Damon Bruccoleri Computer Graphics CISD-0792 Dr. Laszlo Instructor

Assignment 1 Date: 1/19/2015

[**Problem ST1 Blue Square (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=ST1.js)

function square(){

var geom = new THREE.Geometry();

geom.vertices.push(new THREE.Vector3(1, 1, 1));

geom.vertices.push(new THREE.Vector3(1, -1, 1));

geom.vertices.push(new THREE.Vector3(-1, -1, 1));

geom.vertices.push(new THREE.Vector3(-1, 1, 1));

var face = new THREE.Face3(1, 0, 2);

geom.faces.push(face);

var face = new THREE.Face3(3, 2, 0);

geom.faces.push(face);

return geom;

}

function createScene() {

var mat = new THREE.MeshBasicMaterial( {color: 'blue', side: THREE.DoubleSide })

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

scene.add(mesh);

}

[**Problem ST2 Open Cube with Lighting (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=ST2.js)

function createCube() {

var geom = new THREE.Geometry();

for (var x = -1; x < 2; x+=2)

for (var y = -1; y < 2; y+=2)

for (var z = -1; z < 2; z+=2)

geom.vertices.push(new THREE.Vector3(x, y, z));

var faces = [[0, 6, 4], [6, 0, 2],// back

[1, 7, 3], [7, 1, 5], // front

[5, 6, 7], [6, 5, 4],// right

[1, 2, 0], [2, 1, 3], // left

[5, 0, 4], [0, 5, 1] ];// bottom

for (var i = 0; i < 10; i++)

geom.faces.push(new THREE.Face3(faces[i][0], faces[i][1], faces[i][2]) );

geom.computeFaceNormals();

mat = new THREE.MeshLambertMaterial({ color: "red" , side: THREE.DoubleSide, overdraw: true });

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

return mesh;

}

function createScene() {

var mesh = createCube();

var axes = new THREE.AxisHelper( 20 );

scene.add(axes);

// add subtle ambient lighting

var ambientLight = new THREE.AmbientLight(0x1f1F1F);

scene.add(ambientLight);

// directional lighting from camera

var directionalLight = new THREE.DirectionalLight(0x7fffff,1);

directionalLight.position = camera.position;

scene.add(directionalLight);

// directional lighting

var directionalLight2 = new THREE.DirectionalLight(0x7fffff,1);

directionalLight2.position.set(0, -10, 1).normalize();

scene.add(directionalLight2);

scene.add(mesh);

}

[**Problem ST3 N sided Polygon with gradient coloring function (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=ST3.js)

function regularPolygonGeometry(n, innerColor, outerColor){

var geom = new THREE.Geometry();

const inc = 2.0\*Math.PI/n;

const r = 2.0;

var innercolor\_obj = new THREE.Color(innerColor);

var outercolor\_obj = new THREE.Color(outerColor);

geom.vertices.push(new THREE.Vector3(0, 0, 1)); // make a center point

geom.vertices.push(new THREE.Vector3(0,r,1)); // push first edge

for(var i=1, nxt=inc ; i<n ; i++, nxt+=inc){

geom.vertices.push(new THREE.Vector3(r\*Math.sin(nxt), r\*Math.cos(nxt), 1));

var face = new THREE.Face3(i+1, i, 0);

face.vertexColors.push(outercolor\_obj);

face.vertexColors.push(outercolor\_obj);

face.vertexColors.push(innercolor\_obj);

geom.faces.push(face);

}

var face = new THREE.Face3(n, 1, 0); // close it up

face.vertexColors.push(outercolor\_obj);

face.vertexColors.push(outercolor\_obj);

face.vertexColors.push(innercolor\_obj);

geom.faces.push(face);

return geom;

}

function createScene() {

var mat = new THREE.MeshBasicMaterial(

{vertexColors: THREE.VertexColors, side: THREE.DoubleSide })

// mesh

var mesh = new THREE.Mesh(regularPolygonGeometry(8,'red','blue'), mat);

scene.add(mesh);

}

[**Problem OT1 Create Startburst of line segments function (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=OT1.js)

function starburst(n, innercolor, outercolor){

const r = 2.0;

const PI2 = 2.0\*Math.PI;

var geom = new THREE.Geometry();

var origin = new THREE.Vector3(0, 0, 0);

innercolor\_obj = new THREE.Color(innercolor);

outercolor\_obj = new THREE.Color(outercolor);

for(i=0; i<n; i++){

var theta = Math.random()\*PI2;

var psi = Math.random()\*PI2;

geom.vertices.push(origin);

geom.vertices.push(new THREE.Vector3(r\*Math.cos(theta)\*Math.sin(psi),

r\*Math.sin(theta)\*Math.sin(psi),

r\*Math.cos(psi)));

geom.colors.push(innercolor\_obj);

geom.colors.push(outercolor\_obj);

}

var mat = new THREE.LineBasicMaterial({vertexColors: true, linewidth: 20});

var line = new THREE.Line(geom, mat, THREE.LinePieces);

return line;

}

function createScene() {

scene.add(starburst(200,'red','green'));

}

[**Problem OT2 Create N sided cylinder of lines (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=OT2.js)

function ruledCylinder(n, lcolor){

const r = 2.0;

const inc = 2.0\*Math.PI/n;

var mat = new THREE.LineBasicMaterial({ color: lcolor, linewidth: 20});

var basegeom = new THREE.Geometry();

var topgeom = new THREE.Geometry();

var sidegeom = new THREE.Geometry();

for(var i=0, nxt=0.0 ; i<n; i++, nxt+=inc){

var tpt = new THREE.Vector3(r\*Math.sin(nxt), 2.0, r\*Math.cos(nxt));

var bpt = tpt.clone();

bpt.y = -2.0;

basegeom.vertices.push(bpt);

topgeom.vertices.push(tpt);

sidegeom.vertices.push(bpt);

sidegeom.vertices.push(tpt);

}

basegeom.vertices[n] = basegeom.vertices[0]; //close base

topgeom.vertices[n] = topgeom.vertices[0]; //close top

var lines = new THREE.Line(basegeom, mat, THREE.LineStrip); // bottom

lines.add( new THREE.Line(topgeom,mat, THREE.LineStrip)); //top

lines.add( new THREE.Line(sidegeom, mat, THREE.LinePieces)); //sides

return lines;

}

function createScene() {

// triangle geometry

scene.add(ruledCylinder(10,'red'));

}