Devin Hardy

00076619

CS423



<!DOCTYPE html>

<HTML>

<HEAD>

<TITLE>10.01 Basic Texture</TITLE>

<SCRIPT TYPE="text/javascript" SRC="../libs/three.js"></SCRIPT>

<Script TYPE="text/javascript" SRC="../libs/stats.min.js"></Script>

<STYLE>

body {

margin: 0;

overflow: hidden;

}

</STYLE>

</HEAD>

<BODY>

<DIV id="Stats-output"></DIV>

<DIV ID="WebGL-output">

</DIV>

<!-- Scripts that we use for running things -->

<SCRIPT TYPE="text/javascript" Src="./04-basic-texture.js">

</SCRIPT>

</BODY>

</HTML>

//

// File: 04-basic-texture.js

// Purpose:

// Demo some of the basics of working wit the scenegraph.

// Thisis an extension of code from the Learning Three.js textbook

// once everything is loadded, we run our Three.js stuff

function init() {

var stats = initStats();

var scene = new THREE.Scene();

var camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 0.1, 1000);

var webGLRenderer = new THREE.WebGLRenderer();

webGLRenderer.setClearColor(new THREE.Color(0xEEEEEE, 1.0));

webGLRenderer.setSize(window.innerWidth, window.innerHeight);

webGLRenderer.shadowMapEnabled = true;

camera.position.x = 00;

camera.position.y = 12;

camera.position.z = 28;

camera.lookAt(new THREE.Vector3(0, 0, 0));

var ambiLight = new THREE.AmbientLight(0x141414);

scene.add(ambiLight);

var light= new THREE.DirectionalLight();

light.position.set(0, 30, 20);

scene.add(light);

var polyhedron = createMesh(new THREE.IcosahedronGeometry(5, 0), "metal-rust.jpg");

polyhedron.position.x = 12;

scene.add(polyhedron);

var sphere = createMesh(new THREE.SphereGeometry(5, 20, 20), "floor-wood.jpg");

scene.add(sphere);

var cube = createMesh(new THREE.BoxGeometry(5, 5, 5), "brick-wall.jpg");

cube.position.x = -12;

scene.add(cube);

document.getElementById("WebGL-output").appendChild(webGLRenderer.domElement);

var step = 0;

render();

function createMesh(geom, imageFile) {

var texLoader = new THREE.TextureLoader();

var texture = texLoader.load("../assets/textures/" + imageFile);

var mat = new THREE.MeshPhongMaterial();

mat.map = texture;

var mesh = new THREE.Mesh(geom, mat);

return mesh;

}

function initStats() {

var stats = new Stats();

stats.setMode(0);

stats.domElement.style.position = 'absolute';

stats.domElement.style.left = '0px';

stats.domElement.style.top = '0px';

document.getElementById("Stats-output").appendChild(stats.domElement);

return stats;

}

function render() {

stats.update();

polyhedron.rotation.y = step += 0.01;

polyhedron.rotation.x = step;

cube.rotation.y = step;

cube.rotation.x = step;

sphere.rotation.y = step;

sphere.rotation.x = step;

requestAnimationFrame(render);

webGLRenderer.render(scene, camera);

}

}

window.onload = init;