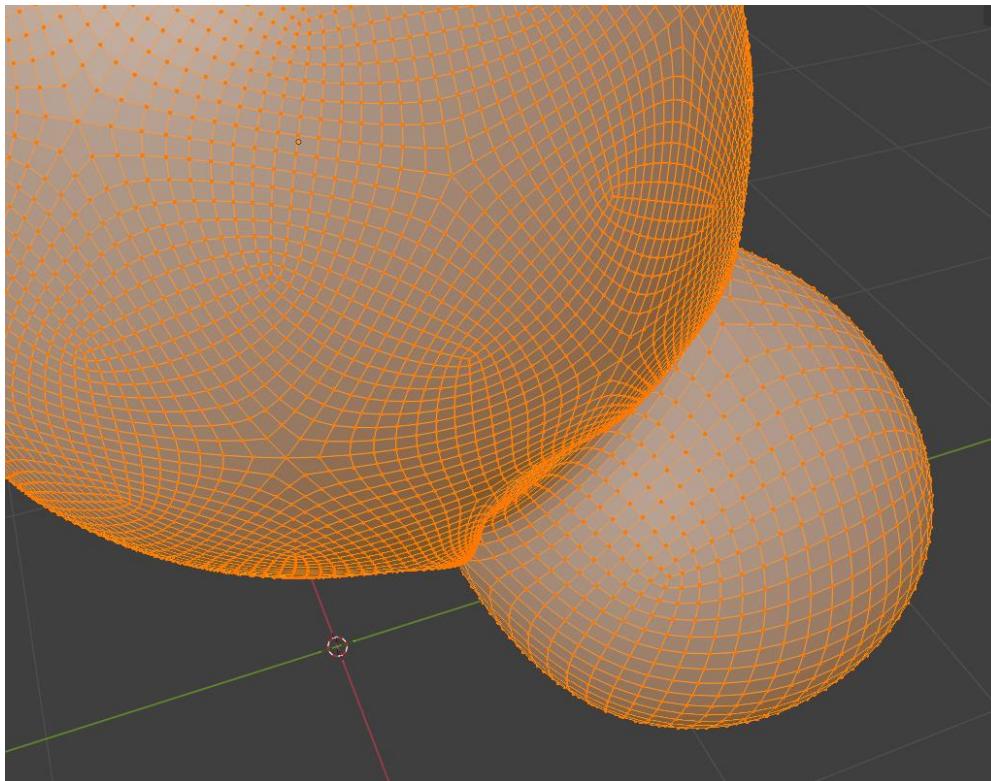
Чтобы сделать из нее high-poly модель, можно использовать предоставляемый блендером модификатор subdivision surface, который оптимально делит каждый из полигонов модели оптимальное количество раз. Применим его на модель:



Это значит, что мы поделили сетку в 4 итерации, вот что вышло:

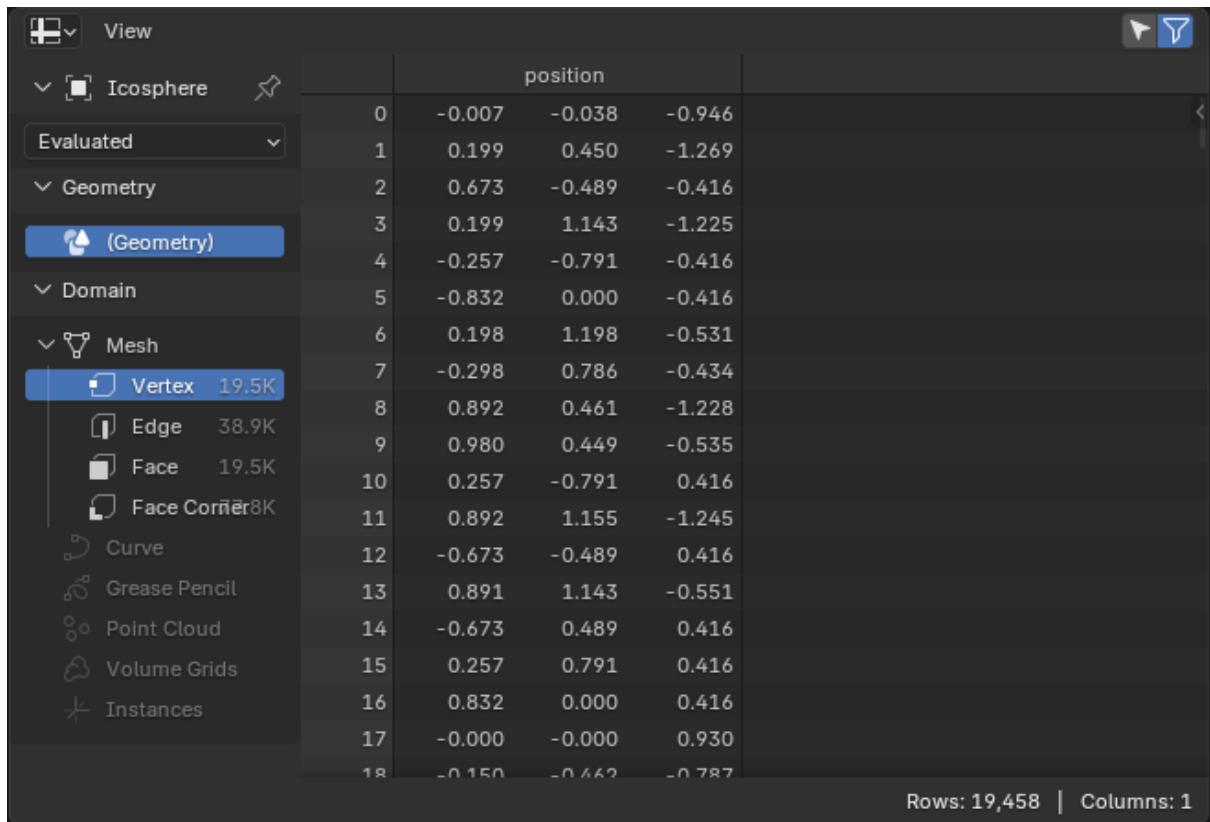


Естественно, в blender мы можем просматривать точки, грани и полигоны модели в отдельном режиме, и манипулировать ими:



Также, blender позволяет просматривать координаты и другую информацию о точках,

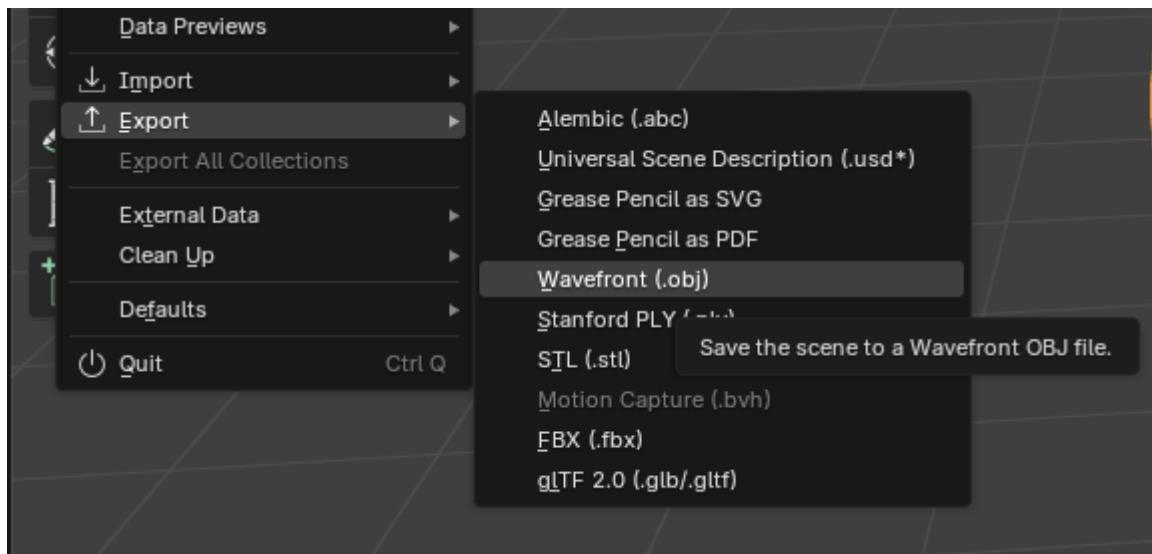
границах и полигонах в удобной таблице:



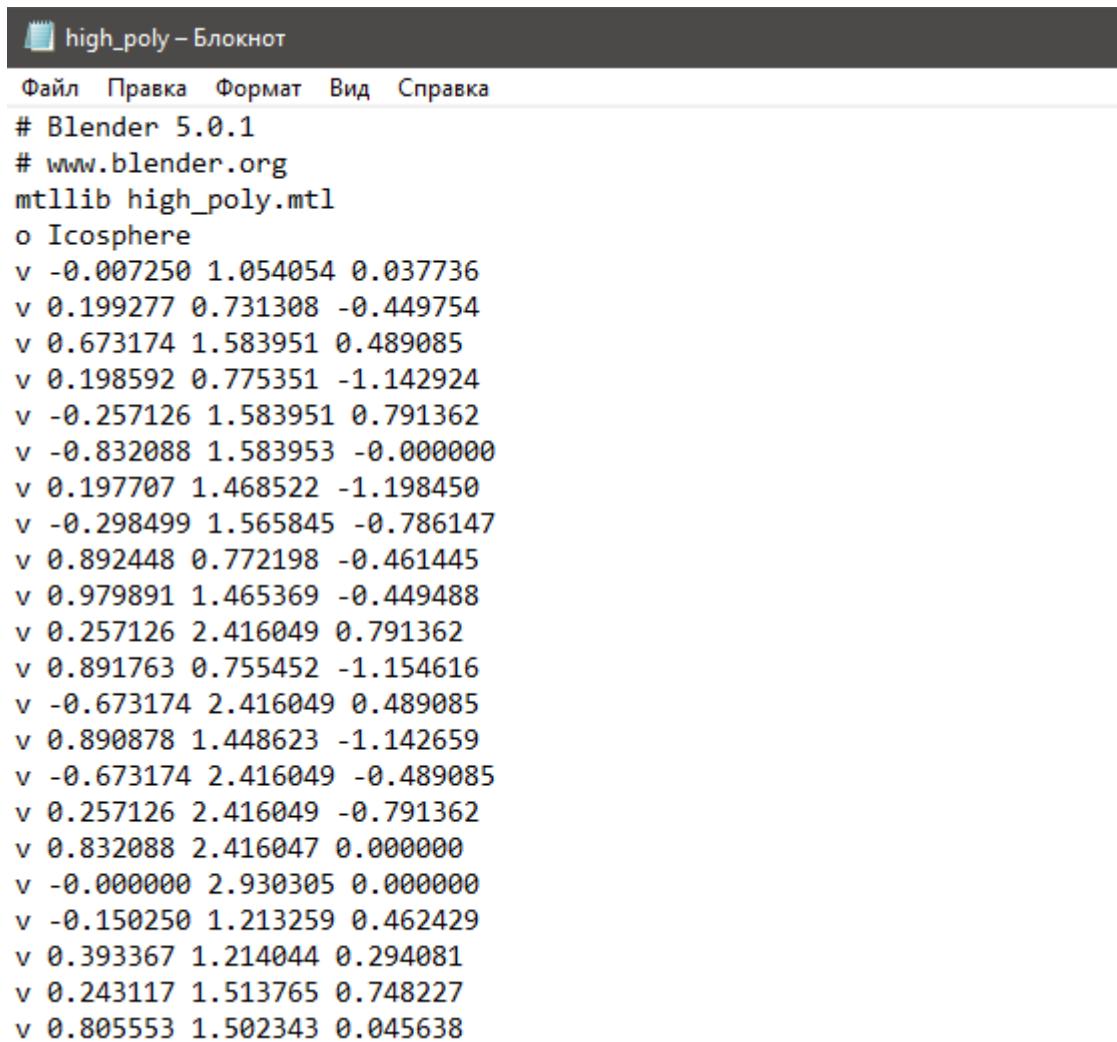
The screenshot shows a 3D modeling software's interface with a table view. The table has a header row 'position' with columns for X, Y, and Z coordinates. Below the header are 19 rows of data, each representing a vertex with its coordinates. The table includes a sidebar with a tree view of the scene structure, including 'Icosphere', 'Evaluated', 'Geometry', 'Domain', 'Mesh', and specific mesh components like 'Vertex 19.5K', 'Edge 38.9K', 'Face 19.5K', etc. At the bottom right of the table, it says 'Rows: 19,458 | Columns: 1'.

position			
0	-0.007	-0.038	-0.946
1	0.199	0.450	-1.269
2	0.673	-0.489	-0.416
3	0.199	1.143	-1.225
4	-0.257	-0.791	-0.416
5	-0.832	0.000	-0.416
6	0.198	1.198	-0.531
7	-0.298	0.786	-0.434
8	0.892	0.461	-1.228
9	0.980	0.449	-0.535
10	0.257	-0.791	0.416
11	0.892	1.155	-1.245
12	-0.673	-0.489	0.416
13	0.891	1.143	-0.551
14	-0.673	0.489	0.416
15	0.257	0.791	0.416
16	0.832	0.000	0.416
17	-0.000	-0.000	0.930
18	-0.150	-0.462	-0.787

Экспортируем обе модели в формате .obj:



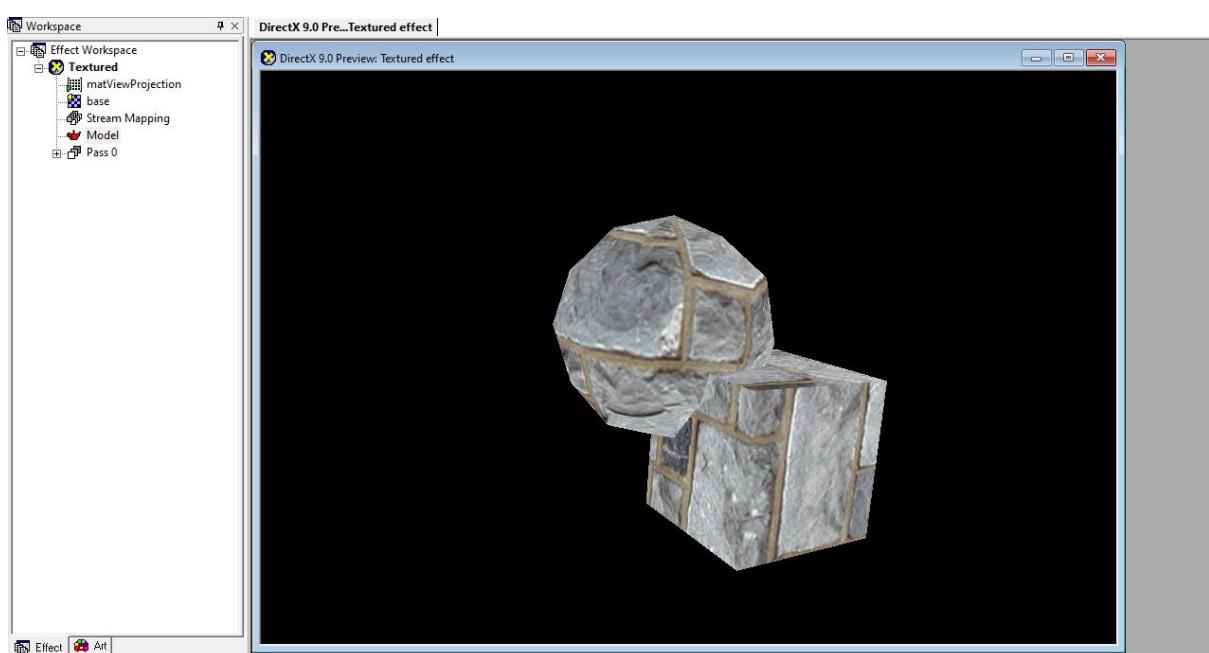
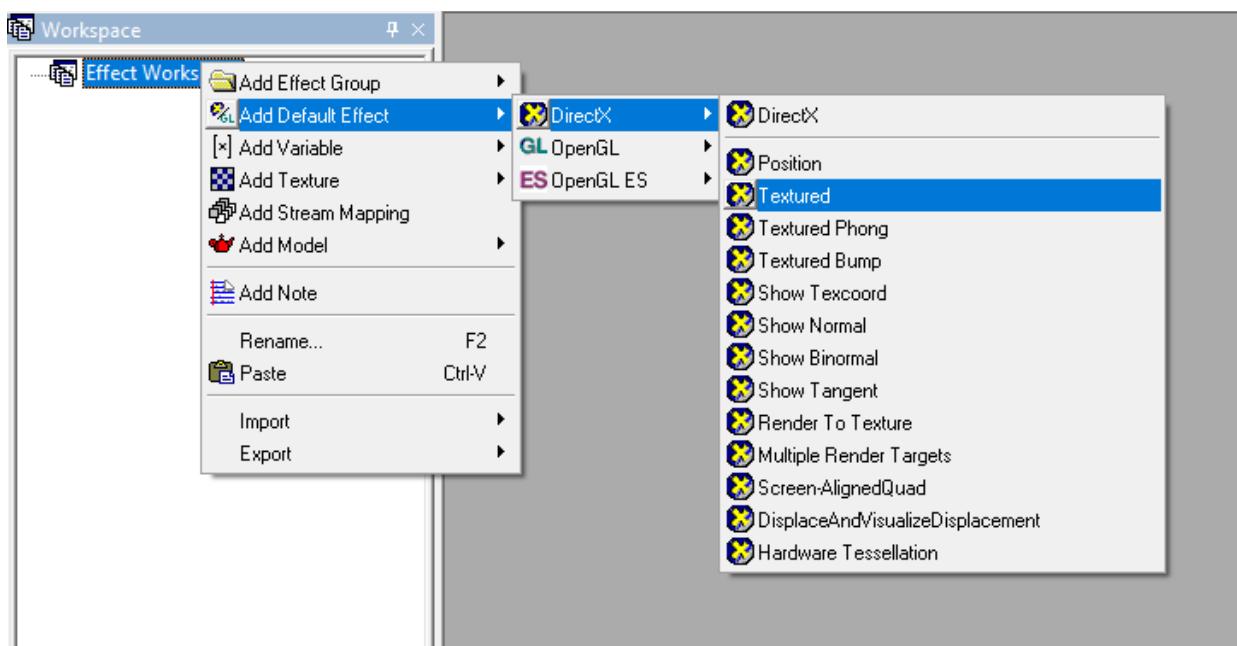
Откроем файл .obj в блокноте, чтобы посмотреть на точки:



```
high_poly – Блокнот
Файл Правка Формат Вид Справка
# Blender 5.0.1
# www.blender.org
mtllib high_poly.mtl
o Icosphere
v -0.007250 1.054054 0.037736
v 0.199277 0.731308 -0.449754
v 0.673174 1.583951 0.489085
v 0.198592 0.775351 -1.142924
v -0.257126 1.583951 0.791362
v -0.832088 1.583953 -0.000000
v 0.197707 1.468522 -1.198450
v -0.298499 1.565845 -0.786147
v 0.892448 0.772198 -0.461445
v 0.979891 1.465369 -0.449488
v 0.257126 2.416049 0.791362
v 0.891763 0.755452 -1.154616
v -0.673174 2.416049 0.489085
v 0.890878 1.448623 -1.142659
v -0.673174 2.416049 -0.489085
v 0.257126 2.416049 -0.791362
v 0.832088 2.416047 0.000000
v -0.000000 2.930305 0.000000
v -0.150250 1.213259 0.462429
v 0.393367 1.214044 0.294081
v 0.243117 1.513765 0.748227
v 0.805553 1.502343 0.045638
```

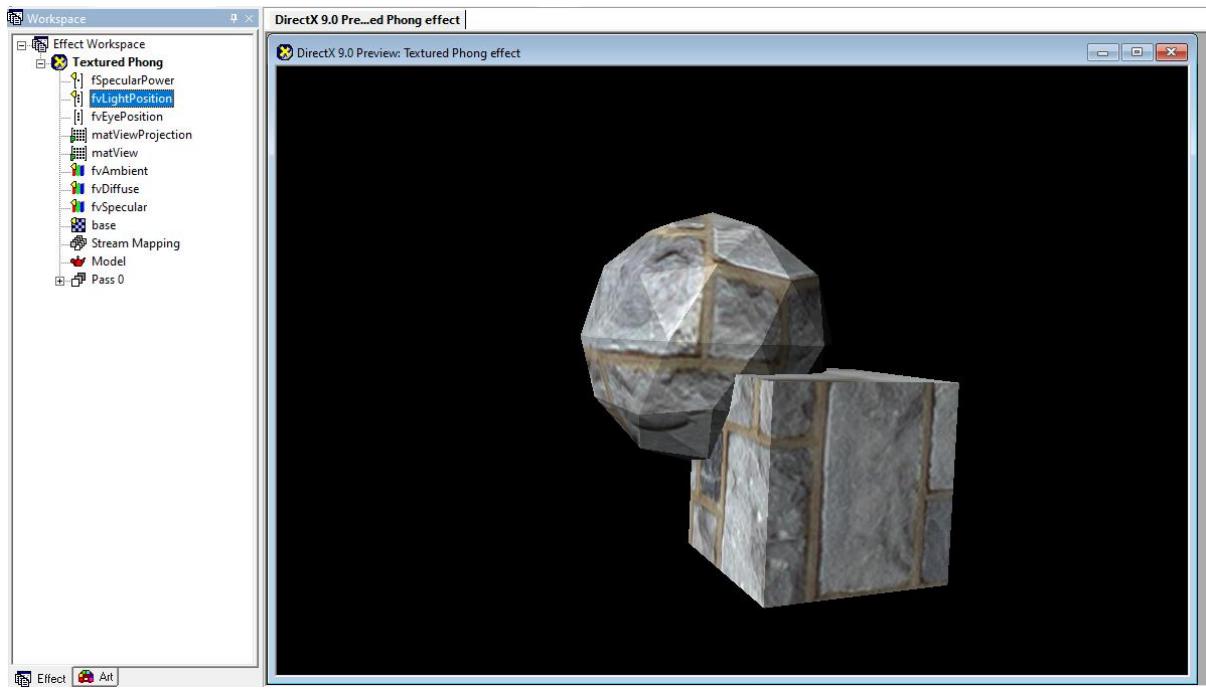
Каждая v-строчка это отдельная вершина с ее XYZ-координатами (скорее всего относительно мира)

Теперь откроем модель в Render Monkey, для этого создадим в рабочем пространстве один из дефолтных эффектов DirectX и поменяем модель на нашу:

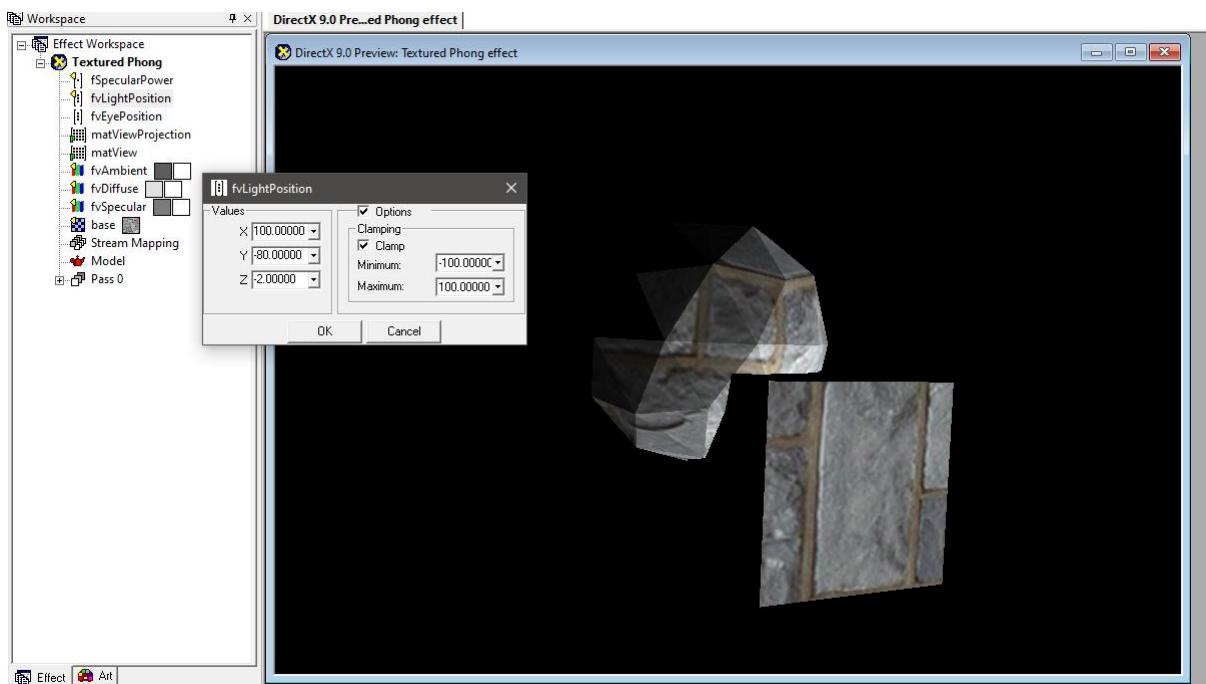


Как видим, модель корректно отображается, а также на нее корректно проецируются текстуры.

Теперь попробуем поставить свет, выбрав уже другой эффект:

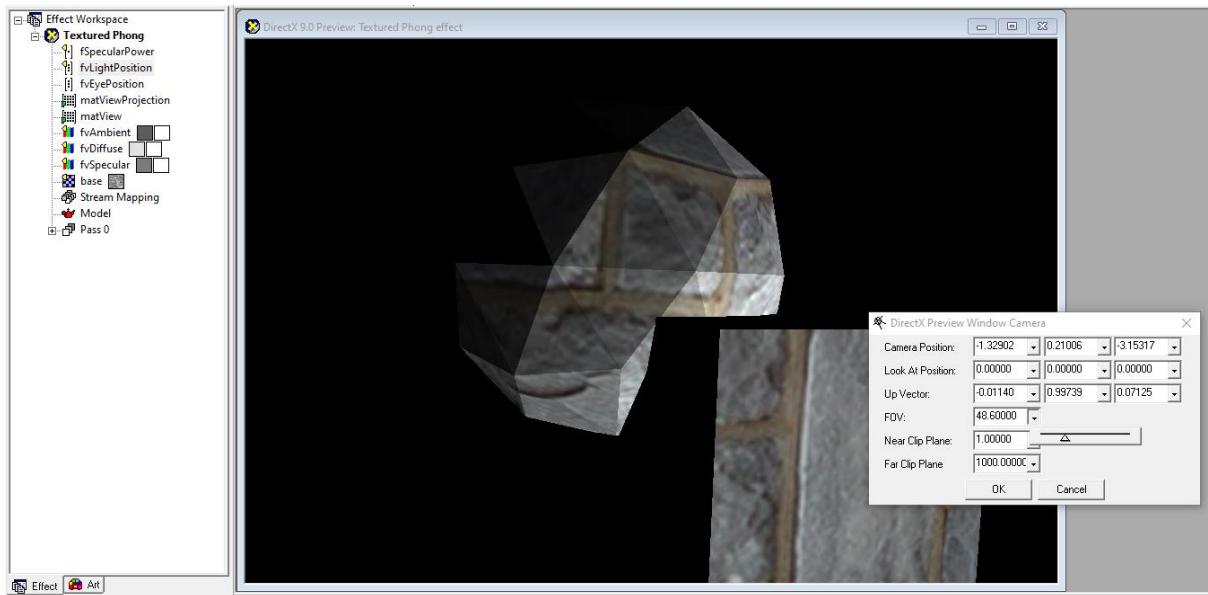


В рабочем пространстве мы видим пункт fvLightPosition, который можем изменить.



Таким образом, я настроил позицию освещения в пространстве.

Теперь изменим FoV камеры, для этого откроем контекстное меню рендер-окна, и исследуем настройки камеры, где и будет параметр FoV, который можно изменить:



Таким образом, я создал примитивную модель в blender, используя принципы CSG, экспортировал ее в двух (low-poly и high-poly) исполнениях, открыл ее в Render Monkey, при этом текстурировав ее, поставив в сцене свет и FoV камеры.