Damon Bruccoleri Computer Graphics CISD-0792 Dr. Laszlo Instructor

Assignment 3 Date: 2/21/2015

[**Problem SG1 Torus Explorer (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=SG1.js)

function Controls() {

this.radius = 4.5;

this.tube = 4.0;

this.radialsegments = 8;

this.tubularsegments = 6;

this.arc = 360;

this.go = function() {

if(currentMesh)

scene.remove(currentMesh);

var geom = new THREE.TorusGeometry(this.radius, this.tube, this.radialsegments,

this.tubularsegments, this.arc\*2\*Math.PI/360);

currentMesh = new THREE.Mesh(geom, currentMat);

scene.add(currentMesh);

};

}

function initGui() {

gui = new dat.GUI();

controls = new Controls();

gui.add(controls, 'radius', 0.1, 25).step(0.1);

gui.add(controls, 'tube',1, 10).step( 0.1 );

gui.add(controls, 'radialsegments', 4, 40).step(1);

gui.add(controls, 'tubularsegments', 6, 40).step(1);

gui.add(controls,'arc', 0, 360).step(5);

gui.add(controls,'go');

}

function createScene() {

currentMat = new THREE.MeshLambertMaterial({color: "blue", shading: THREE.FlatShading});

var geom = new THREE.TorusGeometry(controls.radius,controls.tube,

controls.radialsegments,controls.tubularsegments);

currentMesh = new THREE.Mesh(geom, currentMat);

scene.add(currentMesh);

…

[**Problem MC1 Matrix of Boxes (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=MC1.js)

function ExpandObjectDimension( obj, n){

var newobj = obj.clone();

if (n>1)

newobj.add( ExpandObjectDimension(obj, n-1));

return newobj;

}

// M x N x O matrix of boxes centered in the xyz-space

function createCube3DMatrix(m, n, o, offset) {

var offset = (offset !== undefined) ? offset : 2.0;

var geom = new THREE.CubeGeometry(1, 1, 1);

var mat = new THREE.MeshLambertMaterial({color:'red'});

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

protoMesh.position.set(offset,0,0);

protoMesh = ExpandObjectDimension(protoMesh, m);

protoMesh.position.set(0,offset,0);

protoMesh = ExpandObjectDimension(protoMesh, n);

protoMesh.position.set(0,0, offset);

protoMesh = ExpandObjectDimension(protoMesh, o);

var xMin = -offset \* ((m-1) / 2.0);

var yMin = -offset \* ((n-1) / 2.0);

var zMin = -offset \* ((o-1) / 2.0);

protoMesh.position.set( xMin, yMin, zMin);

scene.add(protoMesh);

}

function createScene(){

createCube3DMatrix(9,9,9);

…

[**Problem MC2 Ziggurat (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=MC2.js)

function ziggurat(n, zheight, sf){

var mat = new THREE.MeshLambertMaterial({ side: THREE.FrontSide });

mat.color = new THREE.Color().setHSL( n/nbrSegments, 1, .5);

if (typeof geom === "undefined") geom = new THREE.BoxGeometry(2, zheight, 2);

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

if(n>1){

var mesh = ziggurat(n-1, zheight, sf);

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

mesh.scale.set(sf,1,sf); //scale whole stack of children

parentmesh.add(mesh); // tack on base

}

return parentmesh;

}

function createScene() {

nbrSegments = 30; //global

scene.add(ziggurat(nbrSegments, 0.2, 0.9));

[**Problem MC3 Zigurrat Explorer (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=MC3.js)

function Controls() {

this.levels = 30;

this.height = 0.2;

this.scale = 0.9;

this.create = function() {

if(currentmesh)

scene.remove(currentmesh);

geom = new THREE.BoxGeometry(2, this.height, 2);

currentmesh = ziggurat(this.levels, this.height, this.scale);

scene.add(currentmesh);

}

}

function initGui() {

gui = new dat.GUI();

controls = new Controls();

gui.add(controls, 'levels', 1, 50).step(1);

gui.add(controls, 'height', 0.1, 1).step(0.1);

gui.add(controls, 'scale', 0.5, 0.99).step(0.01);

gui.add(controls,'create');

}

function ziggurat(n, zheight, sf){

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

mat.color = new THREE.Color().setHSL(n/controls.levels,1,0.5);

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

if(n>1){

var mesh = ziggurat(n-1, zheight, sf);

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

mesh.scale.set(sf,1,sf); //scale whole stack of children

parentmesh.add(mesh);

}

return parentmesh;

}

function createScene() {

geom = new THREE.BoxGeometry(2, controls.height, 2);

currentmesh = ziggurat(controls.levels, controls.height, controls.scale);

scene.add(currentmesh);

...

[**Problem MC4 Helix (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=MC4.js)

function createHelix(object, n, radius, angle, dist){

var helix = new THREE.Object3D();

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

if(object instanceof Function)

var child2= object(i);

else

var child2 = object.clone();

child2.position.z = dist\*i;

child2.position.x = radius;

var child = new THREE.Object3D();

child.add(child2); // force rotation to happen after translation

child.rotation.z = angle\*i;

helix.add(child);

}

return helix;

}

function createScene() {

var mat = new THREE.MeshLambertMaterial({color: 'blue'});

var geom = new THREE.SphereGeometry(1, 12, 12);

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

var helix = createHelix(mesh, 49, 2, Math.PI / 4, 0.5);

scene.add(helix);

…

[**Problem MC5 Helix of Objects (ctrl-click here to load)**](http://damon4.com/Computer%20Graphics%20HW/hw_shell.html?load=MC5.js)

function generateKnots(max){

var geom = new THREE.TorusKnotGeometry(5, 2);

return function(n){

var mat = new THREE.MeshLambertMaterial();//{color: 'hsl( ' +Math.round(n/max)+ ',1,.5)'});

mat.color = new THREE.Color().setHSL(n/max, 1, .5);

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

mesh.scale = new THREE.Vector3(0.1, 0.1, 0.1);

return mesh;

}

}

// this is a second way to make the helix. See MC4 for the other way I used.

function createHelix(object, n, radius, angle, dist){

var helix = new THREE.Object3D();

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

if(object instanceof Function)

var child= object(i);

else

var child = object.clone();

child.position.z = dist\*i; // do this a little different than MCC4

child.position.x = radius\*Math.cos(inc);

child.position.y = radius\*Math.sin(inc);

child.rotation.z = inc; //this rotation is optional

// it keeps the same side facing in

helix.add( child );

}

return helix;

}

function createScene() {

var knotGen = generateKnots(49);

var helix = createHelix(knotGen, 49, 2, Math.PI / 4, 0.5);

scene.add(helix);