МИНОБРНАУКИ РОССИИ

Санкт-Петербургский государственный

электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина)

Кафедра МОЭВМ

отчет

по лабораторной работе №1

по дисциплине «Компьютерная 3D графика»

Тема: Рисование геометрических объектов

|  |  |  |
| --- | --- | --- |
| Студенты гр. 6303 |  | Иванов В.С. |
| Преподаватель |  | Герасимова Т.В. |

Санкт-Петербург

2020

**Цель работы**

Целью работы является ознакомление с основными примитивами WebGL. Требования и рекомендации к выполнению задания:

* проанализировать полученное задание, выделить информационные объекты и действия;
* разработать программу с использованием требуемых примитивов и атрибутов.

**Задание**

Получите удобное рисование с буферами вершин, униформой и шейдерами:

* рисование нескольких вещей с отдельными командами рисования;
* используя разные примитивы;
* изменение размеров линий и точек по умолчанию;
* изменение цвета на лету.

Вариант 35.

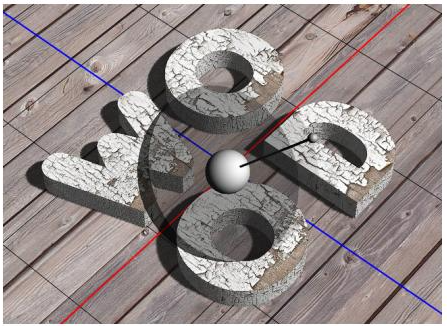


Рисунок 1 – вариант 35

**Ход работы**

1. **Технологии.**

Работа выполнена в JavsScript-фреймворке Vue.js. Использована библиотека [VueGL](https://vue-gl.github.io/) (библиотека для рендеринга 3D-графики с помощью Vue.js и three.js).

1. **Основное.**

Для создания canvas с контекстом WebGL используется компонент VglRenderer.

Для создания сцены использован компонент VglScene.

Для камеры использован компонент VglPerspectiveCamera.

Для рисования геометрии с помощью настраиваемых вершинных и фрагментных шейдеров использован компонент VglShaderMaterial.

1. **Шейдеры.**
   1. **Вершинный шейдер.**

uniform float offsetX;  
uniform float offsetY;  
uniform float offsetZ;  
  
void main() {  
 gl\_Position = projectionMatrix \* modelViewMatrix \* vec4(position.x + offsetX, position.y + offsetY, position.z + offsetZ, 1.0);  
}

* 1. **Фрагментный шейдер.**

uniform float redColor;  
uniform float greenColor;  
uniform float blueColor;  
  
  
void main() {  
 gl\_FragColor = vec4(redColor / 255.0, greenColor / 255.0, blueColor / 255.0, 1.0);  
}

1. **Объекты.**

На изображении, которое должно получится по итогу выделены 4 элемента:

* Буква W.
* Буква O.
* Буква O.
* Буква D.
  1. **Буквы W и D.**

Для реализации использован компонент VglMesh, который представляет объекты с помощью полигональной сетки с треугольными гранями, геометрия которого задана с помощью компонента VglTextGeometry. Демонстрация букв представлена на рис. 2-3.

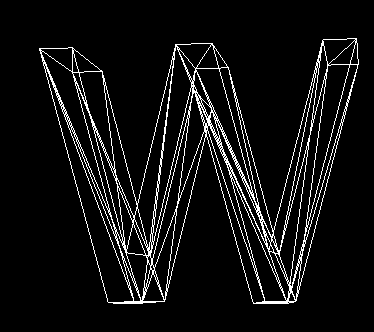


Рисунок 2 – Буква W

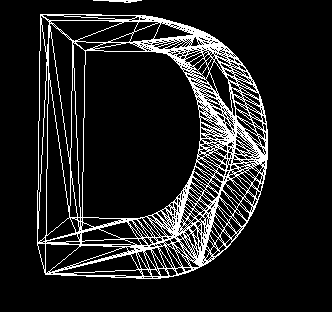


Рисунок 3 – Буква D

* 1. **Буквы O.**

Для реализации использован компонент VglMesh, который представляет объекты с помощью полигональной сетки с треугольными гранями, геометрия которого задана с помощью компонента VglTorusGeometry. Демонстрация букв представлена на рис. 4.

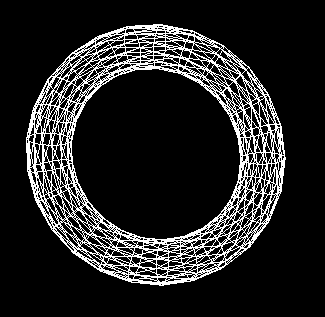


Рисунок 4 – Буква O в виде тора

1. **Интерфейс.**

Добавлены поля изменения смещения по трём осям, изменения цвета через RGB значения и изменения размеров фигур. Поля представлены на рис. 5.



Рисунок 5 – Поля изменения параметров

1. **Демонстрация работы.**

Демонстрация представлена на рис. 6-7.

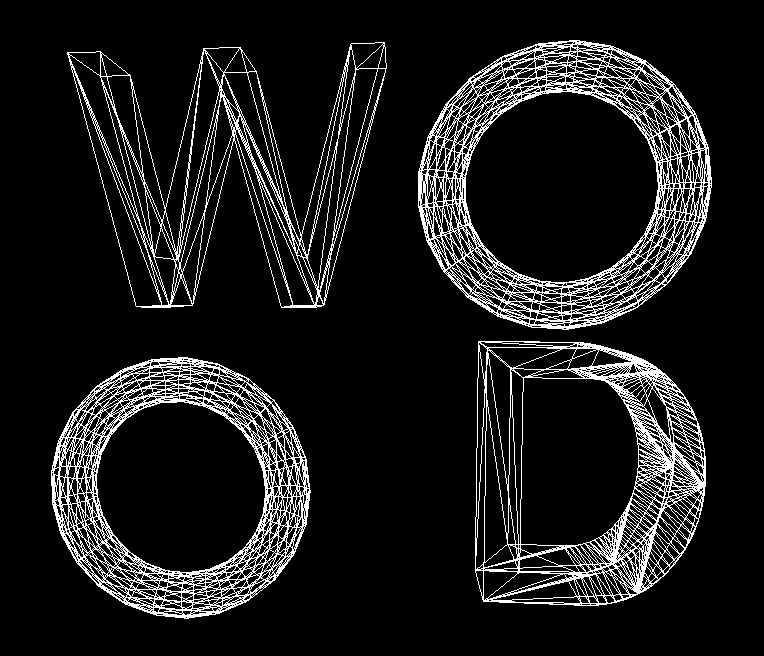


Рисунок 6 – Начальное состояние

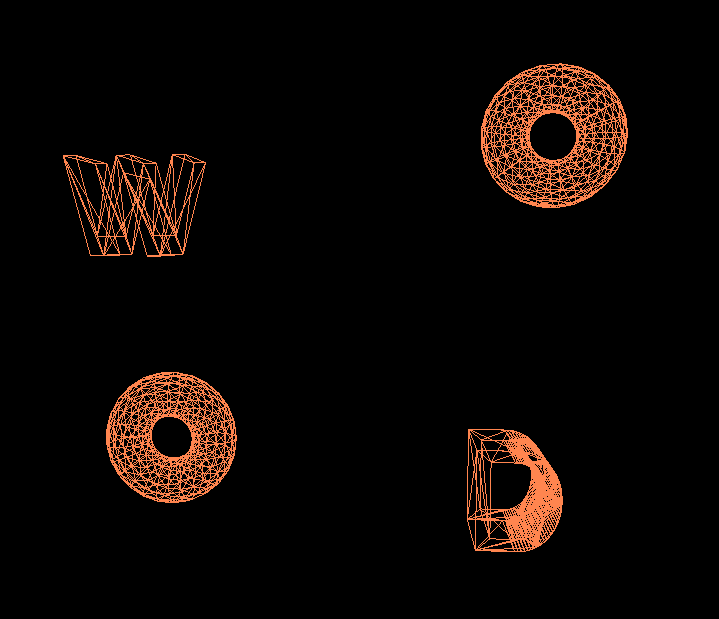
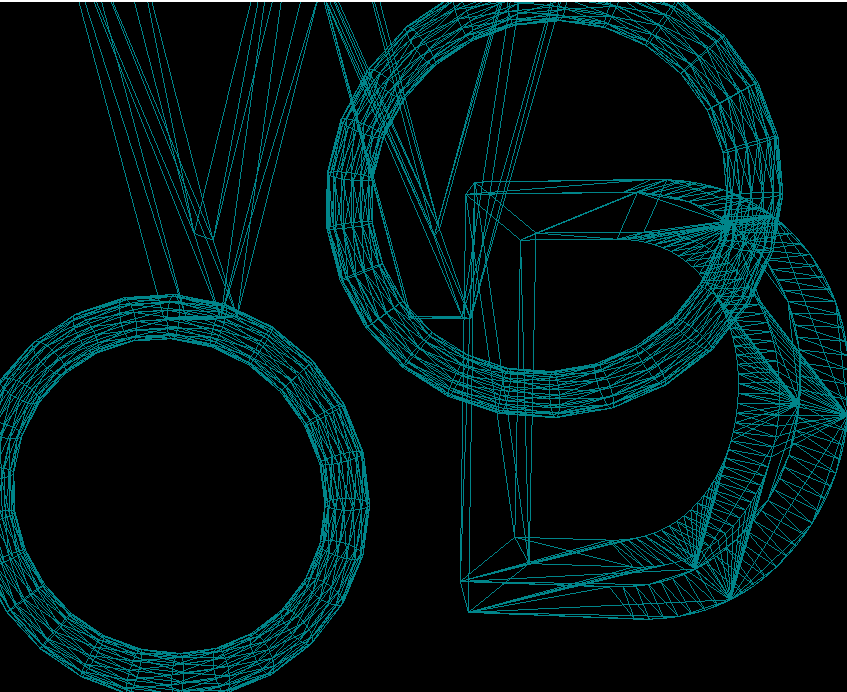
 

Рисунок 7 – Изменены размеры и цвет

**Выводы**

В результате выполнения данной лабораторной работы было выполнено знакомство с основными примитивами WebGL. Для этого был создан canvas, на котором в соответствии с индивидуальным заданием были размещены 4 объекта с возможностью изменения их цвета и размеров.