МИНОБРНАУКИ РОССИИ

Санкт-Петербургский государственный

электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина)

Кафедра МОЭВМ

отчет

по лабораторной работе №2

по дисциплине «Компьютерная 3D графика»

Тема: 3D трансформации

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

Санкт-Петербург

2020

**Цель работы**

Представить 3D сцену и в ней один или несколько объектов из вашего задания (стараться представить все).

Освоить использование стандартных матричных преобразований над этими объектами и осуществить:

* просмотр трансформаций: через lookAt или эквивалентные модельные трансформации;
* трансформацию проекции: через perspective и ortho;
* моделирование трансформации rotate, translate, scale и матричный стек.

**Задание**

Вариант 35.

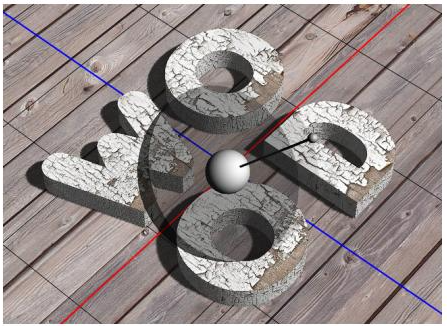


Рисунок 1 – вариант 35

**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.

Шейдеры взяты из предыдущей лабораторной работы:

1. Vertex

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. Fragment

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. **Трансформации.**

Для каждой фигуры сделаны собственные настройки трансформации.

Перенос фигур осуществляется с помощью изменения атрибута position, которой в реальном времени изменяет координаты x, y и z у каждой вершины в используемой геометрии.

Поворот фигур осуществляется с помощью изменения атрибута rotation, который принимает углы Эйлера в радианах относительно каждой из осей и порядок осей, по которому и осуществляет вращательные преобразования.

Изменение размера осуществляется с помощью атрибута size.

Таким образом, одна фигура может быть описана тремя компонентами с настраиваемыми атрибутами. Например, верхний тор:

<vgl-shader-material  
 ref="mat"  
 name="mat"  
 wireframe  
 :uniforms="uniforms"  
 :vertex-shader="vertexShader"  
 :fragment-shader="fragmentShader"  
/>

<vgl-torus-geometry  
 name="torusUp"  
 :radius="parseFloat(sizeMultiply) + parseFloat(figures.torusUp.size)"  
 tube="3"  
 tubular-segments="25"  
 radial-segments="15"  
/>

<vgl-mesh material="mat" name="torusMeshUp" geometry="torusUp"  
 :position="`${figures.torusUp.position.x} ${figures.torusUp.position.y} ${figures.torusUp.position.z}`"  
 :rotation="`${figures.torusUp.rotation.x} ${figures.torusUp.rotation.y} ${figures.torusUp.rotation.z} XYZ`"/>

1. **Камера.**

Для камеры добавлены настройки для изменения сферического положения относительно цели, путём изменения сферических координат (радиуса, зенитного угла и азимутального угла).

Компонент камеры с атрибутами:

<vgl-perspective-camera  
 :orbit-position="`${cameraPosition.r.value} ${cameraPosition.theta.value} ${cameraPosition.phi.value}`"  
 name="mainCamera"/>

1. **Интерфейс.**

Для общей картины добавлены настройки камеры. Настройки представлены на рис. 5.

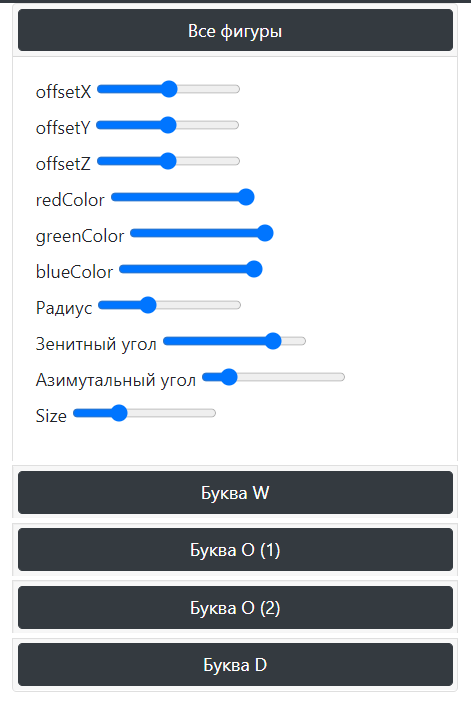


Рисунок 5 – Поля изменения параметров всей картины

Для каждой из фигур добавлены свои настройки. Настройки представлены на рис. 6.

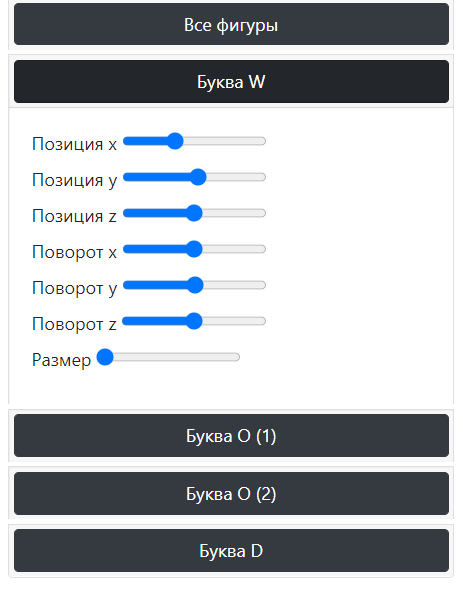


Рисунок 6 – Настройки для отдельной фигуры

1. **Демонстрация работы.**

Демонстрация представлена на рис. 7-9.

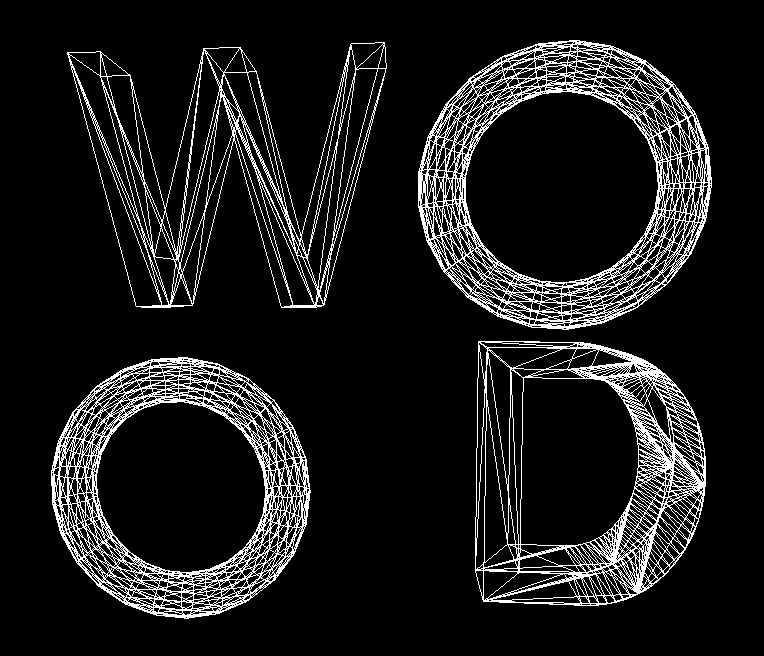


Рисунок 7 – Начальное состояние

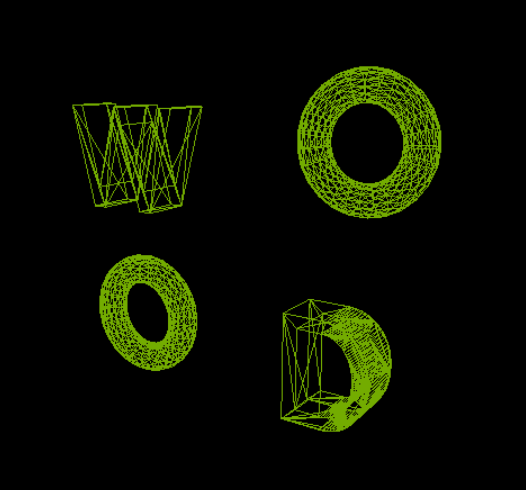


Рисунок 8 – Изменены общие параметры

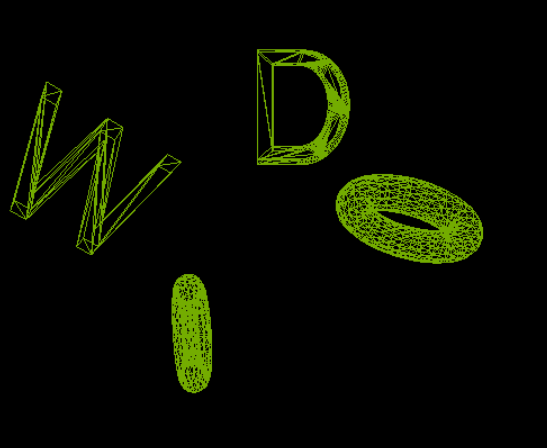


Рисунок 9 – Изменены отдельные настройки фигур

**Выводы**

В результате выполнения данной лабораторной работы было выполнено знакомство с основными 3D трансформациями WebGL. Для этого к предыдущем результатам были добавлены настройки для изменения позиции, размера и поворота для каждой отдельной фигуры.