Damon Bruccoleri Computer Graphics CISD-0792 Dr. Laszlo Instructor

Assignment 2 Date: 2/16/2015

[**Problem HGG1 Stairway (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=HGG1.js)

function createSteps( geom, riser, tread, width, nbrSteps) {

// objects are passed by ref

if(nbrSteps){

// create steps below this one

createSteps(geom, riser, tread, width, nbrSteps-1);

}else{

// when we get to bottom we need two vertices.

geom.vertices.push(new THREE.Vector3(tread\*nbrSteps, riser\*nbrSteps, width));

geom.vertices.push(new THREE.Vector3(tread\*nbrSteps, riser\*nbrSteps, 0));

}

// create a single step

var inx = geom.vertices.length; //where are we ?

// use the two vertices of the step below this one, plus add two more for tread

geom.vertices.push(new THREE.Vector3(tread\*nbrSteps, riser\*(nbrSteps+1), width));

geom.vertices.push(new THREE.Vector3(tread\*nbrSteps, riser\*(nbrSteps+1), 0));

geom.faces.push( new THREE.Face3( inx-1, inx-2, inx));

geom.faces.push( new THREE.Face3( inx-1, inx, inx+1));

geom.vertices.push(new THREE.Vector3(tread\*(nbrSteps+1), riser\*(nbrSteps+1), width));

geom.vertices.push(new THREE.Vector3(tread\*(nbrSteps+1), riser\*(nbrSteps+1), 0));

geom.faces.push( new THREE.Face3(inx+1, inx, inx+2));

geom.faces.push( new THREE.Face3(inx+1, inx+2, inx+3));

}

function createStairs(riser, tread, width, nbrSteps){

var geom = new THREE.Geometry();

createSteps( geom, riser, tread, width, nbrSteps-1);

geom.computeFaceNormals();

mat = new THREE.MeshLambertMaterial(

{ color: "red" , side: THREE.DoubleSide, overdraw: true });

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

return mesh;

}

[**Problem HGG2 Cylinder (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=HGG2.js)

function createCylinder(n, len, rad, isCappedBottom, isCappedTop){

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

var geom = new THREE.Geometry();

// push two vertices to form first side edge

geom.vertices.push( new THREE.Vector3(0, len/2.0, rad));

geom.vertices.push( new THREE.Vector3(0, -len/2.0, rad));

for(var i=2, nxt=inc ; i < 2\*n ;i+=2, nxt+=inc){

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

var bpt = tpt.clone();

bpt.y = -len/2.0;

geom.vertices.push(tpt);

geom.vertices.push(bpt);

// two triangles for side from this edge and previous edge

geom.faces.push( new THREE.Face3( i-2, i-1, i));

geom.faces.push( new THREE.Face3( i-1, i+1, i));

}

// close cylinder

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

geom.faces.push( new THREE.Face3( 1,i-2,i-1));

if( isCappedTop )

for (i = 0; i<n-2 ; i++)

geom.faces.push( new THREE.Face3( 0, 2\*i+2, 2\*i+4));

if( isCappedBottom )

for (i = 0; i<n-2 ; i++)

geom.faces.push( new THREE.Face3( 2\*i+5, 2\*i+3, 1));

geom.computeFaceNormals();

return geom;

}

[**Problem HGG3 Segmented Cylinder (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=HGG3.js)

function divColor (index, max){

var freq = 2\*Math.PI\*index/max;

var phase120 = 2\*Math.PI/3; // 120 degrees

var phase240 = 4\*Math.PI/3; // 240 degrees

var red = Math.cos(freq ) \* 127 + 128;

var green = Math.cos(freq + phase120) \* 127 + 128;

var blue = Math.cos(freq + phase240) \* 127 + 128;

return 'rgb(' + Math.round(red) + ',' + Math.round(green) + ',' + Math.round(blue) + ')';

}

function createSegmentedCylinder(n, i, segmentLen, rad, isCappedBottom, isCappedTop){

var mat = new THREE.MeshLambertMaterial({ color: divColor(i,nbrSegments), side: THREE.DoubleSide, overdraw: true });

// Call the createCylinder function from HGG2

var geom = createCylinder(n, segmentLen, rad, (i==nbrSegments) ? isCappedBottom : false, (i==1) ? isCappedTop : false);

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

if(i>1){

var mesh = createSegmentedCylinder(n, i-1, segmentLen, rad, isCappedBottom, isCappedTop);

mesh.position.y = segmentLen; //shift whole stack of children 1 segment up

parentmesh.add(mesh);

}

return parentmesh;

}

function createScene() {

nbrSegments = 15; // note must be global

var n = 12;

scene.add(createSegmentedCylinder(n, nbrSegments, 0.5, 2, true, true));

…

}

[**Problem HGG4 Knotty Cylinder (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=HGG4.js)

function createKnottedCylinder(n, heights, scales, isCappedBottom, isCappedTop){

const nbrSegments = heights.length-1;

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

var mesh = new THREE.Mesh();

var k=0;

var geom = new THREE.Geometry();

var mat = new THREE.MeshFaceMaterial();

var face\_ndx = 0;

for( k=0, nxt=0.0; k<n; k++, nxt+=inc)// push base vertices

geom.vertices.push(new THREE.Vector3(scales[0]\*Math.sin(nxt), heights[0], scales[0]\*Math.cos(nxt)));

for(var j=0, k=n; j<nbrSegments ; j++, k+=n){ // this iterates the segments

// use divColor function from HGG3

mat.materials.push(new THREE.MeshLambertMaterial({color: divColor(j,nbrSegments), side: THREE.DoubleSide}));

// push vertices to form first side edge of a segment

geom.vertices.push( new THREE.Vector3(0, heights[j+1], scales[j+1]));

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

geom.vertices.push(new THREE.Vector3(scales[j+1]\*Math.sin(nxt), heights[j+1], scales[j+1]\*Math.cos(nxt)));

// two triangles for side from this edge and previous edge

geom.faces.push( new THREE.Face3(k+ i, k+i-1, k+i-n-1));

geom.faces[face\_ndx++].materialIndex = j;

geom.faces.push( new THREE.Face3( k+i, k+i-n-1, k+i-n));

geom.faces[face\_ndx++].materialIndex = j;

}

// close cylinder segment

geom.faces.push( new THREE.Face3( k, k+i-1, k-1));

geom.faces[face\_ndx++].materialIndex = j;

geom.faces.push( new THREE.Face3( k, k-1, k-n));

geom.faces[face\_ndx++].materialIndex = j;

}

if( isCappedBottom )

for (var i = 0; i<n-2 ; i++){

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

geom.faces[face\_ndx++].materialIndex = 0;

}

if( isCappedTop )

for (var i = n\*nbrSegments; i< n\*(nbrSegments+1)-2 ; i++){

geom.faces.push( new THREE.Face3( n\*nbrSegments, i+1, i+2));

geom.faces[face\_ndx++].materialIndex = nbrSegments-1;

}

geom.computeFaceNormals();

mesh.add( new THREE.Mesh(geom,mat));

return mesh;

}

[**Problem HGG5 Torus (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=HGG5.js)

function createTorus(n, nbrKnots, outerRadius, innerRadius, startAngle, endAngle){

var heights = [];

var scales = [];

const angleInc = (endAngle-startAngle)/(nbrKnots-1);

for (var i = 0, angle=startAngle; i<nbrKnots ; i++, angle += angleInc){

heights[i] = innerRadius \* Math.sin( angle);

scales[i] = outerRadius + innerRadius \* Math.cos( angle);

}

// use createKnottedCylinder function from HGG4

return createKnottedCylinder(n, heights, scales);

}

function createScene() {

var n = 20;

var nbrKnots = 24;

var startAngle = Math.PI / 2;

var endAngle = 3 \* Math.PI / 2;

var outerRadius = 8;

var innerRadius = 2;

var geom = createTorus(n, nbrKnots, outerRadius, innerRadius, startAngle, endAngle);

…