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 № 9

по дисциплине "Программирование графических приложений"

ТЕМА РАБОТЫ:

Полигональный объекты

Выполнил:

студент гр. ПРИм-124

Парахин К.В.

Принял:

Жигалов И.Е.

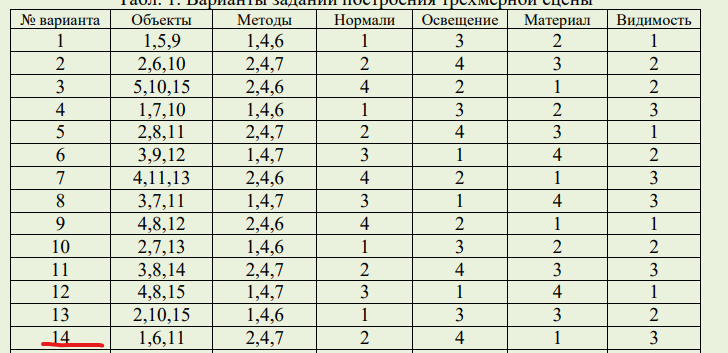
Владимир 2024 г.

Цель работы:

Изучение методов работы с полигональными объектами в WebGL

Выполнение работы:

Индивидуальный вариант 14



Пространственные объекты: прямоугольный параллепипед, призма, тор

Методы: добавление к объекту полигонального объекта, создание копии объекта, построение ограничивающей сферы

Нормали: нормали граней

Источники освещения: полусферическое освещение

Материалы: однородный цвет

Видимость материала: виден снаружи и изнутри

Листинг кода с комментариями по выполнению:

var scene, camera, renderer, box, prizma, clone\_prizma, torus, bounding\_sphere, polygon\_object;

init();

animate();

function init(){

    renderer = new THREE.WebGLRenderer();

    renderer.setSize(600,600);

    document.body.appendChild( renderer.domElement );

    camera = new THREE.PerspectiveCamera( 70, 600 / 600, 1, 1000 );

    camera.translateZ(150);

    scene = new THREE.Scene();

        var obj\_material = [

            // однородный материал видный с двух сторон

            new THREE.MeshPhongMaterial( { wireframe: false, side: THREE.DoubleSide, color: 0xff0000 } )

    ];

    var polygon\_geometry = new THREE.BoxGeometry( 10, 30, 5, 2, 3, 1);

/\*  box = THREE.SceneUtils.createMultiMaterialObject( box\_geometry, obj\_material );

    box.position.set( -70, 60, 0 ); \*/

    var box\_geometry = new THREE.BoxGeometry( 20, 40, 10, 2, 4, 1);

    // присоединение полигонального объекта

    box\_geometry.mergeMesh(polygon\_geometry);

    box = THREE.SceneUtils.createMultiMaterialObject( box\_geometry, obj\_material );

    box.position.set( -70, 60, 0 );

    var radius\_top = 16;

    var radius\_bottom = 16;

    var heigth = 30;

    var segments = 3;

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

    prizma = new THREE.SceneUtils.createMultiMaterialObject( prizma\_geometry, obj\_material );

    prizma.position.set(-20,30,0);

    // копирование объекта

    var clone\_prizma\_geometry = prizma\_geometry.clone();

    clone\_prizma = new THREE.SceneUtils.createMultiMaterialObject( clone\_prizma\_geometry, obj\_material );

    clone\_prizma.position.set(-20,60,0);

    var torus\_geometry = new THREE.TorusGeometry( 15, 5, 8, 16, 2\*Math.PI );

    torus = THREE.SceneUtils.createMultiMaterialObject( torus\_geometry, obj\_material );

    torus.position.set( 50, 0, 0 );

    // вычисление ограничивающей сферы

    var boundingSphereGeometry = torus\_geometry.computeBoundingSphere();

    bounding\_sphere = THREE.SceneUtils.createMultiMaterialObject( boundingSphereGeometry, obj\_material );

    bounding\_sphere.position.set( 50, 0, 0 );

    // освещение полусферическое

    var light = new THREE.HemisphereLight( 0x666666, 0xff0000, 10 );

    // вычисление нормалей к граням

    box\_geometry.computeFaceNormals();

    prizma\_geometry.computeFaceNormals();

    torus\_geometry.computeFaceNormals();

    scene.add( light, box, torus, prizma, clone\_prizma, bounding\_sphere);

}

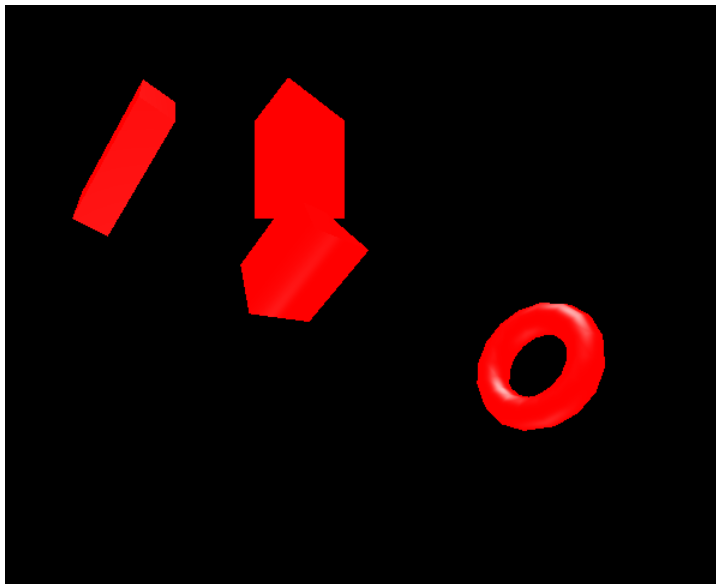


Рисунок 1. Выполнение задания



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

init();

animate();

function init(){

    renderer = new THREE.WebGLRenderer();

    renderer.setSize(600,600);

    document.body.appendChild( renderer.domElement );

    camera = new THREE.PerspectiveCamera( 70, 600 / 600, 1, 1000 );

    camera.translateZ(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 heigth = 20;

    var segments = 3;

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

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

    // рыбка 1

    piramida1 = new THREE.SceneUtils.createMultiMaterialObject( pyramid\_geometry, obj\_material1 );

    piramida1.position.set(-40,60,0);

    piramida1.rotateY(-Math.PI/2);

    piramida1.translateY(20);

    piramida1.translateX(-10);

    sphere1 = THREE.SceneUtils.createMultiMaterialObject( sphere\_geometry, obj\_material1 );

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

    sphere1.rotateY(-Math.PI/2);

    // рыбка 2

    piramida2 = new THREE.SceneUtils.createMultiMaterialObject( pyramid\_geometry, obj\_material2 );

    piramida2.position.set(-40,0,0);

    piramida2.rotateY(-Math.PI/2);

    piramida2.translateY(20);

    piramida2.translateX(-10);

    sphere2 = THREE.SceneUtils.createMultiMaterialObject( sphere\_geometry, obj\_material2 );

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

    sphere2.rotateY(-Math.PI/2);

    // освещение полусферическое

    var light = new THREE.HemisphereLight( 0x666666, 0xff0000, 10 );

    scene.add( light, piramida1, sphere1, piramida2, sphere2);

}

function animate(){

    renderer.render( scene, camera );

    requestAnimationFrame( animate );

}

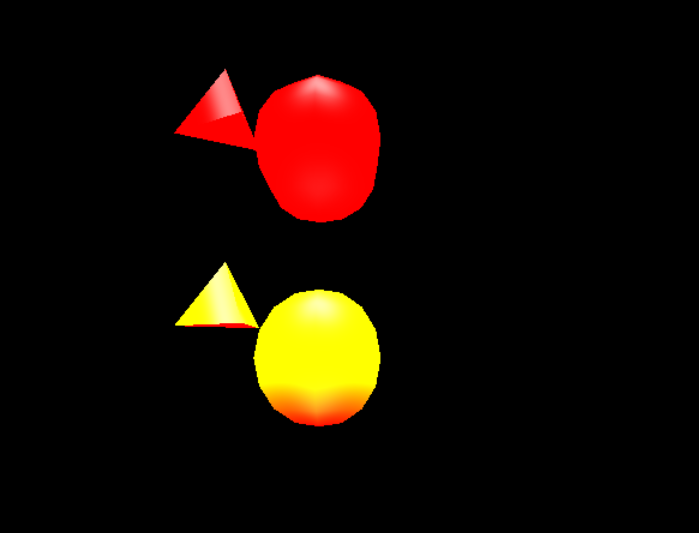


Рисунок 2. Задание 2

Вывод

В результате выполнения работы я изучил методы работы с полигональными объектами в WebGL