Федеральное агентство по образованию

ФГБОУ ВО Уфимский государственный авиационный технический

университет

Кафедра ВМиК

ЛАБОРАТОРНАЯ РАБОТА №4

По дисциплине: «Инженерная и компьютерная графика»

«ПРОГРАММИРОВАНИЕ WebGL ГРАФИКИ»

Выполнил: студент группы ПРО-222

Сафронов И.А.

Проверил: Котельников В.А.

Уфа 2021

1. **Цель работы:** ознакомиться с WebGL-графикой.
2. **Задачи:**

* Рассмотреть примеры компьютерной графики в html5 (открывать в IE).
* Визуализировать Солнечную систему.

1. **Ход** **работы**

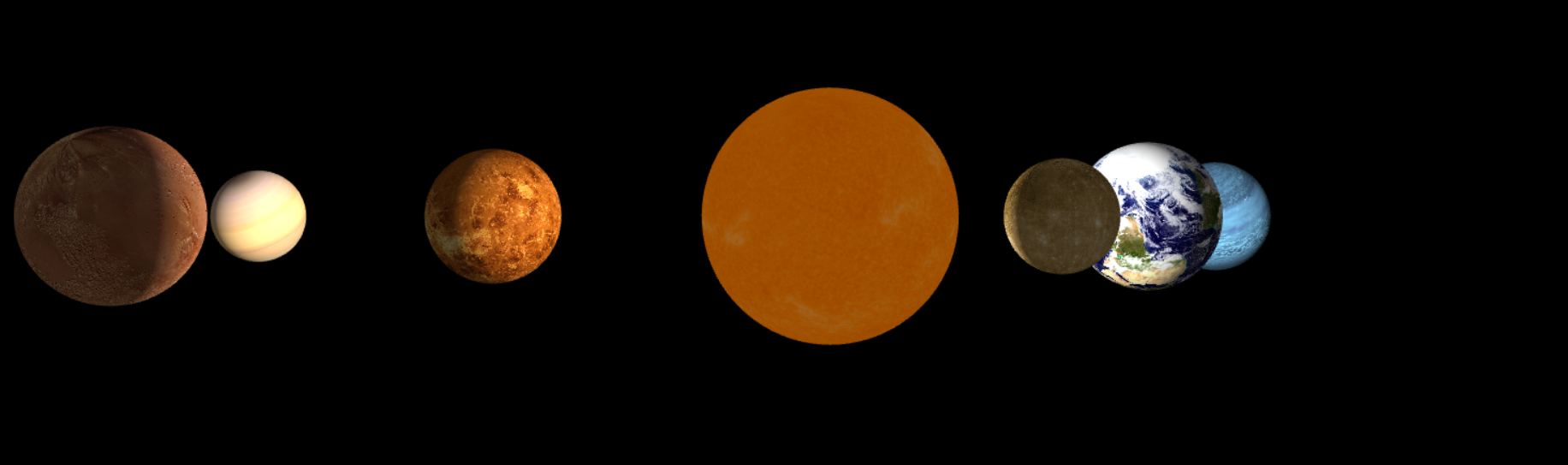


Рис. 1 Стартовое изображение

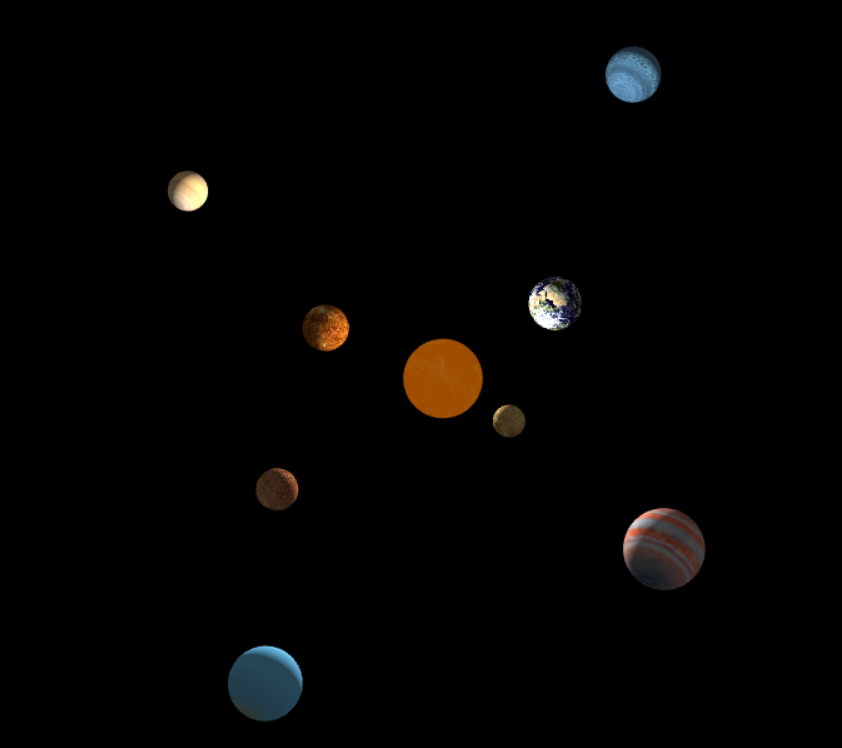


Рис. 2 Вид сверху

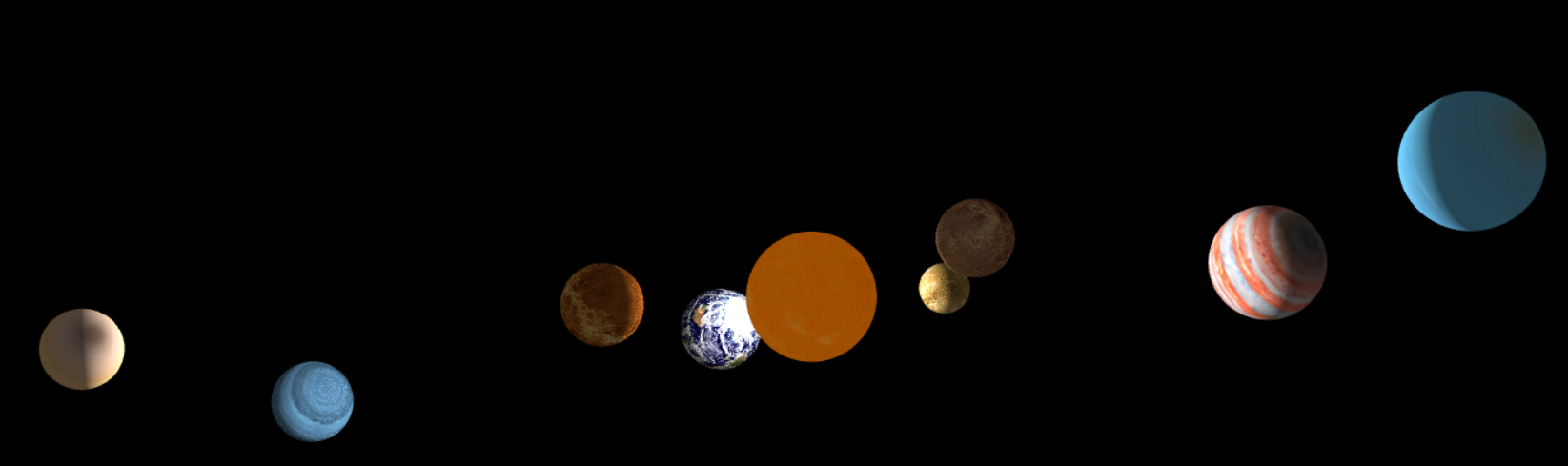


Рис. 3 Вид сбоку

1. **Вывод:**

В лабораторной работе №4 «Программирование WebGL графики» я ознакомился с WebGL-графикой.

Приложение 1. Листинг программы:

**index.html:**

<!doctype html>

<html lang="en">

<head>

<title>Солнечная система</title>

<style>

body {

background: #000;

}

</style>

<script src="Three.js"></script>

<script src="https://threejs.org/build/three.js"></script>

<script src="Detector.js"></script>

</head>

<body>

<script>

var radius = 6371,

rotationSpeed = 0.1,

moonScale = 0.23,

height = window.innerHeight,

width = window.innerWidth,

container,

camera, controls, scene, renderer,

geometry, meshMoon,

dirLight, ambientLight,

clock = new THREE.Clock();

window.onload = function () {

if (!Detector.webgl) {

Detector.addGetWebGLMessage();

return;

}

init();

animate();

}

function init() {

container = document.createElement('div');

document.body.appendChild(container);

scene = new THREE.Scene();

renderer = new THREE.WebGLRenderer({ clearAlpha: 1, clearColor: 0x000000, antialias: true });

renderer.setSize(width, height);

renderer.gammaInput = true;

renderer.gammaOutput = true;

container.appendChild(renderer.domElement);

camera = new THREE.PerspectiveCamera(25, width / height, 50, 1e7);

camera.position.z = radius ;

scene.add(camera);

controls = new THREE.TrackballControls(camera, renderer.domElement);

controls.rotateSpeed = 1.0;

controls.zoomSpeed = 1.2;

controls.panSpeed = 0.2;

controls.noZoom = false;

controls.noPan = false;

controls.staticMoving = false;

controls.dynamicDampingFactor = 0.3;

controls.minDistance = radius \* 1.1;

controls.maxDistance = radius \* 1000;

controls.keys = [65, 83, 68]; // [ rotateKey, zoomKey, panKey ]

dirLight = new THREE.PointLight(0xFFFFFF,1);

dirLight.position.set(0, 0, 0).normalize();

scene.add(dirLight);

scene.add(new THREE.AmbientLight(0x404040));

var shader = THREE.ShaderUtils.lib["normal"],

uniforms = THREE.UniformsUtils.clone(shader.uniforms);

dist = 1000;

//Солнце

scene.add(new THREE.AmbientLight(0x404040));

Sun\_light = new THREE.PointLight(0xffffff);

Sun\_light.position.set(0, 0, 0);

scene.add(Sun\_light);

Sun\_rad = 500;

Sun\_materials = [

new THREE.MeshLambertMaterial({ ambient: 0xbbbbbb, map: THREE.ImageUtils.loadTexture('Sun.jpg') }),

new THREE.MeshBasicMaterial({ color: 0xff8000, transparent: true, opacity: 0.5 })

];

Sun\_object = THREE.SceneUtils.createMultiMaterialObject(new THREE.SphereGeometry(Sun\_rad, 100, 50), Sun\_materials);

Sun\_object.position.set(0, 0, 0);

scene.add(Sun\_object);

//Меркурий

Mercury\_rad = 200;

Mercury\_geometry = new THREE.SphereGeometry(Mercury\_rad, 100, 50);

var Mercury\_Texture = THREE.ImageUtils.loadTexture("Mercury.jpg");

var Mercury\_materials = new THREE.MeshPhongMaterial({ color: 0xffffff, map: Mercury\_Texture });

Mercury = new THREE.Mesh(Mercury\_geometry, Mercury\_materials);

Mercury.position.set(0.8\*dist, 0, 0.8\*dist);

scene.add(Mercury);

//Венера

Venera\_rad = 300;

Venera\_geometry = new THREE.SphereGeometry(Venera\_rad, 100, 50);

var Venera\_Texture = THREE.ImageUtils.loadTexture("Venera.jpg");

var Venera\_materials = new THREE.MeshPhongMaterial({ color: 0xffffff, map: Venera\_Texture });

Venera = new THREE.Mesh(Venera\_geometry, Venera\_materials);

Venera.position.set(-1.5\*dist, 0, -dist);

scene.add(Venera);

//Земля

Earth\_rad = 350;

Earth\_geometry = new THREE.SphereGeometry(Earth\_rad, 100, 50);

var Earth\_Texture = THREE.ImageUtils.loadTexture("Earth.jpg");

var Earth\_materials = new THREE.MeshPhongMaterial({ color: 0xffffff, map: Earth\_Texture });

Earth = new THREE.Mesh(Earth\_geometry, Earth\_materials);

Earth.position.set(1.5\*dist, 0, -1.5\*dist);

scene.add(Earth);

//Марс

Mars\_rad = 250;

Mars\_geometry = new THREE.SphereGeometry(Mars\_rad, 100, 50);

var Mars\_Texture = THREE.ImageUtils.loadTexture("Mars.jpg");

var Mars\_materials = new THREE.MeshPhongMaterial({ color: 0xffffff, map: Mars\_Texture });

Mars = new THREE.Mesh(Mars\_geometry, Mars\_materials);

Mars.position.set(-2\*dist, 0, 2\*dist);

scene.add(Mars);

//Юпитер

Jupiter\_rad = 470;

Jupiter\_geometry = new THREE.SphereGeometry(Jupiter\_rad, 100, 50);

var Jupiter\_Texture = THREE.ImageUtils.loadTexture("Jupiter.jpg");

var Jupiter\_materials = new THREE.MeshPhongMaterial({ color: 0xffffff, map: Jupiter\_Texture });

Jupiter = new THREE.Mesh(Jupiter\_geometry, Jupiter\_materials);

Jupiter.position.set(2.5\*dist, 0, 3\*dist);

scene.add(Jupiter);

//Сатурн

Saturn\_rad = 280;

Saturn\_geometry = new THREE.SphereGeometry(Saturn\_rad, 100, 50);

var Saturn\_Texture = THREE.ImageUtils.loadTexture("Saturn.jpg");

var Saturn\_materials = new THREE.MeshPhongMaterial({ color: 0xffffff, map: Saturn\_Texture });

Saturn = new THREE.Mesh(Saturn\_geometry, Saturn\_materials);

Saturn.position.set(-3.5\*dist, 0, -4\*dist);

scene.add(Saturn);

//Уран

Uran\_rad = 400;

Uran\_geometry = new THREE.SphereGeometry(Uran\_rad, 100, 50);

var Uran\_Texture = THREE.ImageUtils.loadTexture("Uran.jpg");

var Uran\_materials = new THREE.MeshPhongMaterial({ color: 0xffffff, map: Uran\_Texture });

Uran = new THREE.Mesh(Uran\_geometry, Uran\_materials);

Uran.position.set(-2\*dist, 0, 5\*dist);

scene.add(Uran);

//Нептун

Neptune\_rad = 420;

Neptune\_geometry = new THREE.SphereGeometry(Neptune\_rad, 100, 50);

var Neptune\_Texture = THREE.ImageUtils.loadTexture("Neptune.jpg");

var Neptune\_materials = new THREE.MeshPhongMaterial({ color: 0xffffff, map: Neptune\_Texture });

Neptune = new THREE.Mesh(Neptune\_geometry, Neptune\_materials);

Neptune.position.set(3\*dist, 0, -7\*dist);

scene.add(Neptune);

window.addEventListener('resize', onWindowResize, false);

};

function onWindowResize(event) {

width = window.innerWidth;

height = window.innerHeight;

renderer.setSize(width, height);

camera.aspect = width / height;

camera.updateProjectionMatrix();

controls.screen.width = width;

controls.screen.height = height;

camera.radius = (width + height) / 4;

};

function animate() {

requestAnimationFrame(animate);

render();

stats.update();

};

function render() {

for (var i = 0, l = scene.children.length; i < l; i++) {

var object = scene.children[i];

object.rotation.x += 0.005;

object.rotation.y += 0.005;

object.rotation.z += 0.005;

}

controls.update();

renderer.clear();

renderer.render(scene, camera);

};

</script>

</body>

</html>

Приложение 2. Скриншот коммитов

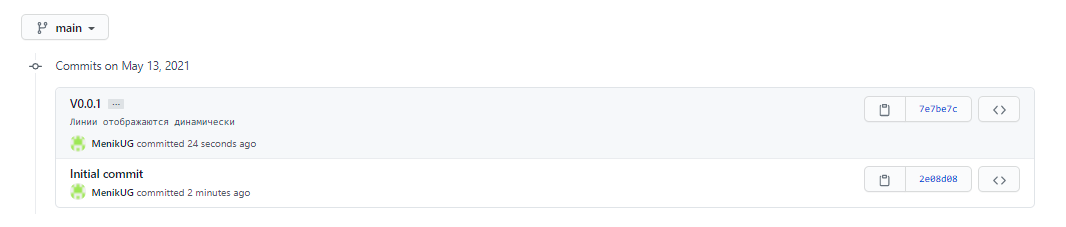


Рис. 4 Скриншот коммитов

Приложение 3. Ссылка на GitHub

<https://github.com/MenikUG/3_laba_Graphics>