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 № 12

по дисциплине "Программирование графических приложений"

ТЕМА РАБОТЫ:

Материалы и карты текстур

Выполнил:

студент гр. ПРИм-124

Парахин К.В.

Принял:

Жигалов И.Е.

Владимир 2024 г.

Цель работы:

Изучение методов работы с материалами и картами текстур в WebGL с использованием библиотеки Three.js.

Выполнение работы:

На основе приведенных примеров программ применить материалы и карты текстур к объектам трехмерных сцен, построенным в результате выполнения лабораторной работы по теме 9. В том числе применить несколько материалов к одному объекту. Использовать тени. Применяемые материалы и карты текстур должны соответствовать виду объектов и использовать несколько различных параметров.

Для начала поработаем с простыми текстурами map для сцены с аквариумными рыбками из лабораторной работы 9.

Листинг скрипта:

var scene, camera, renderer, pyramid1, sphere1, pyramid2, sphere2;

    init();

    function init() {

        renderer = new THREE.WebGLRenderer({ canvas: document.getElementById('primer') });

        var width = 783;

        renderer.setSize(width, width \* 2 / 3);

        camera = new THREE.PerspectiveCamera(70, 1.5, 1, 1000);

        camera.translateZ(150);

        scene = new THREE.Scene();

        var texture = new THREE.ImageUtils.loadTexture('fish.jpg');

        texture.wrapS = THREE.RepeatWrapping;

        texture.wrapT = THREE.RepeatWrapping;

        texture.repeat.set(2, 2);

        var obj\_material1 = new THREE.MeshPhongMaterial({

            map:texture

        });

        var obj\_material2 = new THREE.MeshPhongMaterial({

            map:texture

        });

        var radius\_top = 0;

        var radius\_bottom = 16;

        var height = 20;

        var segments = 3;

        var pyramid\_geometry = new THREE.CylinderGeometry(radius\_top, radius\_bottom, height, segments);

        var sphere\_geometry = new THREE.SphereGeometry(20, 6, 8);

        pyramid1 = new THREE.Mesh(pyramid\_geometry, obj\_material1);

        pyramid1.position.set(-40, 60, 0);

        pyramid1.rotateY(-Math.PI / 2);

        pyramid1.translateY(20);

        pyramid1.translateX(-10);

        sphere1 = new THREE.Mesh(sphere\_geometry, obj\_material1);

        sphere1.position.set(-10, 60, 0);

        sphere1.rotateY(-Math.PI / 2);

        pyramid2 = new THREE.Mesh(pyramid\_geometry, obj\_material2);

        pyramid2.position.set(-40, 0, 0);

        pyramid2.rotateY(-Math.PI / 2);

        pyramid2.translateY(20);

        pyramid2.translateX(-10);

        sphere2 = new THREE.Mesh(sphere\_geometry, obj\_material2);

        sphere2.position.set(-10, 0, 0);

        sphere2.rotateY(-Math.PI / 2);

        scene.add(pyramid1, sphere1, pyramid2, sphere2);

        var light = new THREE.AmbientLight(0x666666);

        var light1 = new THREE.DirectionalLight(0xffffff, 0.9);

        light1.position.set(50, 50, 50);

        scene.add(light, light1);

        animate();

    }

    function animate() {

        pyramid1.rotateY(Math.PI / 360);

        sphere1.rotateY(Math.PI / 360);

        pyramid2.rotateY(Math.PI / 360);

        sphere2.rotateY(Math.PI / 360);

        renderer.render(scene, camera);

        requestAnimationFrame(animate);

    }

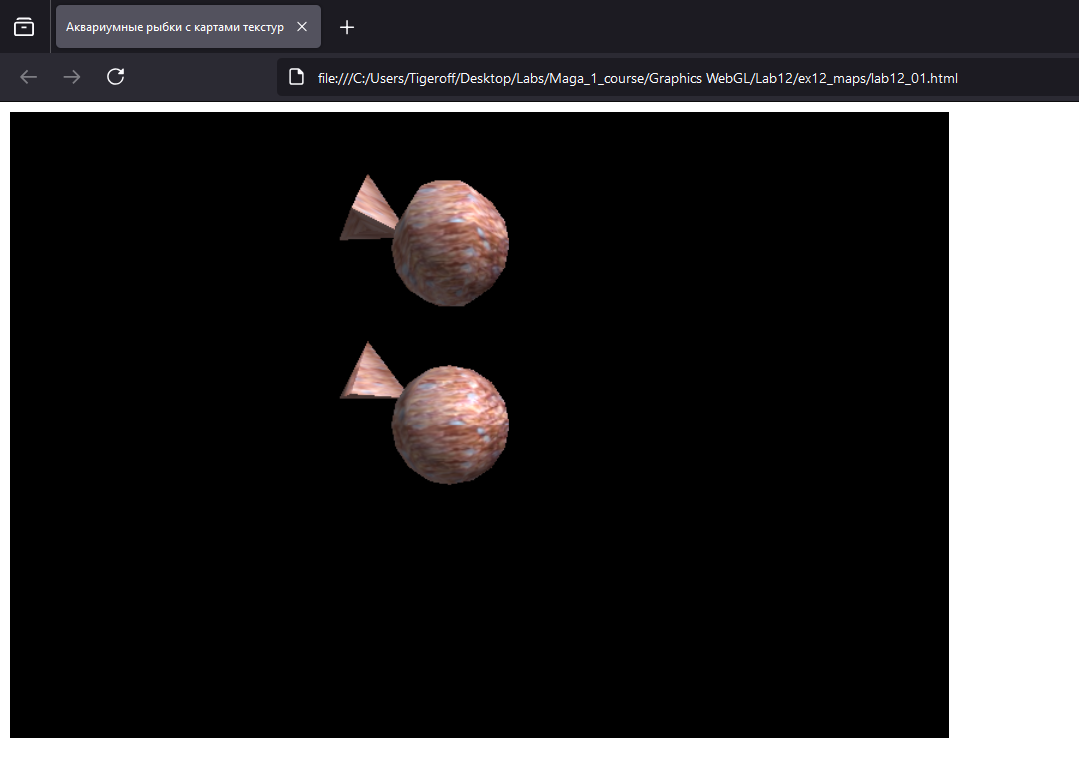


Рисунок 1. 3D объекты с одним материалом

Далее попробуем применить несколько материалов к одному объекту на этой же сцене.

var scene, camera, renderer, pyramid1, sphere1, pyramid2, sphere2;

init();

function init() {

    renderer = new THREE.WebGLRenderer({ canvas: document.getElementById('primer') });

    var width = 783;

    renderer.setSize(width, width \* 2 / 3);

    camera = new THREE.PerspectiveCamera(70, 1.5, 1, 1000);

    camera.position.z = 150;

    scene = new THREE.Scene();

    var obj\_material1 = new THREE.MeshPhongMaterial({ wireframe: false, side: THREE.DoubleSide, color: 0xff0000 });

    var obj\_material2 = new THREE.MeshPhongMaterial({ wireframe: false, side: THREE.DoubleSide, color: 0xaacb00 });

    var radius\_top = 0;

    var radius\_bottom = 16;

    var height = 20;

    var segments = 3;

    var pyramid\_geometry = new THREE.CylinderGeometry(radius\_top, radius\_bottom, height, segments);

    var sphere\_geometry = new THREE.SphereGeometry(20, 6, 8);

    pyramid1 = createMultiMaterialObject(pyramid\_geometry, [obj\_material1, obj\_material2]);

    pyramid1.position.set(-40, 60, 0);

    sphere1 = createMultiMaterialObject(sphere\_geometry, [obj\_material1, obj\_material2]);

    sphere1.position.set(-10, 60, 0);

    pyramid2 = createMultiMaterialObject(pyramid\_geometry, [obj\_material1, obj\_material2]);

    pyramid2.position.set(-40, 0, 0);

    sphere2 = createMultiMaterialObject(sphere\_geometry, [obj\_material1, obj\_material2]);

    sphere2.position.set(-10, 0, 0);

    scene.add(pyramid1, sphere1, pyramid2, sphere2);

    // Освещение

    var light = new THREE.AmbientLight(0x666666);

    var light1 = new THREE.DirectionalLight(0xffffff, 0.9);

    light1.position.set(50, 50, 50);

    scene.add(light, light1);

    animate();

}

function animate() {

    pyramid1.rotation.y += Math.PI / 360;

    sphere1.rotation.y += Math.PI / 360;

    pyramid2.rotation.y += Math.PI / 360;

    sphere2.rotation.y += Math.PI / 360;

    renderer.render(scene, camera);

    requestAnimationFrame(animate);

}

function createMultiMaterialObject(geometry, materials) {

    var group = new THREE.Group();

    materials.forEach(material => {

        var mesh = new THREE.Mesh(geometry, material);

        group.add(mesh);

    });

    return group;

}

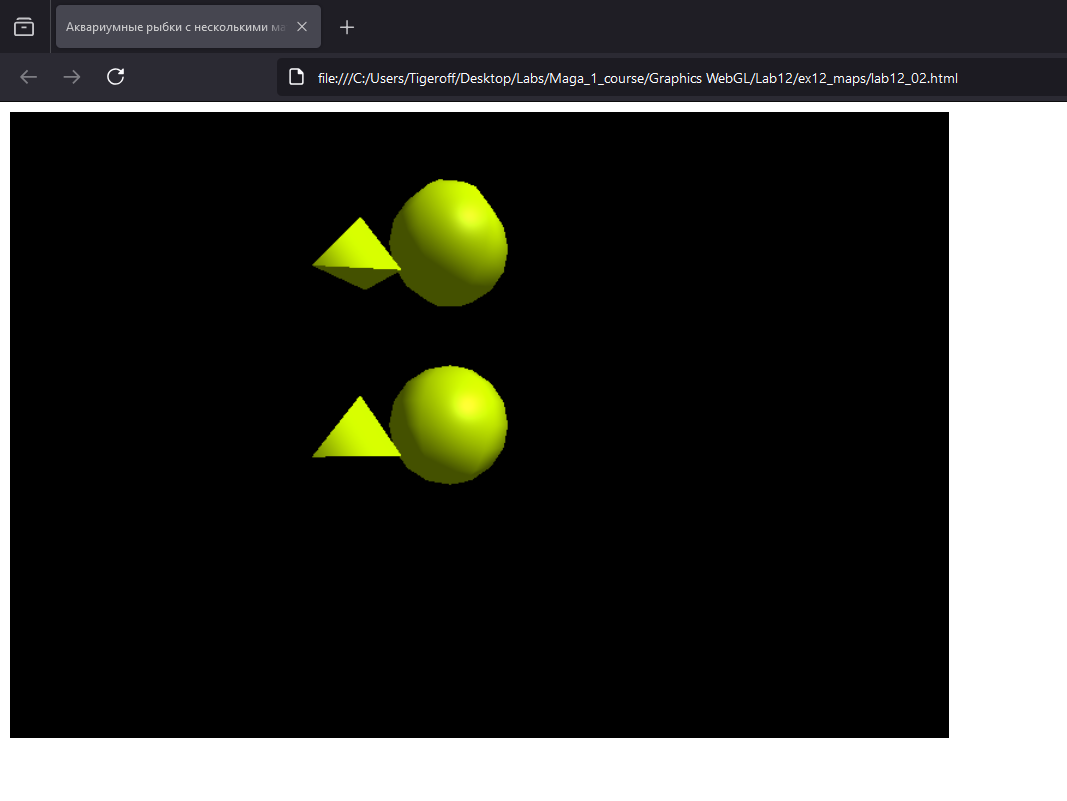


Рисунок 2. 3D объекты с объединением нескольких простых материалов

Вывод

В результате выполнения работы я провел изучение методов работы с материалами и картами текстур в WebGL с использованием библиотеки Three.js.