**4.10 開口圓錐**

var Debug = Core.Debug;

var Mesh3D = Core.Mesh3D;

var Path2D = Core.Path2D;

var Plugin = Core.Plugin;

var Tess = Core.Tess;

var Solid = Core.Solid;

params = [

{ "id": "r1", "displayName": "Base Radius", "type": "length", "rangeMin": 0, "rangeMax": 100, "default": 10.0 },

{ "id": "r2", "displayName": "Top Radius", "type": "length", "rangeMin": 0, "rangeMax": 100, "default": 10.0 },

{ "id": "height", "displayName": "Height", "type": "length", "rangeMin": 1.0, "rangeMax": 100.0, "default": 20.0 },

{ "id": "arc", "displayName": "Arc", "type": "angle", "rangeMin": 0, "rangeMax": 360.0, "default": 270.0 },

{ "id": "x\_pos","displayName": "x position","type": "float","rangeMin": -100,"rangeMax": 100,"default": 0},

{ "id": "y\_pos","displayName": "y position","type": "float","rangeMin": -100,"rangeMax": 100,"default": 0 },

{ "id": "z\_pos","displayName": "z position","type": "float","rangeMin": -100,"rangeMax": 100,"default": 0},

{ "id": "times", "displayName": "precision", "type": "int", "rangeMin": 3, "rangeMax": 200, "default": 54}

];

function process(params) {

var x\_pos = params.x\_pos;

var y\_pos = params.y\_pos;

var z\_pos = params.z\_pos;

var times = params["times"];

var height = params["height"];

var arc = params ["arc"];

var r1 = params["r1"];

var r2 = params["r2"];

var cl = [x\_pos+0,y\_pos+0,z\_pos+0]; //中心低點(起始點)   
 var ch = [x\_pos+0,y\_pos+0,z\_pos+height]; //中心高點(起始點)

var pl = [x\_pos+r1,y\_pos+0,z\_pos+0]; //外圈低點(起始點)

var ph = [x\_pos+r2,y\_pos+0,z\_pos+height]; //外圈高點(起始點)

var mesh = new Mesh3D();

// open cap

if (arc < 360) {

mesh.triangle(cl, pl, ch); //起始側邊三角形1

mesh.triangle(cl, ph, ch); //起始側邊三角形2

}

var ndivs=times;

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

var a = (i+1)/ndivs \* arc / 180 \* Math.PI;

var s = Math.sin(a);

var c = Math.cos(a);

var nl = [x\_pos+r1\*c, y\_pos-r1\*s,z\_pos+ 0]; //外圈低點(掃描點)

var nh = [x\_pos+r2\*c, y\_pos-r2\*s, z\_pos+height]; //外圈高點(掃描點)

mesh.triangle(pl, ph, nl); //外部三角形1

mesh.triangle(nl, ph, nh); //外部三角形2

mesh.triangle(cl, pl, nl); //底部三角形

mesh.triangle(ch, nh, ph); //頂部三角形

pl = nl;

ph = nh;

}

// open cap

if (arc < 360) {

mesh.triangle(cl, ph, ch); //結束側邊三角形1

mesh.triangle(cl, ph, pl); //結束側邊三角形2

}

var solid = Solid.make(mesh);

return solid;

}

presets = [

{ 'color': [233, 29, 45] }

**];**