Федеральное агентство связи

Сибирский государственный университет телекоммуникаций и информатики

**Межрегиональный учебный центр переподготовки специалистов**

# Контрольная работа

# **по дисциплине: Программирование трёхмерной графики**

# **Конструктор трёхмерных сцен**

**Выполнил**: Сизов Николай Дмитриевич

**Группа**: МИТ-02

**Проверил**: Шлаузер Андрей Иванович

Новосибирск, 2021

**Задание:**

Необходимо разработать веб-приложение, представляющее собой конструктор трёхмерных сцен.

Минимальная сцена приложения должна содержать плоскость с наложенной на неё текстурой (см. лабораторную работу №1) и сферу с текстурой неба (по аналогии с текстурой звёздного неба из лабораторной работы №2).

Пользовательский интерфейс, реализованный на основе библиотеки dat.GUI, должен предоставлять следующие возможности:

1. изменение ландшафта плоскости при помощи инструмента «кисть». Радиус кисти должен регулироваться при помощи колесика мыши;
2. добавление и удаление трёхмерных объектов из файлов (дом, дерево, ограда, куст, птица и др.);
3. выбор, перемещение и поворот объектов в сцене при помощи мыши;

**Результат работы:**

**Часть 1. Изменение ландшафта**

**Задание:** изменение ландшафта плоскости при помощи инструмента «кисть». Радиус кисти должен регулироваться при помощи колесика мыши.

Код добавления курсора:

function addCursor(radius)

{

    var geometry = new THREE.CylinderGeometry( 1.5, 0, 5, 64 );

    var cyMaterial = new THREE.MeshLambertMaterial({color: 0x888888});

    var cylinder = new THREE.Mesh(geometry, cyMaterial);

    scene.add(cylinder);

    return cylinder;

}

Код добавления кисти:

function addCircle(L)

{

    var dashed\_material = new THREE.LineBasicMaterial({

        color: 0xffff00,

    });

    var points = [];

    var k = 360 / L;

    for (var i = 0; i < L; i++)

    {

        var x = Math.cos((i\*k) \* Math.PI / 180);

        var z = Math.sin((i\*k) \* Math.PI / 180);

        points.push(new THREE.Vector3(x, 0, z));

    }

    var geometry = new THREE.BufferGeometry().setFromPoints(points);

    var line = new THREE.LineLoop(geometry, dashed\_material);

    line.computeLineDistances();

    line.scale.set(rad, 1, rad);

    scene.add(line);

    return line;

}

Код добавления кисти:

function hSphere(k, delta){

    var pos = new THREE.Vector3();

    pos.copy(cursor.position);

    var vertices = geometry.getAttribute('position'); // получение массива вершин

    for (var i = 0; i < vertices.array.length; i += 3)

    {

        var x = vertices.array[i];

        var z = vertices.array[i + 2];

        var h = (rad \* rad) - (((x - pos.x) \* (x - pos.x)) + ((z - pos.z)\*(z - pos.z)));

        if (h > 0)

        {

            vertices.array[i + 1] += Math.sqrt(h) \* k \* delta;

        }

    }

    geometry.setAttribute('position', vertices);

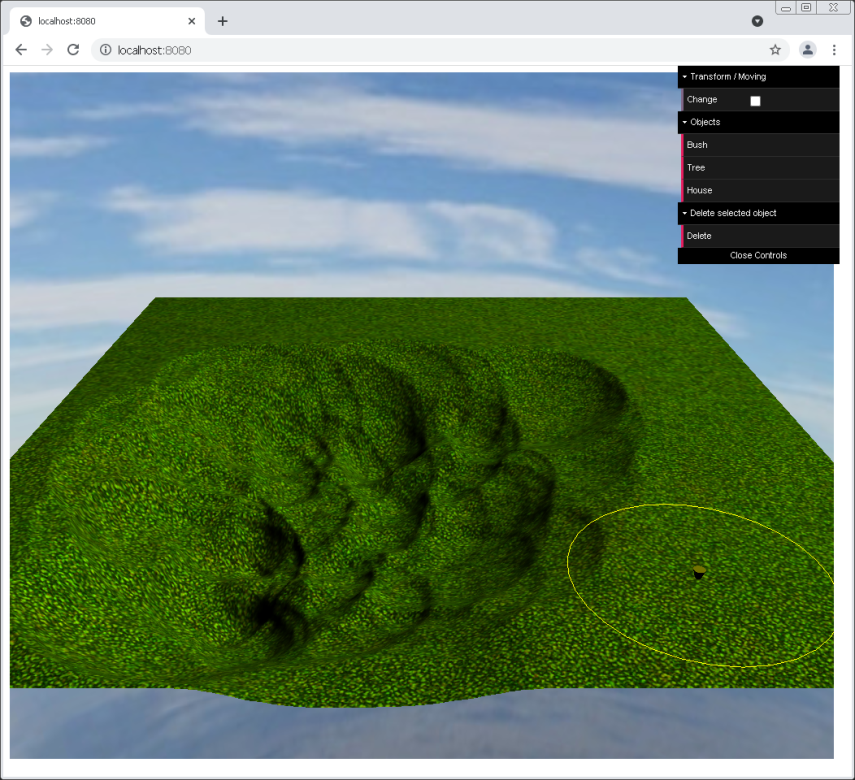
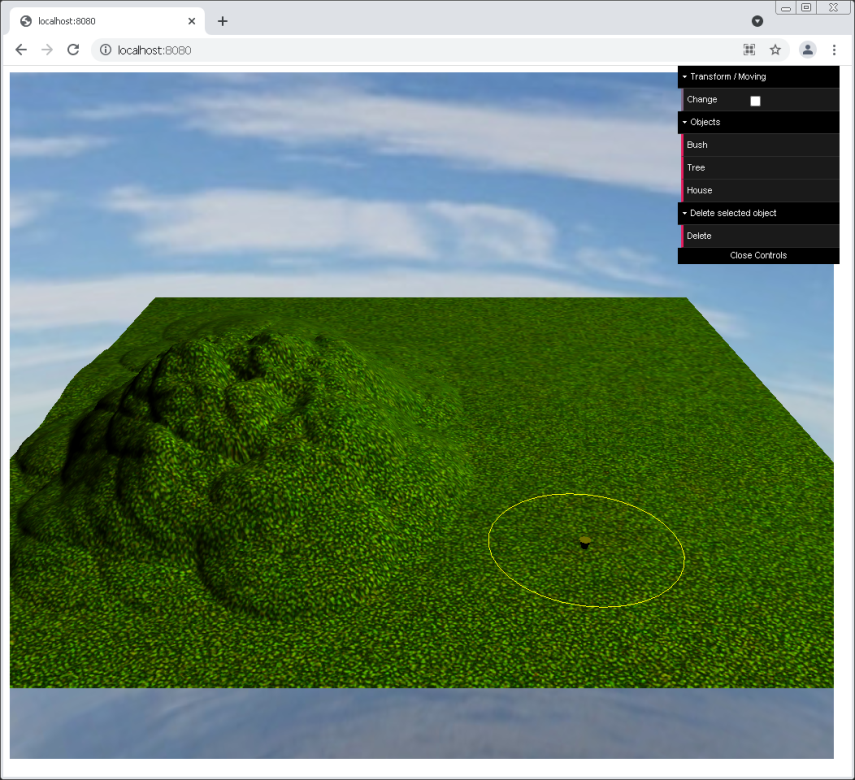
    geometry.computeVertexNormals();

    geometry.attributes.position.needsUpdate = true;

    geometry.attributes.normal.needsUpdate = true;

}

Результат работы:



**Часть 2. Вершины, индексы, цвет**

**Задание:** добавление трёхмерных объектов из файлов (дом, дерево, ограда, куст, птица и др.).

Код загрузки моделей:

function loadModel(path, objName, mtlName, scale)

{

    // функция, выполняемая в процессе загрузки модели (выводит процент загрузки)

    var onProgress = function(xhr) {

        if (xhr.lengthComputable) {

            var percentComplete = xhr.loaded / xhr.total \* 100;

            console.log(Math.round(percentComplete, 2) + '% downloaded' );

        }

    };

    // функция, выполняющая обработку ошибок, возникших в процессе загрузки

    var onError = function(xhr) { };

    var mtlLoader = new THREE.MTLLoader();

    mtlLoader.setPath(path);

    // функция загрузки материала

    mtlLoader.load(mtlName, function(materials)

    {

        materials.preload();

        var objLoader = new THREE.OBJLoader();

        objLoader.setMaterials(materials);

        objLoader.setPath(path);

        // функция загрузки модели

        objLoader.load(objName, function (object)

        {

            object.position.x = 0;

            object.position.y = 0;

            object.position.z = 0;

            object.scale.set(scale, scale, scale);

            var model = object;

            preload.push(object);

        }, onProgress, onError);

    });

}

Код добавления сетки:

function addMeshes(number)

{

        var model = preload[number];

        model.position.x = 130;

        model.position.z = 150;

        var st = {};

        st.model = model.clone();;

        //Создание параллелипипеда и установка его по размерам модели

        //var bbox = new THREE.BoundingBoxHelper( st.model, 0x00ff00 );

        var bbox = new THREE.BoxHelper( st.model, 0x00ff00 );

        bbox.update();

        //Изначально параллелипипед невидим

        bbox.material.visible = false;

        //структура OBB:

        //basis - оси локальной системы координат

        //halfSize - половина размера стороны AABB

        //position - центр AABB

        var obb = {};

        obb.basis = new THREE.Matrix4();

        obb.halfSize = new THREE.Vector3();

        obb.position = new THREE.Vector3();

        var scale, aabb, w;

        //обновление матрицы мира объекта (на всякий случай)

        st.model.updateMatrixWorld();

        //получение данных AABB

        aabb = bbox.box;

        //получение матрицы поворота объекта

        obb.basis.extractRotation( st.model.matrixWorld );

        st.obb = obb;

        bbox.userData = st;

        //Добавление параллелипипеда и модели в сцену

        scene.add(bbox);

        scene.add(bbox.userData.model);

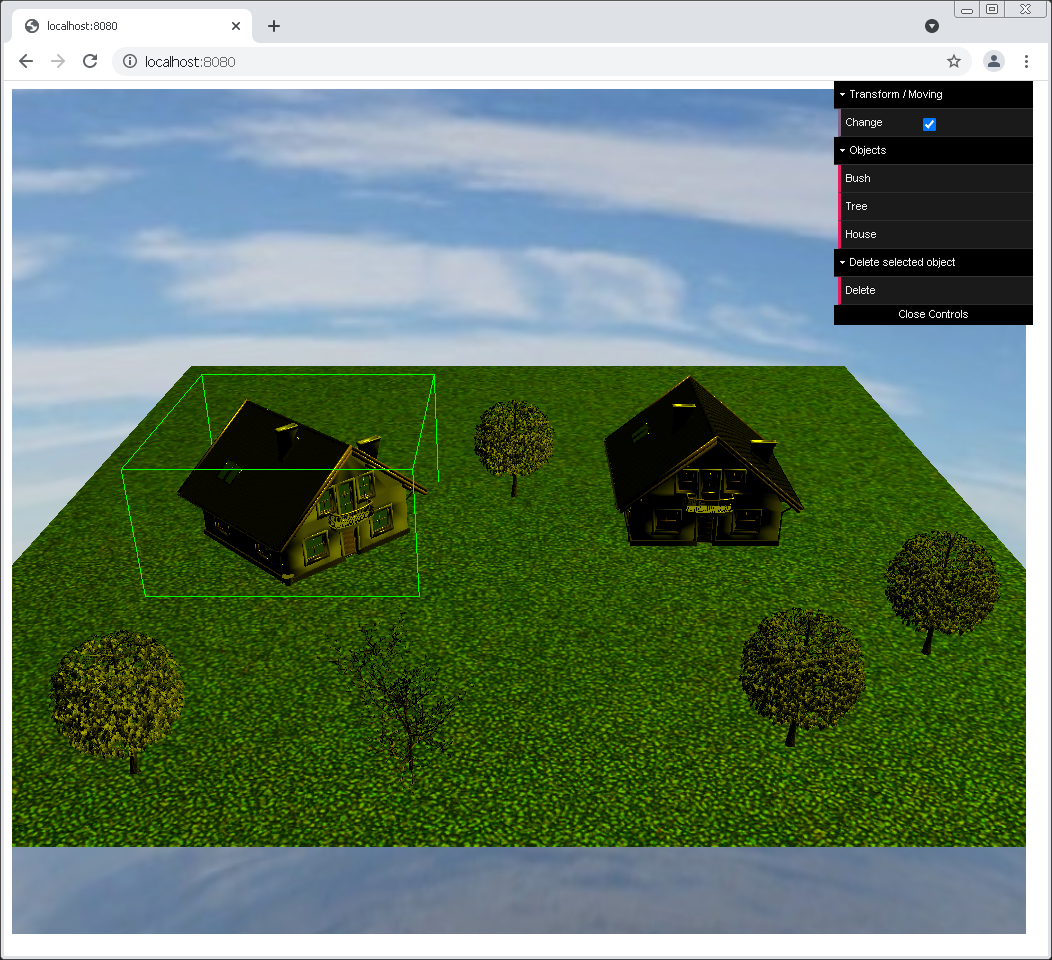
       //Добавление параллелипипеда в список моделей, с которыми будет проверяться пересечение

        targets.push(bbox);

    //}

}

Результат работы:



**Часть 3. Текстуры**

**Задание:** выбор, перемещение и поворот объектов в сцене при помощи мыши

Код выбора объектов:

function onDocumentMouseDown( event )

{

    if (checkbox == false)

    {

        isPressed = true;

        document.onmousedown = function(e){

            switch (e.which) {

                case 1:

                    mousedown = 1;

                    break;

                case 3:

                    mousedown = -1;

                    break;

                default:

                    //console.log(e.which);

            }

        };

    }

    else

    {

        //Если уже выбран какая-то параллелипипед, выделение с него будет снято

        if (picked != null)

        {

            picked.material.visible = false

        }

        //Построение луча через позиции камеры и курсора мыши

        var vector = new THREE.Vector3( mouse.x, mouse.y, 1 );

        vector.unproject(camera);

        var ray = new THREE.Raycaster( camera.position, vector.sub( camera.position ).normalize() );

        //Поиск пересечений луча и параллелипипедов в списке

        var intersects = ray.intersectObjects( targets );

        //Если было найдено пересечение, параллелипипед становится видимым

        if ( intersects.length > 0 )

        {

            picked = intersects[0].object;

            picked.material.visible = true;

        } else

        {

            //Если пересечения не найдены, то текущее выделение сбрасывается

            picked = null;

        }

    }

}

Код перемещения объектов:

function onDocumentMouseMove( event )

{

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

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

    var vector = new THREE.Vector3(mouse.x, mouse.y, 1);

    vector.unproject(camera);

    var ray = new THREE.Raycaster(camera.position, vector.sub(camera.position).normalize());

    var intersects = ray.intersectObjects(targetList);

    if (intersects.length > 0)

    {

        //console.log(intersects[0]);

        cursor.position.copy(intersects[0].point);

        circle.position.copy(intersects[0].point);

        circle.position.y += 1;

        if (picked)

        {

            picked.userData.model.position.copy(intersects[0].point);

            picked.update();

        }

    }

}

Код вращения объектов:

function onDocumentMouseScroll( event )

{

    if (event.wheelDelta > 0)

    {

        if (checkbox == false)

        {

            rad++;

        }

        else

        {

            if (picked)

            {

                picked.update();

                picked.rotation.y += 0.1;

                picked.userData.model.rotation.y += 0.1;

                //обновление матрицы поворота объекта/OBB модели

                picked.userData.obb.basis.extractRotation( picked.userData.model.matrixWorld);

            }

        }

    }

    if (event.wheelDelta < 0)

    {

        if (checkbox == false)

        {

            rad--;

        }

        else

        {

            if (picked)

            {

                picked.update();

                picked.rotation.y -= 0.1;

                picked.userData.model.rotation.y -= 0.1;

                //обновление матрицы поворота объекта/OBB модели

                picked.userData.obb.basis.extractRotation( picked.userData.model.matrixWorld);

            }

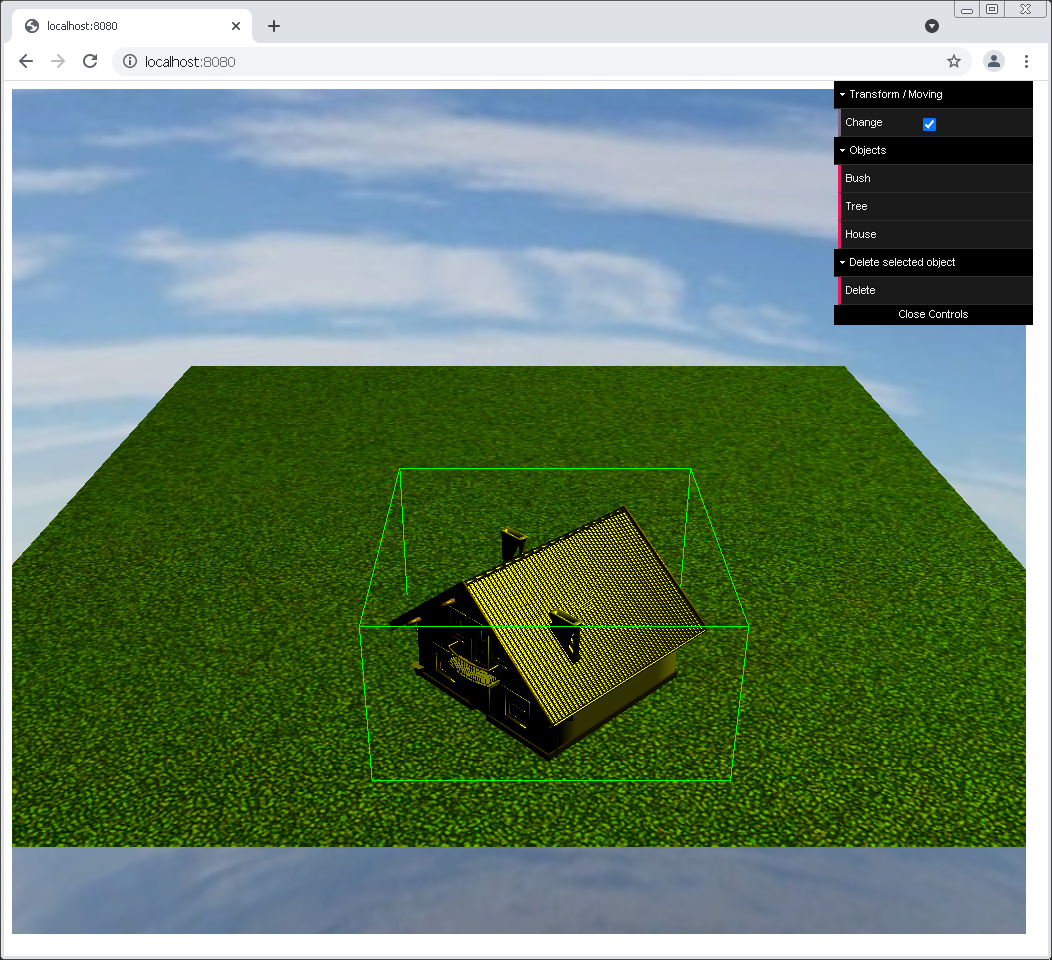
        }

    }

    circle.scale.set(rad, 1, rad);

}

Результат работы:



**Вывод:**

Результатом работы стало веб-приложение на языке JavaScript, осуществляющее построение и визуализацию трёхмерного пространства с возможностью изменения ландшафта, добавления и удаления трёхмерных объектов. Добавленные трёхмерные объекты можно выбирать, перемещать и поворачивать. Присутствует возможность переключения режимов курсора между “кистью” и курсором выбора объектов.