Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий  
Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Отчет

По лабораторной работе №4

«Создание простейшей реалистической сцены средствами OpenGL»

по дисциплине «Компьютерная графика»

Студент группы ПИ-02 Чередов Р.А.

Преподаватель Андреева А.Ю.

Барнаул 2023

Задание: Вариант 8

Стадион  
Код:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>stadium</title>

</head>

<body>

    <style>

        \* {margin:0;padding:0;}

        body {overflow: hidden;}

        .container {width: 100%; height: 100vh;}

    </style>

    <div class="container"></div>

    <script type="importmap">

        {

            "imports": {

                "three": "https://unpkg.com/three@0.139.0/build/three.module.js",

                "OrbitControls": "https://unpkg.com/three@0.139.0/examples/jsm/controls/OrbitControls.js",

                "GLTFLoader": "https://unpkg.com/three@0.139.0/examples/jsm/loaders/GLTFLoader.js",

                "RectAreaLightHelper": "https://unpkg.com/three@0.139.0/examples/jsm/helpers/RectAreaLightHelper.js",

                "RectAreaLightUniformsLib": "https://unpkg.com/three@0.139.0/examples/jsm/lights/RectAreaLightUniformsLib.js"

            }

        }

    </script>

    <script type="module">

        import \* as THREE from 'three';

        import { OrbitControls } from 'OrbitControls';

        import { GLTFLoader } from 'GLTFLoader';

        import { RectAreaLightHelper } from 'RectAreaLightHelper'

        import { RectAreaLightUniformsLib } from 'RectAreaLightUniformsLib';

        function init() {

            let container = document.querySelector('.container');

            //Scene

            const scene = new THREE.Scene()

            scene.background = new THREE.Color("#E2DFE1");

            //Camera

            const camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 3000);

            camera.position.set(0, 0.5, 1)

            //render

            const renderer = new THREE.WebGLRenderer({antialias: true})

            renderer.setSize(window.innerWidth, window.innerHeight)

            container.appendChild(renderer.domElement)

            let plain;

            {

                plain = new THREE.Mesh(

                    new THREE.PlaneGeometry(1000, 1000),

                    new THREE.MeshBasicMaterial({color: "#E2DFE1"})

                )

                plain.reciveShadow = true;

                plain.position.set(0, -1, 0)

                plain.rotateX(-Math.PI / 2);

                scene.add(plain)

            }

            // Model

            const textureLoader = new THREE.TextureLoader();

            const texture = textureLoader.load('./model2/textures/defaultMat\_baseColor.jpeg');

            const sphereMaterial = new THREE.MeshBasicMaterial({ map: texture });

            const sphereGeometry = new THREE.SphereGeometry(0.5, 20, 20);

            const sphereMesh = new THREE.Mesh(sphereGeometry, sphereMaterial);

            sphereMesh.position.set(2, 0.5, 0);

            scene.add(sphereMesh);

            {

                const loader = new GLTFLoader();

                loader.load('./model1/scene.gltf', gltf => {

                scene.add(gltf.scene);

                },

                    function (error) {

                        console.log('Error: ' + error)

                    }

                )

            }

            {

                const loader = new GLTFLoader();

                loader.load('./model2/scene.gltf', gltf => {

                    const object = gltf.scene;

                    object.scale.set(0.1, 0.1, 0.1); // Set the scale to 0.5 in all three dimensions

                    object.position.set(0, 0.125, 0);

                    scene.add(object);

                },

                    function (error) {

                        console.log('Error: ' + error)

                    }

                )

            }

            {

                const light = new THREE.DirectionalLight(0xffffff, 1)

                light.position.set(-2, 0, 10)

                light.lookAt(0, -1, 0)

                scene.add(light)

            }

            {

                const light = new THREE.DirectionalLight(0xffffff, 1)

                light.position.set(-2, 0, 10)

                light.lookAt(0, 1, 0)

                scene.add(light)

            }

            {

                const light = new THREE.DirectionalLight(0xffffff, 1)

                light.position.set(2, 0, 5)

                light.lookAt(0, 1, 0)

                scene.add(light)

            }

            RectAreaLightUniformsLib.init();

            {

                const rectLight = new THREE.RectAreaLight(0xffffff, 1, 100, 100);

                rectLight.position.set(-10,0,0)

                rectLight.rotation.y = Math.PI + Math.PI/4;

                scene.add(rectLight)

            }

            {

                const rectLight = new THREE.RectAreaLight(0xffffff, 1, 100, 100);

                rectLight.position.set(10,0,0)

                rectLight.rotation.y = Math.PI - Math.PI/4;

                scene.add(rectLight)

            }

            {

                const rectLight = new THREE.RectAreaLight(0xffffff, 1, 200, 200);

                rectLight.position.set(1,10,0)

                rectLight.rotation.y = Math.PI - Math.PI/4;

                scene.add(rectLight)

            }

            //OrbitControls

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

            controls.enableDamping = true;

            //Resize

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

            function onWindowResize() {

                camera.aspect = window.innerWidth / window.innerHeight;

                camera.updateProjectionMatrix();

                renderer.setSize(window.innerWidth, window.innerHeight)

            }

            // Animate

            function animate() {

                requestAnimationFrame(animate)

                controls.update();

                renderer.render(scene, camera)

            }

            animate()

        }

        init()

    </script>

</body>

</html>

