

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Санкт-Петербургский национальный исследовательский  
университет информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ  
ТЕХНОЛОГИЙ

## Задание

по дисциплине

«Алгоритмы компьютерной графики»

***Выполнил студент:***

Баянов Равиль Динарович Р3334

***Преподаватель:***

Андреев А.С.

Санкт-Петербург

2024 г.

## Оглавление

Задание 1 .....	3
Решение 1 .....	4
Задание 2 .....	6
Решение 2 .....	7
Задание 3 .....	9
Решение 3 .....	10
Задание 4 .....	11
Решение 4 .....	12
Задание 5 .....	15
Решение 5 .....	16
Задание 7 .....	17
Решение 7 .....	18
Задание 8 .....	19
Решение 8 .....	20
Вывод .....	21

## Задание 1

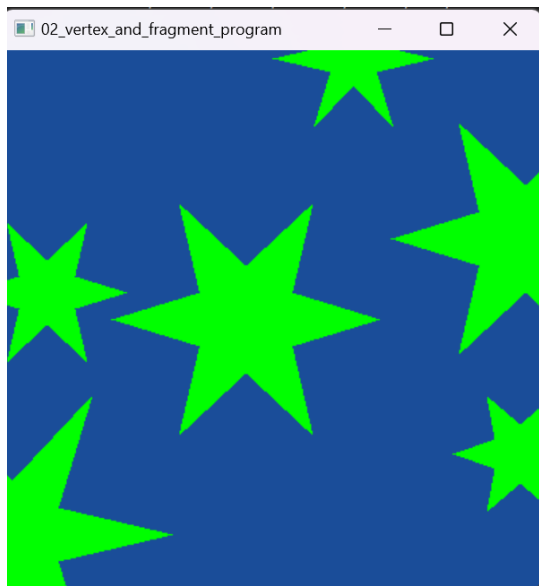
Взяв любые примеры – OpenGL GLSL, HLSL пример отрисовки, преобразовать вывод звёздочки с 5 лучами на иное число лучей – 4, 6.

Поменять цвет фона, заданный в исполняемом файле.

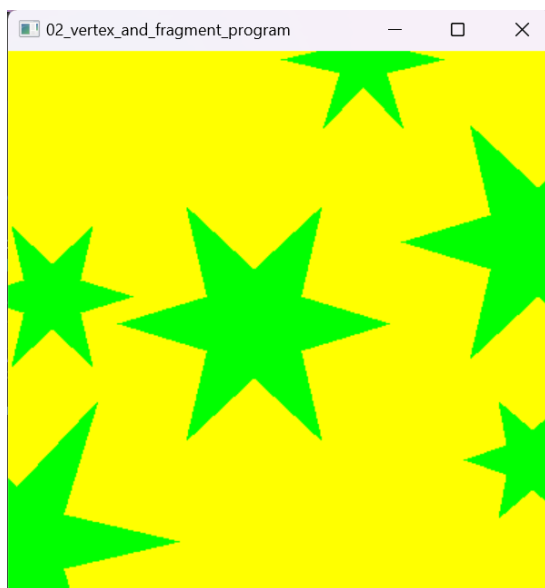
Реализовать вместо triangle fan: triangle list и triangle strip.

# Решение 1

Вывод звёздочек с числом лучей - 6:



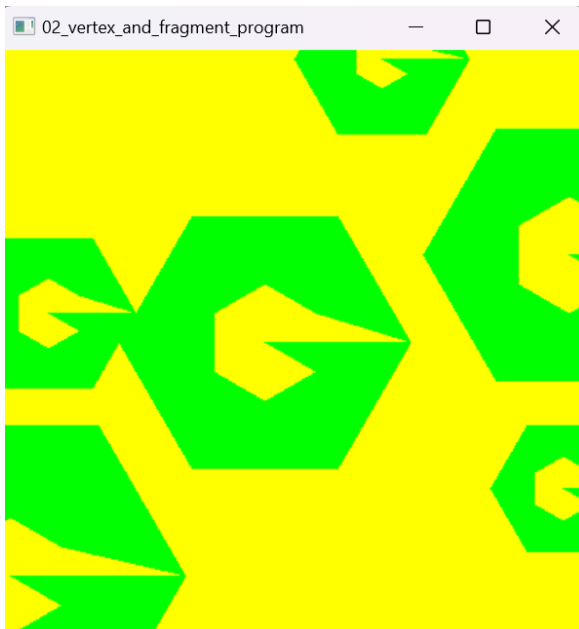
Вывод звёздочек с другим фоном:



Triangle list:



triangle strip:



## Задание 2

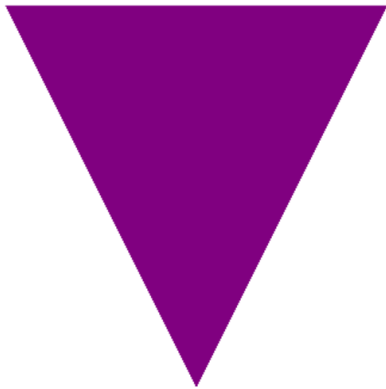
1. Учимся выводить текстуру треугольника.

Показать в отчете passthrough координаты в нормализованной системе координат, помним, что деление на  $w$  осуществляет ускоритель сам. И показать, реализовав  $0.5 * x_u + \text{vec2}(0.5)$  как они пробегают все значения на экране

2. Выводим UV координаты
3. Выводим текстуру на экран, обязательно свою, а не всяких «чертей» из директории, сопровождающих NV примеры.

## Решение 2

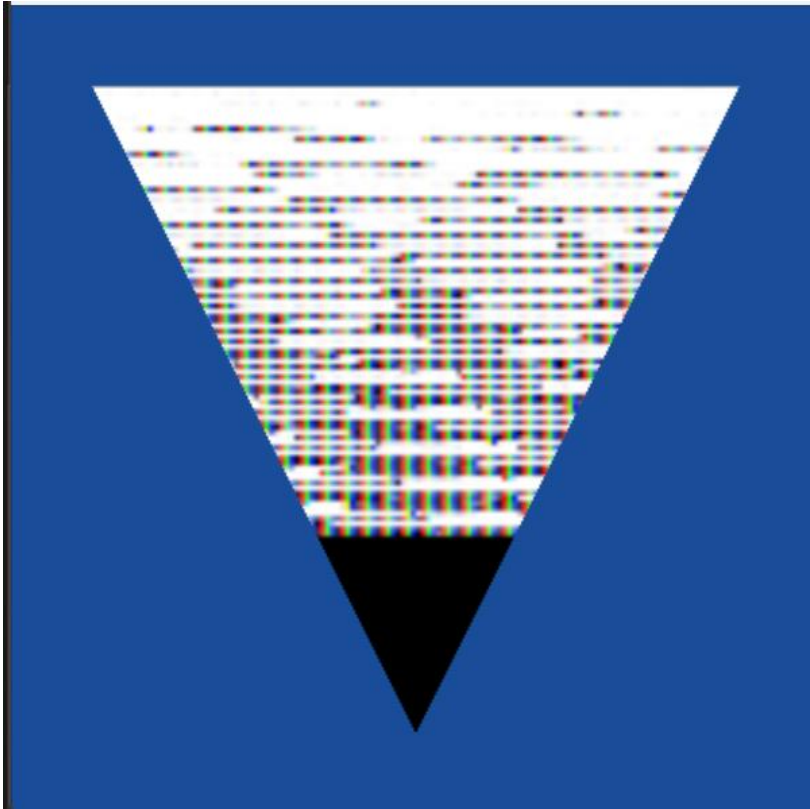
Вывод текстуры:



Фрагментный шейдер для вывода UV координат:

```
struct C3E2v_Output {  
  
    float4 position : POSITION;  
  
    float3 color : COLOR;  
  
    float2 texCoord : TEXCOORD0;  
  
    float4 pos2: TEXCOORD1;  
  
};  
  
struct C2E2f_Output {  
  
    float4 color : COLOR;  
  
};  
  
C2E2f_Output C2E2f_uv_debug(C3E2v_Output input)  
{  
  
    C2E2f_Output OUT;  
  
    // Output UV coordinates as color  
    OUT.color = float4(input.texCoord, 0.0, 1.0);  
  
    return OUT;  
}
```

Также можно вывести текстуру вот такой незамысловатой картинки:





## Задание 3

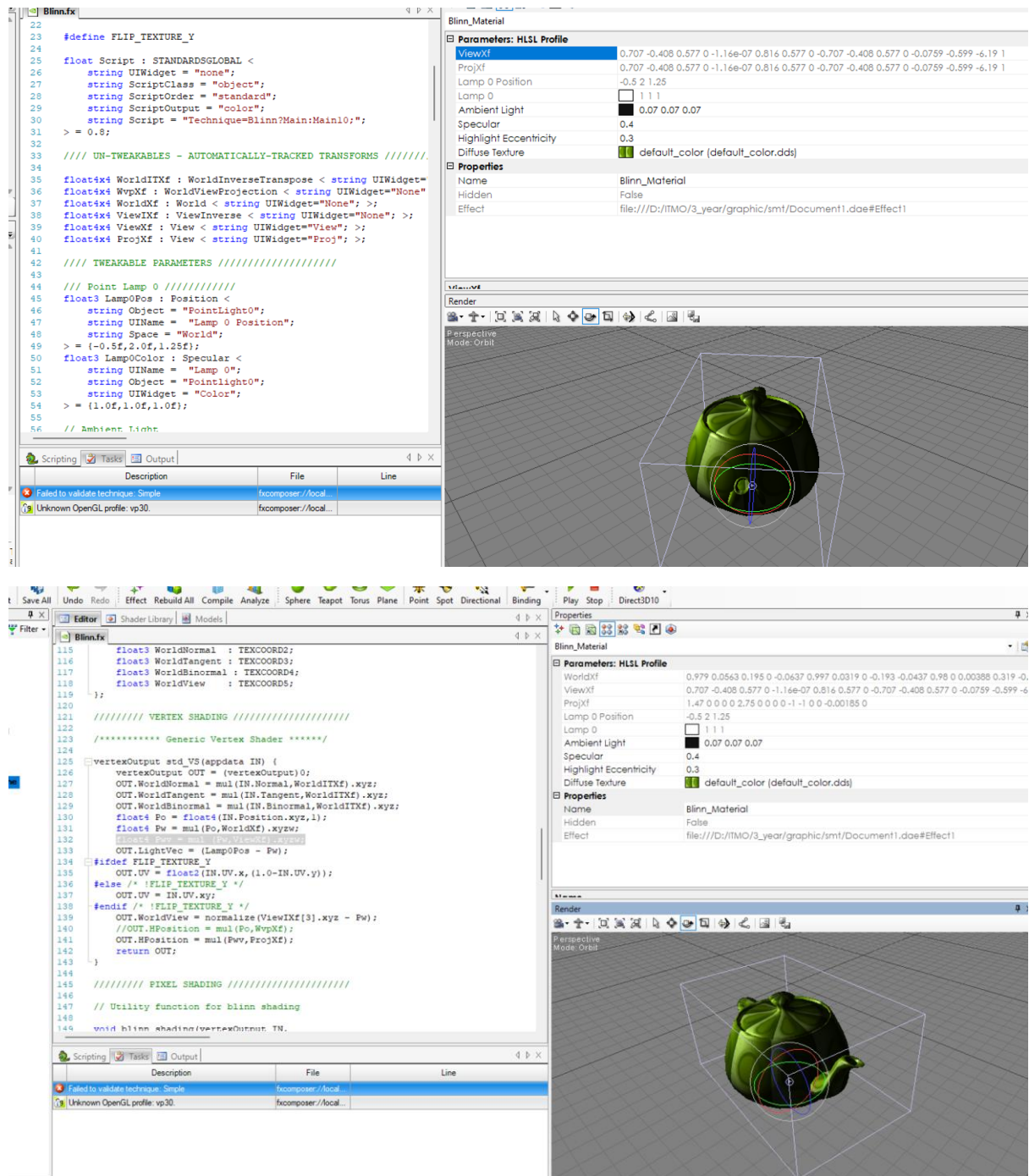
Можно в FX Composer

Проверить, как задаются и передаются последовательность матриц.

Помножить точки на матрицы в вершинном шейдере получая последовательные промежуточные результаты, затем одной WorldViewProj матрицей

## Решение 3

В вершинном шейдере определим матрицы, которые мы будем использовать



Ну и с помощью команды mul помножим точки на матрицы.

## Задание 4

В любом 3d редакторе выполнить модель, используя средства CSG – constructive solid geometry.

Получить примеры высоко и низкополигональной моделей.

Научиться экспортировать файл.

Находить точки на модели и в файле.

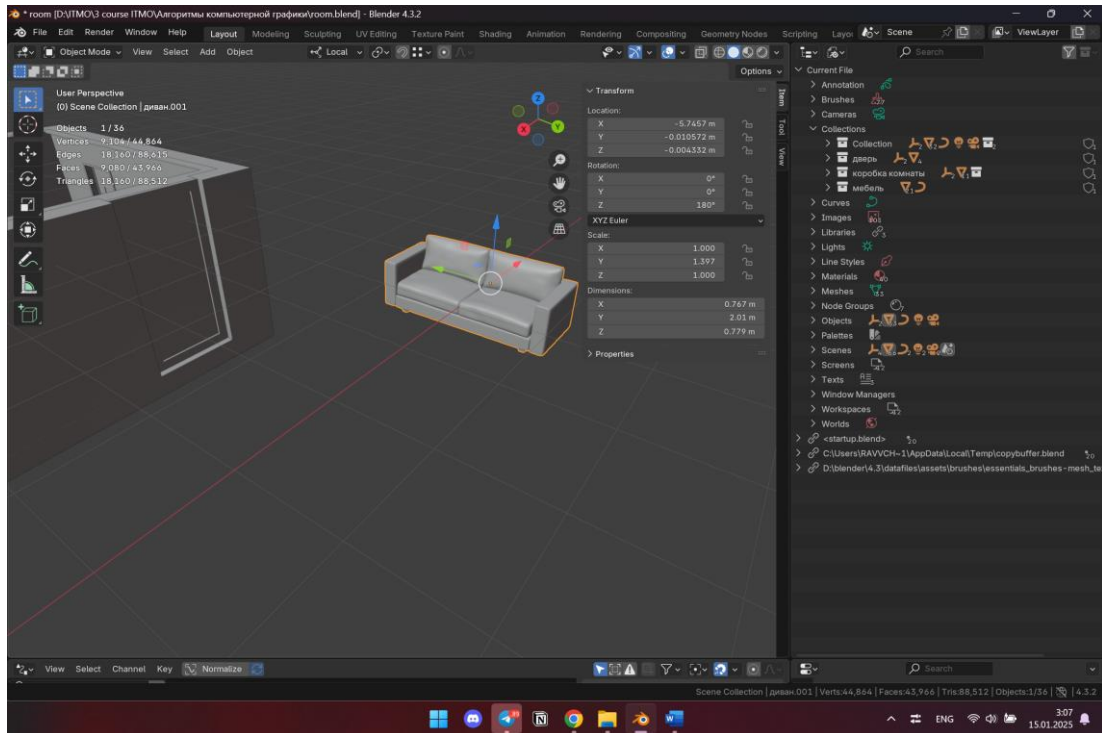
Открывать этот файл в FX Composer или Render Monkey

Задавать материалы (генерировать материалы из прототипа Wizard), ставить свет, найти где управлять FoV камеры.

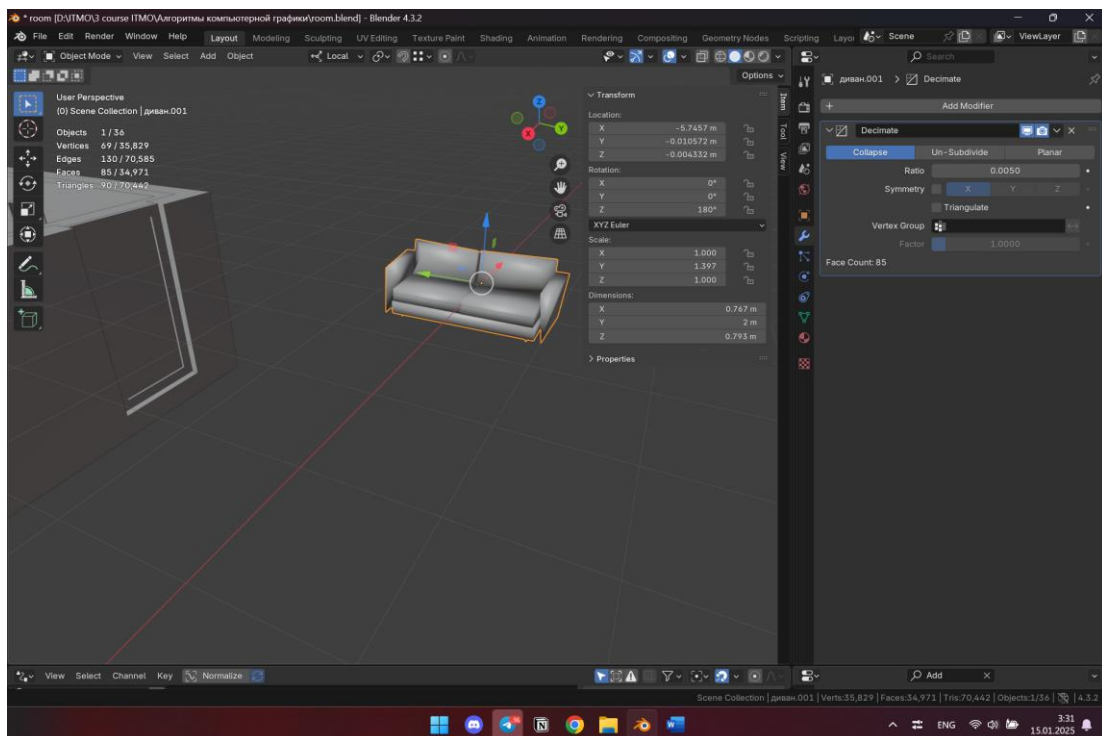
## Решение 4

Возьмём диван из моего старого проекта, который сделан с помощью средств CSG

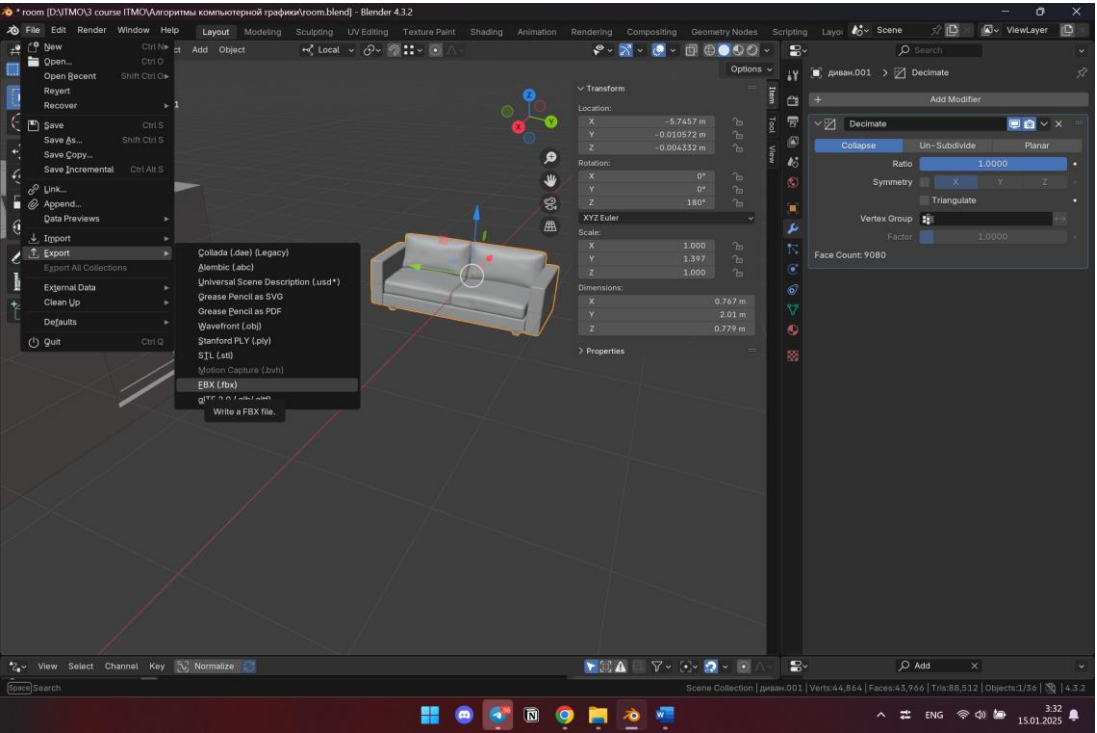
Посмотрим на кол-во полигонов на данном диване до уменьшения полигонов. То есть посмотрим на высокополигональную модель:



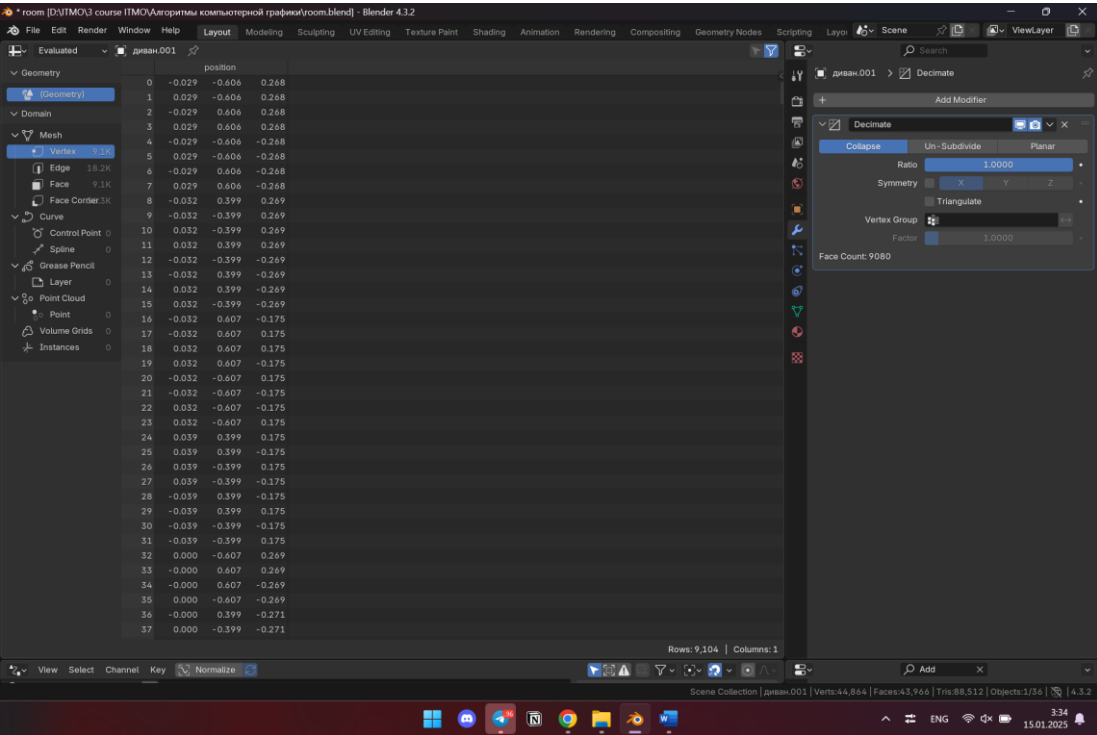
Теперь воспользуемся модификатором decimate и уменьшим кол-во полигонов:



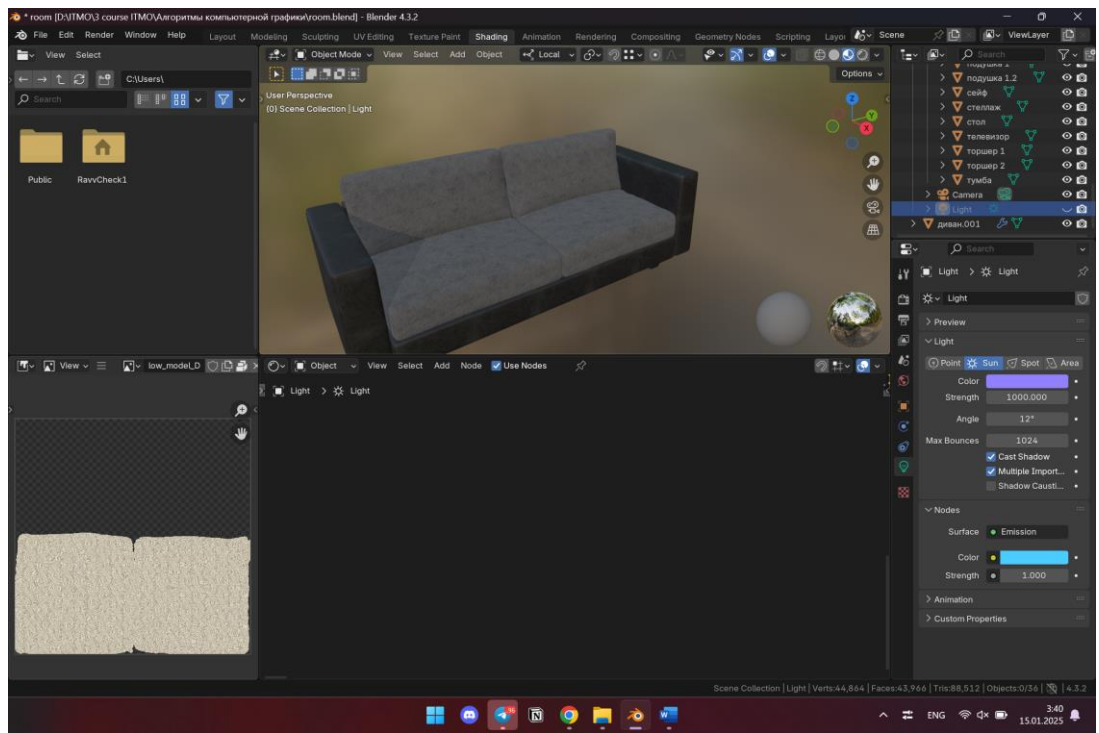
Вот таким способом можно экспортировать нашу модель в любой нам удобный формат:



На этом скриншоте мы видим как мы можем посмотреть координаты каждого из полигонов нашего объекта:



Здесь мы видим генерацию света и так же управление FoV камерой:



## Задание 5

Поискать какие инструменты в инструментах 3d визуализации могут быть полезны для упрощения полигональных сеток. Mesh Reduce в разных пакетах.

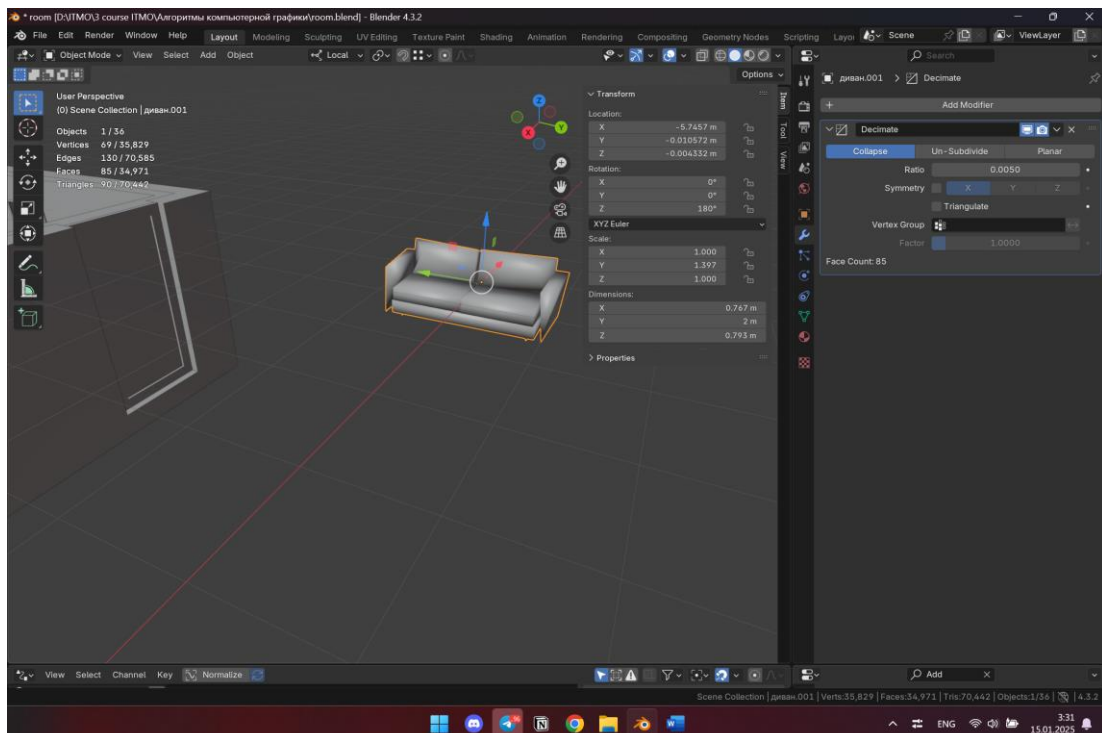
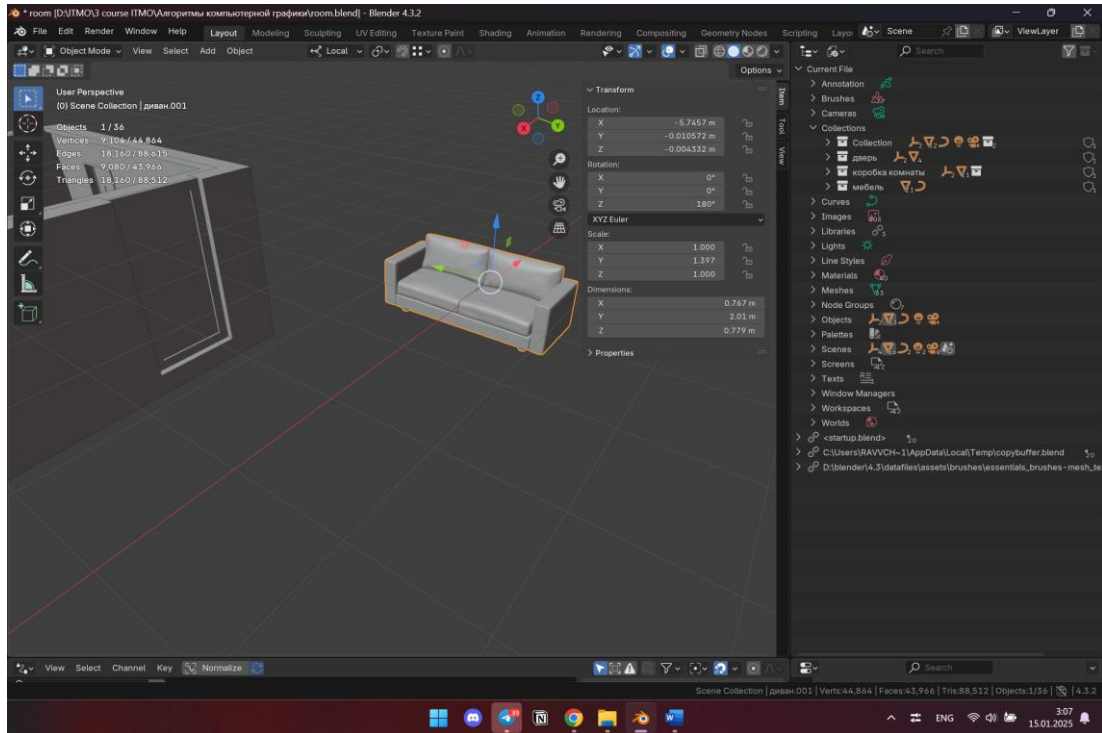
Встроенные средства DirectX 9 сейчас считаются уже устаревшими, GPU обладают достаточной мощностью.

Посмотреть как сокращение числа полигонов влияет на внешний вид моделей.

Использовать ваши 3d модели.

## Решение 5

Базовый инструмент decimate в blender я показывал в прошлом задании так что просто продублирую скриншоты изменения полигонов:



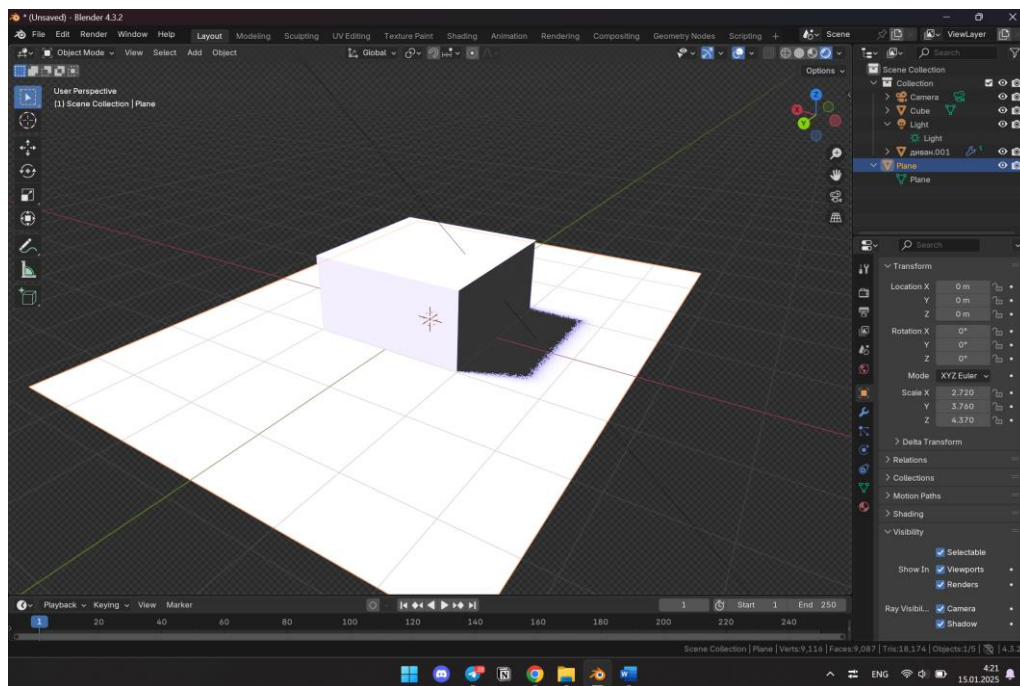


## **Задание 7**

Методы генерации теней с помощью depth buffer.

# Решение 7

А вот и добавление теней на куб с фиолетовым мерцанием):

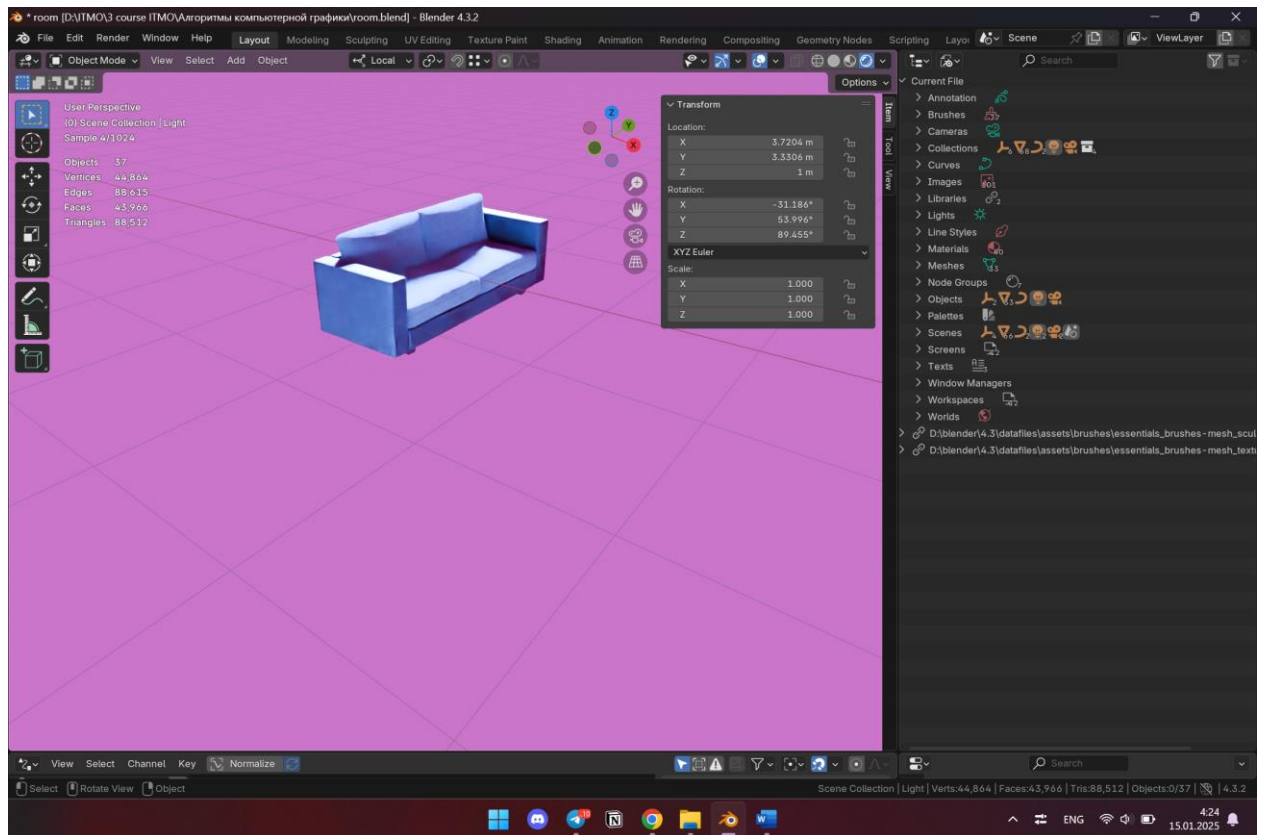


## **Задание 8**

Объемный метод генерации теней, подать свои модели, посмотреть

## Решение 8

А вот так примерно выглядит объёмный метод генерации теней:



## **Вывод**

Выполнив данные задания по алгоритмам компьютерной графики, я вспомнил что такое blender и в целом узнал много нового о создании моделей и изображений.