МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по курсовой работе

по дисциплине «Компьютерная графика»

Тема: Визуализация 3D объекта зонт

Студент гр.8382 _____ Нечепуренко Н.А.

Студент гр.8382 _____ Терехов А.Е.

Преподаватель _____ Герасимова Т.В.

Санкт-Петербург

2021

Цели работы.

С помощью средств библиотеки OpenGL реализовать 3D сцену с зонтом.

Задание.

Задачи:

1. Подбор материала по теме для обзора (1-2 стр), материал д.б. творчески переработан, дополнен примерами вашей реализации. Обязательны ссылки на литературу.
2. Создать описание генерации вашей модели (не создавать в средствах типа Blender, 3D MAX).
3. Разработка демонстрационного примера
4. В отчете обязателен список использованной литературы

   Сцена управляема – можно облететь вокруг.

Рисунок 1 – Пример зонта для задания

Выполнение работы.

Зонт состоит из двух отличающихся по построению частей – палочка с ручкой и непосредственно полотно. Для генерации палочки и ручки были использованы цилиндры разных диаметров (см. рис. 2).

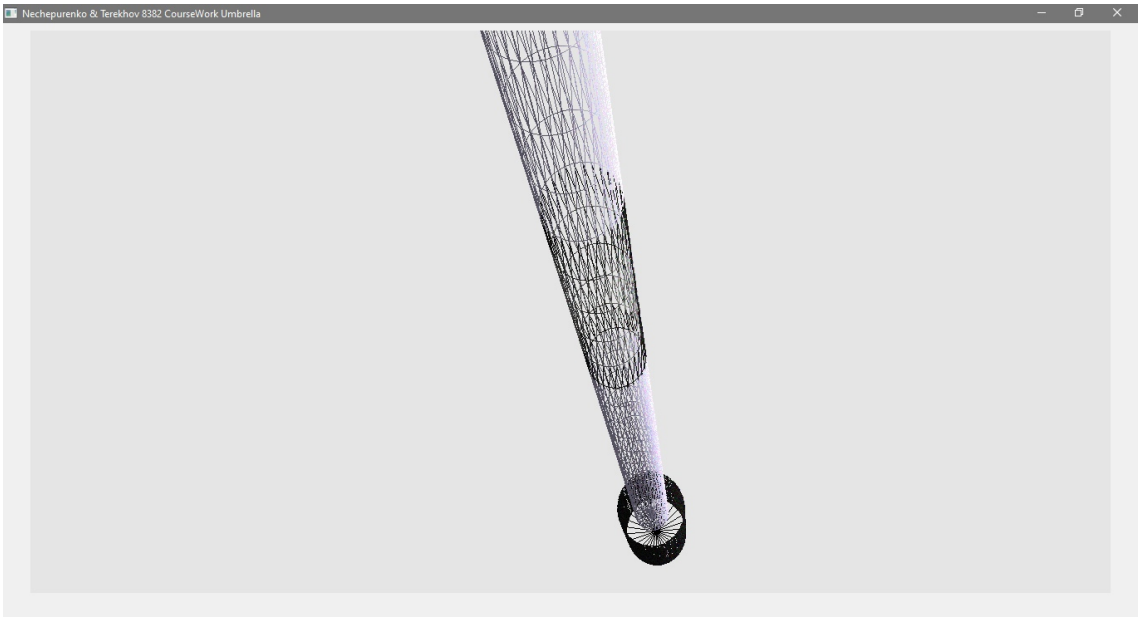

Рисунок 2 – Каркас палочки с ручкой из цилиндров

Генерация цилиндров тривиальна, не будем приводить ее в отчете. Алгоритм подробно описан в статье Song Ho Ahn (ссылка в разделе использованные источники).

Для генерации полотна зонтика пришлось лучше изучить топологию зонта из задания. Глобально полотно представляет собой полусферу, но так как у зонтика из задания есть вогнутые части, то пришлось варьировать радиус полусферы в зависимости от сектора, для получения такого эффекта.

На рисунке ниже приведен зонт из задания и полученная каркасная модель.

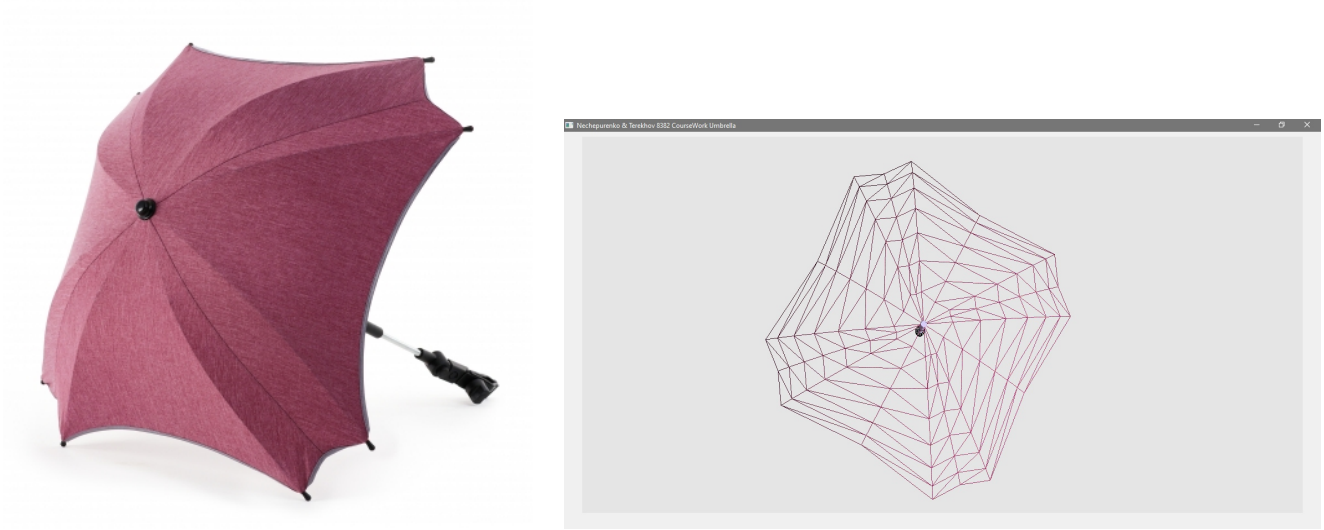

Рисунок 3 – Сравнение заданного зонта и полученного каркаса

Для построения такой сетки необходимо изменить стандартный алгоритм генерации полусферы и добавить условие на радиус и инкремент угла сектора.

```
switch (j % 4) {
    case 0:
        sectorAngle += sectorStep / 2;
        xy_cur = xy * 0.95f;
        break;
    case 1:
        sectorAngle += sectorStep / 2;
        xy_cur = xy;
        break;
    case 2:
        xy_cur = xy * 0.78f;
        sectorAngle += 3 * sectorStep / 2;
        break;
    case 3:
        xy_cur = xy;
        sectorAngle += 3 * sectorStep / 2;
```

```
            break;
        }
```

где `sectorStep` равен $\frac{2\pi}{sectorCount}$.

Сбоку каркас сгенерированного полотна выглядит следующим образом:

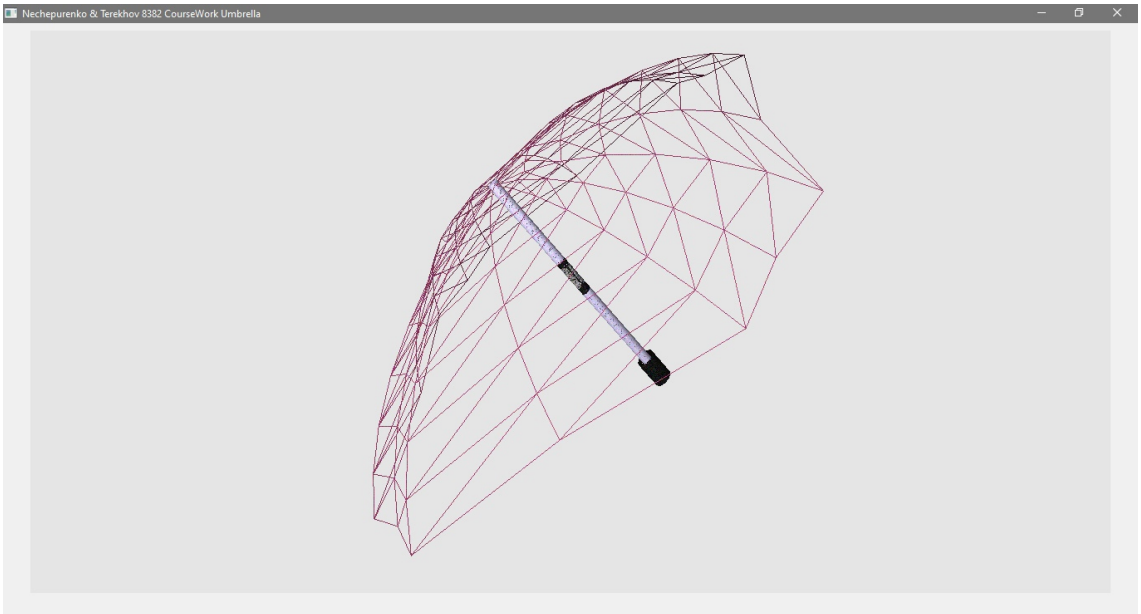

Рисунок 4 – Каркас полотна сбоку

После наложения текстур и добавления шума на поверхность полотна для лучшего соответствия с зонтом из задания был получен следующий результат.

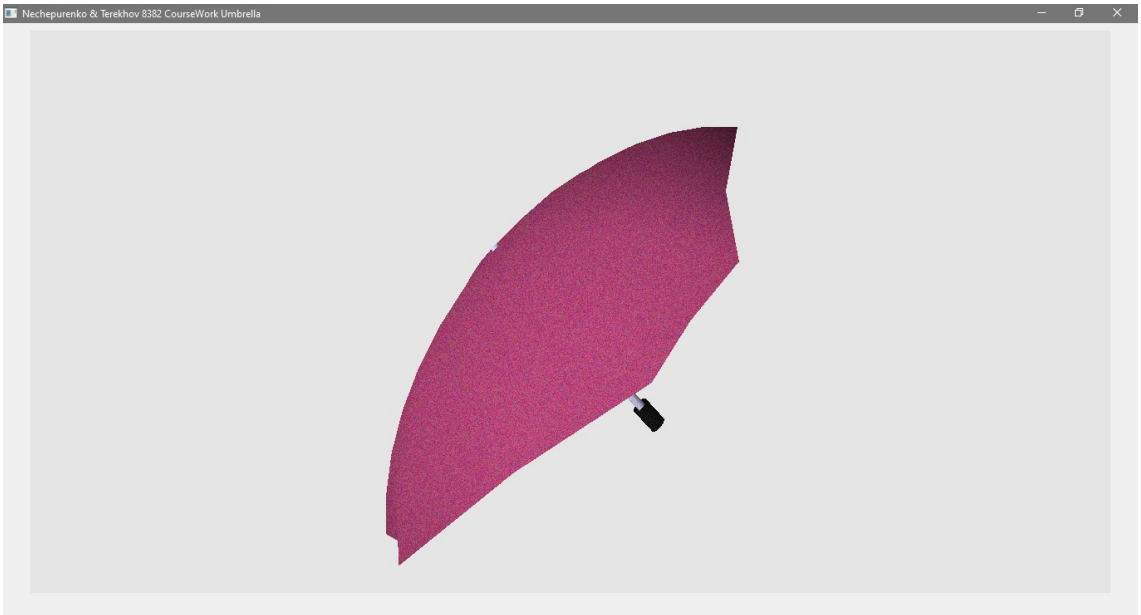

Рисунок 5 – Полученный зонт

Шум был добавлен в связи с тем, что нет исходной текстуры, которая представлена на зонте из задания. Алгоритм генерации был взят статьи The Book of Shaders: Noise (см. раздел использованных источников). Шум генерируется во фрагментном шейдере и добавляется к цвету текстуры

```
float rand2D(in vec2 co){
    return fract(sin(dot(co.xy ,vec2(12.9898,78.233))) *
       43758.5453);
}
...
vec4 color_t=texture2D(texture,v_texCoord);
float noise_1 = rand2D(vec2(v_texCoord.x, v_texCoord.y));
float noise_2 = rand2D(vec2(v_texCoord.x * 336, v_texCoord.y)
   );
float noise_3 = rand2D(vec2(v_texCoord.x, v_texCoord.y * 336)
   );
color_t = color_t + vec4(noise_1*0.2, noise_2*0.2, noise_3
   *0.2, 0);
gl_FragColor=color_t*vec4((ambient+diffuse),1.0f);
```

Вид сверху и снизу представлены на рисунке 6.

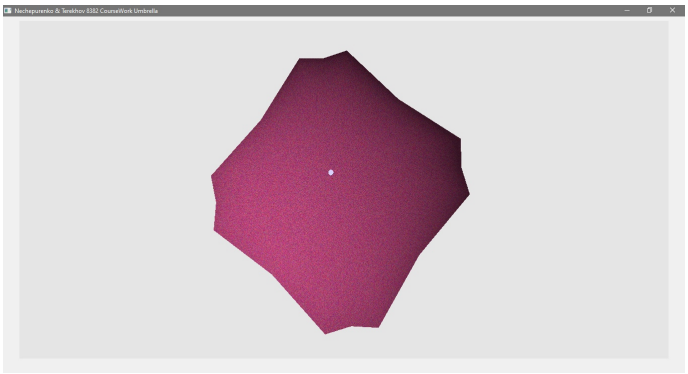

Рисунок 6 – Вид сверху и снизу

Освещение и перемещение камеры были реализованы с помощью шейдеров аналогично ЛР №6 и №7.

Выводы.

В результате выполнения курсовой работы была реализована 3D сцена с возможностью перемещать камеру и осматривать объекты на ней. Был сгенерирован зонтик согласно требуемому виду (см. рис. 1). На сцене существует освещение, реализованное средствами шейдерного языка GLSL.

Список использованных источников.

1. Learn OpenGL – https://learnopengl.com/
2. Уроки по OpenGL | Ravesli – https://ravesli.com/uroki-po-opengl/
3. OpenGL Cylinder, Prism and Pipe – http://www.songho.ca/opengl/gl_cylinder.html
4. The Book of Shaders: Noise – https://thebookofshaders.com/11/