Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №2

по дисциплине "Программирование графических приложений"

ТЕМА РАБОТЫ:

Работа с базовыми примитивами в WebGL

Выполнил:

студент гр. ПРИм-124

Парахин К.В.

Принял:

Жигалов И.Е.

Владимир 2024 г.

Цель работы:

Изучение принципов построения изображений в WebGL на основе базовых примитивов.

Выполнение работы:

Индивидуальный вариант 2 (14 % 12 = 2)

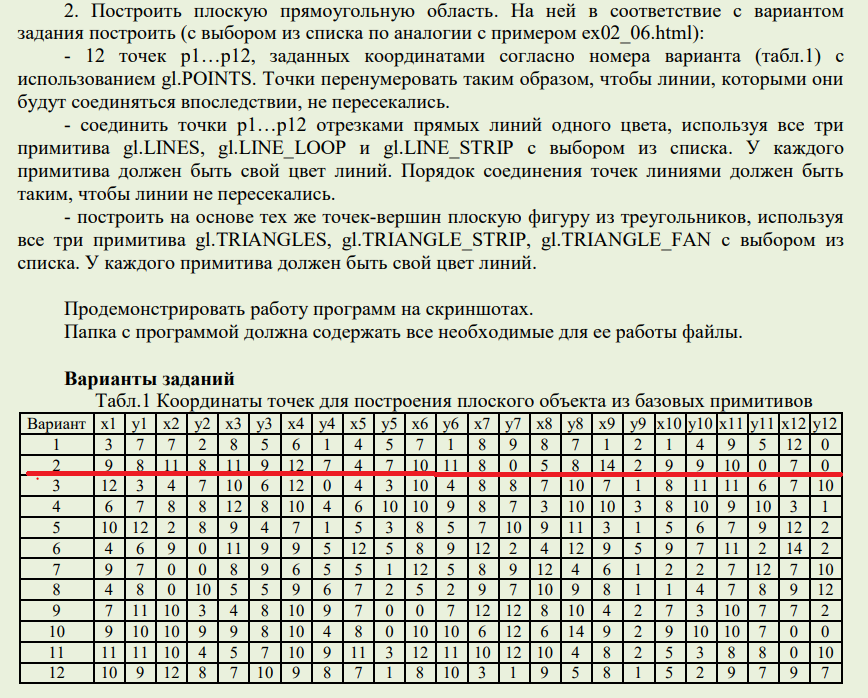




Рисунок 1. Расположение точек согласно варианту

Чтобы линии потом не пересекались, располагаем их в таком порядке: 10, 6, 3, 2, 4, 9, 11, 7, 12, 5, 8, 1

   function initBuffers() {

        vertexBuffer = gl.createBuffer();

        gl.bindBuffer(gl.ARRAY\_BUFFER, vertexBuffer);

        old\_verticles = [

            9, 8, 0,

            11, 8, 0,

            11, 9, 0,

            12, 7, 0,

            4, 7, 0,

            10, 11, 0,

            8, 0, 0,

            5, 8, 0,

            14, 2, 0,

            9, 9, 0,

            10, 0, 0,

            7, 0, 0,

         ];

         new\_verticles = [

            9, 9, 0,

            10, 11, 0,

            11, 9, 0,

            11, 8, 0,

            12, 7, 0,

            14, 2, 0,

            10, 0, 0,

            8, 0, 0,

            7, 0, 0,

            4, 7, 0,

            5, 8, 0,

            9, 8, 0,

         ];

        verticles = new\_verticles.map((x) => x / 10 - 0.5);

        gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(verticles), gl.STATIC\_DRAW);

        colors = [

            1.0, 0.0, 0.0, // red

            1.0, 0.0, 0.0,

            1.0, 0.0, 0.0,

            0.0, 1.0, 0.0, // green

            0.0, 1.0, 0.0,

            0.0, 1.0, 0.0

        ];

        vertexBuffer.itemSize = 3;

        vertexBuffer.numItems = 12;

    }

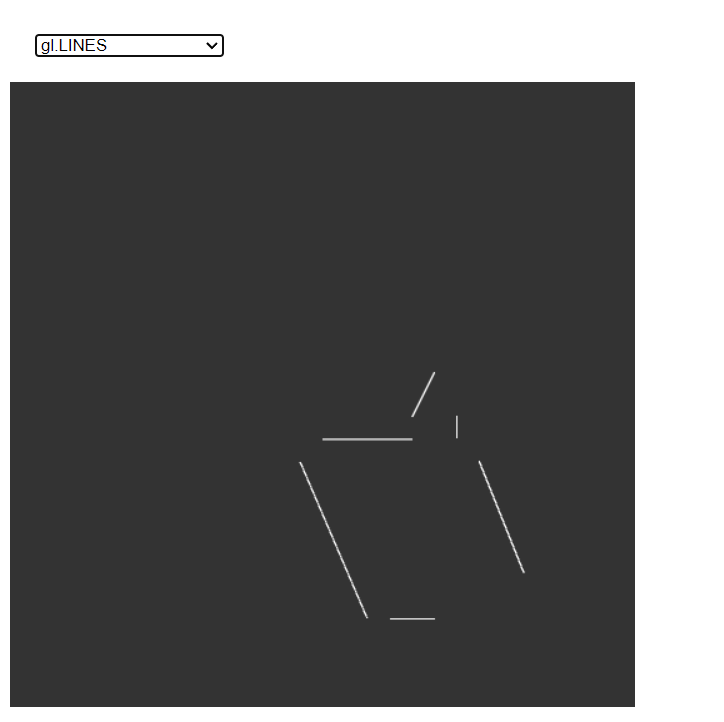


Рисунок 2.1. Режим Lines

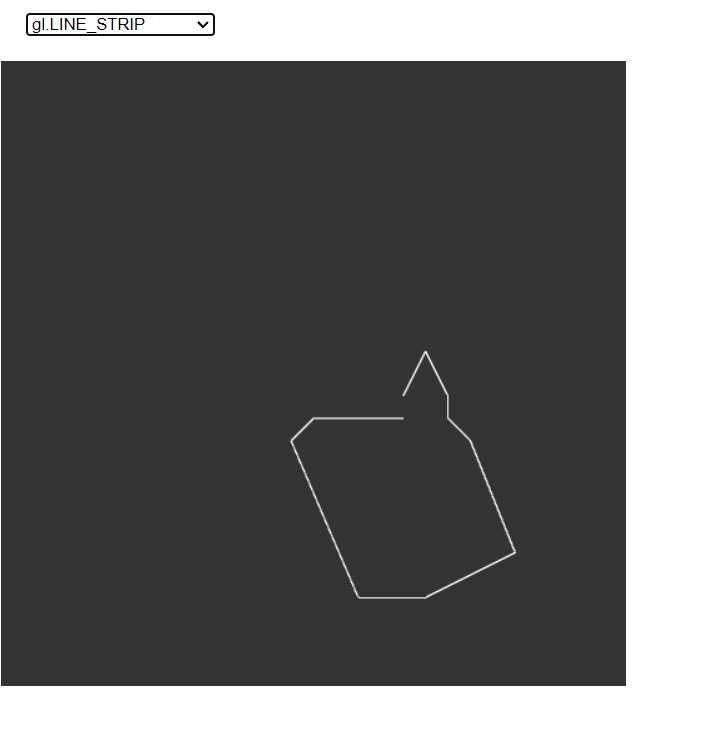


Рисунок 2.2. Режим Lines Strip

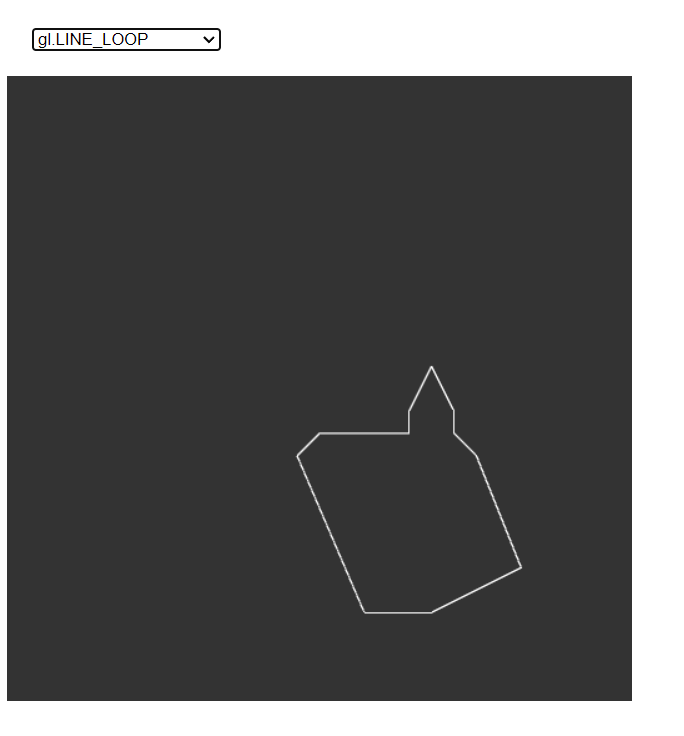


Рисунок 2.3. Режим Line Loop

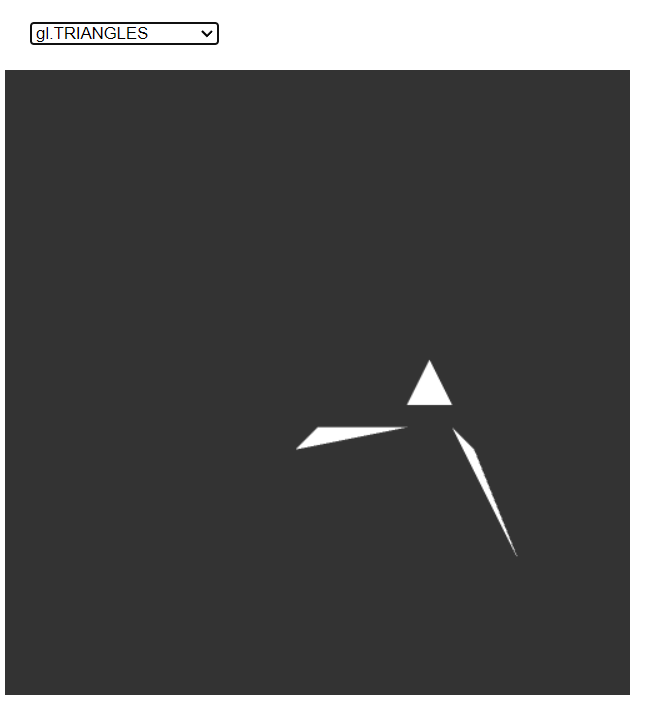


Рисунок 3.1. Режим Triangles



Рисунок 3.2. Режим Triangles Strip

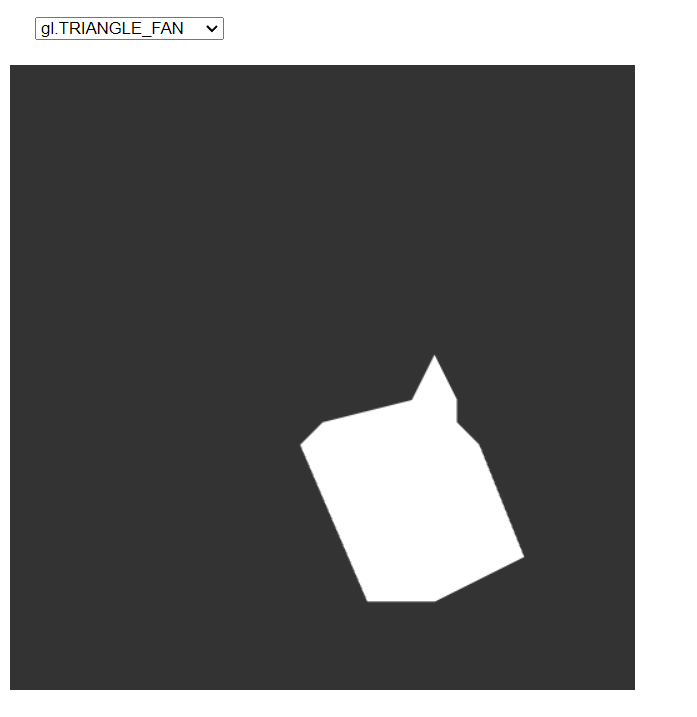


Рисунок 3.3. Режим Triangles Fan

Со цветом линий я не разобрался, буфер цвета (массив colors) в методе initBuffers пытался использовать – но не получилось раскрасить линии и точки.

Вывод

В результате выполнения работы я освоил основные принципы построения изображений в WebGL на основе базовых примитивов.