Липецкий государственный технический университет

Кафедра АСУ

Лабораторная работа №3

по организации графических систем и систем мультимедиа

3D сцена

Студент Бровкин А.Н.

Группа М-АС-19

Руководитель Кургасов В.В.

Липецк

2020

**Задание кафедры**

Реализовать программный продукт с 3D сценой, в которой есть как минимум три объекта.

# Теоретические сведения

Приложение было разработано с использованием языка JavaScript.

Изначально JavaScript был создан, чтобы «сделать веб-страницы живыми». Программы на этом языке называются скриптами. Они могут встраиваться в HTML и выполняться автоматически при загрузке веб-страницы.

Скрипты распространяются и выполняются, как простой текст. Им не нужна специальная подготовка или компиляция для запуска.

Сегодня JavaScript может выполняться не только в браузере, но и на сервере или на любом другом устройстве, которое имеет специальную программу, называющуюся «движком» JavaScript.

У браузера есть собственный движок, который иногда называют «виртуальная машина JavaScript».

Современный JavaScript – это «безопасный» язык программирования. Он не предоставляет низкоуровневый доступ к памяти или процессору, потому что изначально был создан для браузеров, не требующих этого.

Возможности JavaScript сильно зависят от окружения, в котором он работает. Например, Node.JS поддерживает функции чтения/записи произвольных файлов, выполнения сетевых запросов и т.д.

В браузере для JavaScript доступно всё, что связано с манипулированием веб-страницами, взаимодействием с пользователем и веб-сервером.

Например, в браузере JavaScript может:

* Добавлять новый HTML-код на страницу, изменять существующее содержимое, модифицировать стили.
* Реагировать на действия пользователя, щелчки мыши, перемещения указателя, нажатия клавиш.
* Отправлять сетевые запросы на удалённые сервера, скачивать и загружать файлы (технологии AJAX и COMET).
* Получать и устанавливать куки, задавать вопросы посетителю, показывать сообщения.
* Запоминать данные на стороне клиента («local storage»).

# Библиотека Three.js

Three.js — легковесная кроссбраузерная библиотека JavaScript, используемая для создания и отображения анимированной компьютерной 3D графики при разработке веб-приложений. Three.js скрипты могут использоваться совместно с элементом HTML5 CANVAS, SVG или WebGL. Исходный код расположен в репозитории GitHub (<https://github.com/mrdoob/three.js>).

Большинство информации о данной библиотеке на английском языке.

Эта библиотека достаточно популярна так как многие пользователи выбирают именно ее для 3D моделирования в браузере, это говорит о достаточно большом количестве звезд у данного проекта.

Так же эта библиотека постоянно обновляется. Последний выпуск новой версии 0.115.0 был произведен 25 марта 2020 года. Так же разработка ведется и сейчас (последний коммит был 24 апреля).

Данные указаны на момент 25 апреля 2020 года.

# Листинг кода

var raycaster = new THREE.Raycaster();

var mouseCoords = new THREE.Vector2();

var loader = new THREE.TextureLoader()

var camera

var controls

var meshSphere

var renderer

var scene

var plane

var cubeLeftDown

var planeGeometry

var planeMaterial

var ball = {

    rotationX: 0,

    rotationY: 0,

    rotationZ: 0,

    positionX: 0,

    positionY: 200,

    positionZ: 0

};

var light

function setPosition(meshObject, x, y, z){

    meshObject.position.x = x;

    meshObject.position.y = y;

    meshObject.position.z = z;

}

window.onload = function(){

    var width = window.innerWidth;

    var height = window.innerHeight;

    var canvas = this.document.getElementById('canvas');

    canvas.width = width;

    canvas.height = height;

    var gui = new dat.GUI();

    gui.add(ball, 'rotationX').min(-0.1).max(0.1).step(0.0001);

    gui.add(ball, 'rotationY').min(-0.1).max(0.1).step(0.0001);

    gui.add(ball, 'rotationZ').min(-0.1).max(0.1).step(0.0001);

    gui.add(ball, 'positionX').min(-300).max(300).step(0.01);

    gui.add(ball, 'positionY').min(-300).max(300).step(0.01);

    gui.add(ball, 'positionZ').min(-300).max(300).step(0.01);

    renderer = new THREE.WebGLRenderer({canvas: canvas});

    renderer.setClearColor(0x00000);

    //создаем перспетивную камеру

    camera = new THREE.PerspectiveCamera(45, width / height, 0.1, 5000);

    camera.position.set(0, 0, 1000);

    //создаем сцену

    scene = new THREE.Scene();

    //создаем OrbitCintrols который позволяет вращать камеру используя мышь

    controls = new OrbitControls( camera, renderer.domElement );

    //устанавливаем рассеянный источник света

    light = new THREE.AmbientLight(0x555555);

    scene.add(light)

    //инициализурем танцпол

    planeGeometry = new THREE.PlaneGeometry(1000,1000,5, 5);

    planeMaterial = new THREE.MeshPhongMaterial({color: 0xffffff, vertexColors: THREE.FaceColors});

    plane = new THREE.Mesh(planeGeometry, planeMaterial);

    plane.rotation.x = 3 \* Math.PI / 2;

    plane.position.y = -140;

    scene.add(plane);

    //создаем геометрию и материал для колонок

    var spekerGeometry = new THREE.BoxGeometry(120, 85, 120);

    var spekerMaterial = new THREE.MeshPhongMaterial({

        map: loader.load(speakerSrc),

    });

    //создаем для колонок меши и раствляем их по местам

    cubeLeftDown = new THREE.Mesh(spekerGeometry, spekerMaterial);

    setPosition(cubeLeftDown, -220, -100, -150)

    var cubeLeftUp = new THREE.Mesh(spekerGeometry, spekerMaterial);

    setPosition(cubeLeftUp, -220, -15, -150)

    var cubeRightDown = new THREE.Mesh(spekerGeometry, spekerMaterial);

    setPosition(cubeRightDown, 220, -100, -150)

    var cubeRigthUp = new THREE.Mesh(spekerGeometry, spekerMaterial);

    setPosition(cubeRigthUp, 220, -15, -150)

    scene.add(cubeLeftDown);

    scene.add(cubeLeftUp);

    scene.add(cubeRightDown);

    scene.add(cubeRigthUp);

    //создаем шар

    var geometry = new THREE.SphereGeometry(100, 30, 30);

    var material = new THREE.MeshPhongMaterial({color: 0xFFFFFF, vertexColors: THREE.FaceColors});

    //делаем его разноцветным

    for(var i = 0; i < geometry.faces.length; i++){

        geometry.faces[i].color.setRGB(Math.random(), Math.random(), Math.random())

    }

    meshSphere = new THREE.Mesh(geometry, material);

    meshSphere.position.y = 200

    scene.add(meshSphere);

    animate()

    initRaycast()

}

var lasttime = 0

var deltaTime = 500

var needAnimate = false

function animate() {

    requestAnimationFrame( animate );

    //меняем параметры для шара

    meshSphere.rotation.y += ball.rotationY;

    meshSphere.rotation.x += ball.rotationX;

    meshSphere.rotation.z += ball.rotationZ;

    meshSphere.position.y = ball.positionY;

    meshSphere.position.x = ball.positionX;

    meshSphere.position.z = ball.positionZ;

    //получем время чтобы танцпол менялся через каждые 0.5 секунд

    var currentTime = new Date().getTime()

    if (needAnimate && currentTime-lasttime >= deltaTime){

        lasttime = currentTime

        //пересоздаем танцпол

        scene.remove(plane)

        planeMaterial = new THREE.MeshPhongMaterial({color: 0xffffff, vertexColors: THREE.FaceColors});

        planeGeometry = new THREE.PlaneGeometry(1000,1000,5, 5);

        //задем цвет по прямоугольникам

        for(var i = 0; i < planeGeometry.faces.length; i+=2){

            var rgb = {

                r: Math.random(),

                g: Math.random(),

                b: Math.random()

            }

            planeGeometry.faces[i].color.setRGB(rgb.r, rgb.g, rgb.b)

            planeGeometry.faces[i+1].color.setRGB(rgb.r, rgb.g, rgb.b)

        }

        plane = new THREE.Mesh(planeGeometry, planeMaterial)

        plane.rotation.x = 3 \* Math.PI / 2;

        plane.position.y = -140;

        scene.add(plane)

    }

    controls.update();

    renderer.render( scene, camera );

}

var raycaster, mouse = { x : 0, y : 0 };

//инициализируем рейкаст для улавливания кликов мыши на сцене

function initRaycast () {

    raycaster = new THREE.Raycaster();

    renderer.domElement.addEventListener( 'click', raycast, false );

}

function raycast ( e ) {

    //поймали события нажатия на сцену, считываем параметры для мыши

    mouse.x = ( e.clientX / window.innerWidth ) \* 2 - 1;

    mouse.y = - ( e.clientY / window.innerHeight ) \* 2 + 1;

    //устанавливаем чтобы рейкастер работал относительно камеры

    raycaster.setFromCamera( mouse, camera);

    //получаем все элементы с которым пересекался луч рейкаста

    var intersects = raycaster.intersectObjects( scene.children );

    for ( var i = 0; i < intersects.length; i++ ) {

        //если мы нажали на колнку то запускаем музыку

        if (intersects[i].object.geometry.type == "BoxGeometry"){

            console.log( intersects[ i ] );

            needAnimate = !needAnimate

            var audio = document.getElementById("audio")

            if (needAnimate){

                audio.play()

                ball.rotationY = 0.005

                //меняем освещение на более яркое

                scene.remove(light)

                light = new THREE.AmbientLight(0xffffff);

                scene.add(light)

            } else {

                audio.pause()

                ball.rotationY = 0

                //меняем освещение на более тусклое

                scene.remove(light)

                light = new THREE.AmbientLight(0x555555);

                scene.add(light)

            }

            return

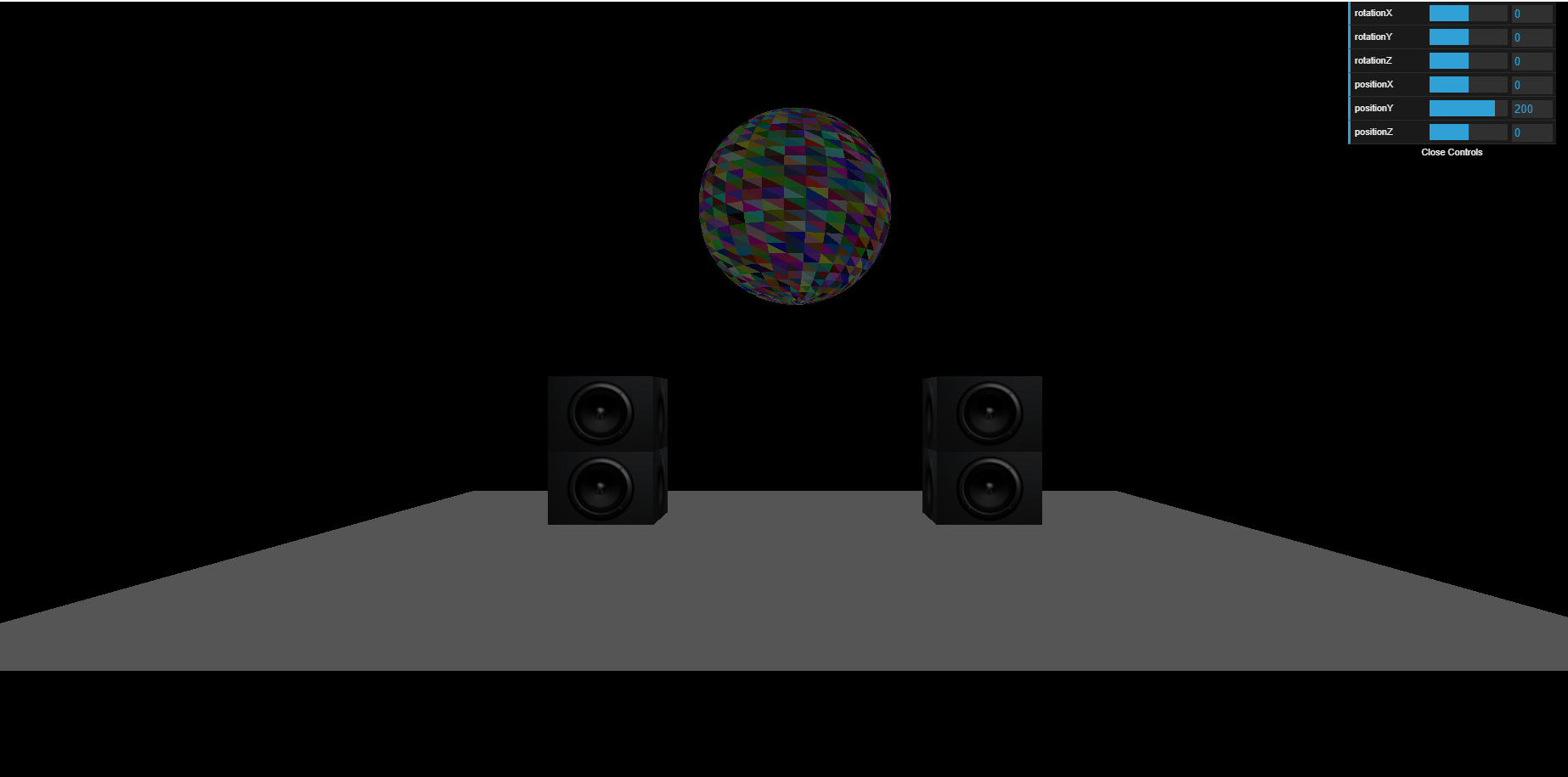
        }

    }

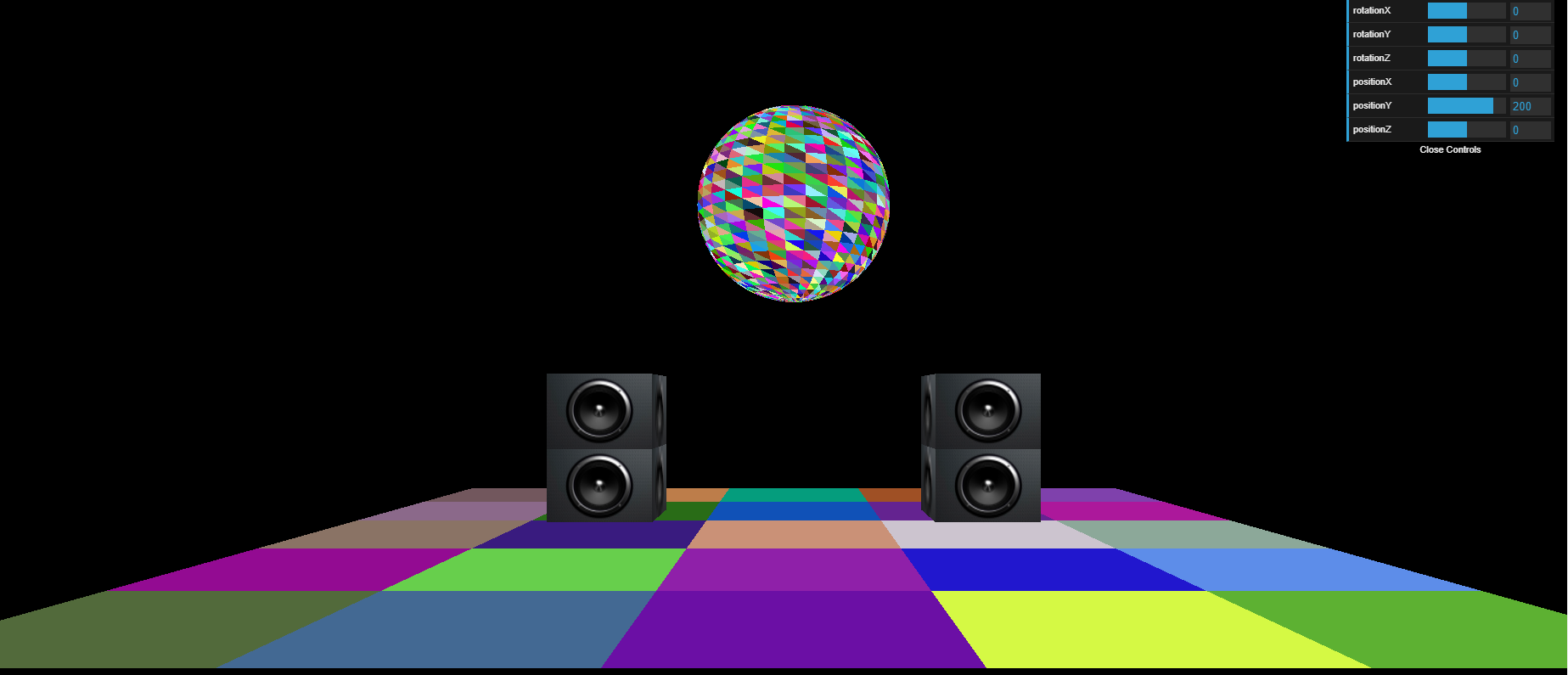
}

# Результаты выполнения работы

На рисунках 1-2 показаны результаты выполнения работы.



1. Начальное изображение



1. Изображение после нажатия на колонку

**Вывод**

В ходе выполнения данной лабораторной работы были получил практические навыки составления 3d сцены. Так же был получен опыт в использовании библиотеки Three.js которая позволяет намного быстрее создавать необходимые 3d объекты.