**4.8 可變多邊角柱**

var Debug = Core.Debug;

var Mesh3D = Core.Mesh3D;

var Path2D = Core.Path2D;

var Plugin = Core.Plugin;

var Tess = Core.Tess;

var Solid = Core.Solid;

// -------------------------------------------

// Original Shape Script:

// -------------------------------------------

params = [

{ "id": "topradius","displayName": "TopRadius","type": "length", "rangeMin": 1, "rangeMax": 100,"default": 10},

{ "id": "bottomradius","displayName": "BottomRadius", "type": "length", "rangeMin": 1, "rangeMax": 100, "default": 20 },

{ "id": "height","displayName": "Height","type": "length", "rangeMin": 1,"rangeMax": 100, "default": 20 },

{ "id": "sides","displayName": "NumOfSides", "type": "int", "rangeMin": 1,"rangeMax": 30, "default": 6 },

{ "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}

];

function process(params) {

var x\_pos = params.x\_pos;

var y\_pos = params.y\_pos;

var z\_pos = params.z\_pos;

var tr = params.topradius;

var br = params.bottomradius;

var height = params.height;

var sides = params.sides;

var angle = 2\*Math.PI / sides;

var mesh = new Mesh3D();

var top = [];   
 var bottom = [];

var base = [x\_pos+0,y\_pos+0,z\_pos+0];

var peak = [x\_pos+0, y\_pos+0, z\_pos+height];

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

{

var tx = tr \* Math.cos(i \* angle);

var ty = tr \* Math.sin(i \* angle);

var bx = br \* Math.cos(i \* angle);

var by = br \* Math.sin(i \* angle);

bottom.push([x\_pos+bx, y\_pos+by, z\_pos+0]);

top.push([x\_pos+tx, y\_pos+ty, z\_pos+height]);

}

for (var i = 0; i < sides-1; i++)

{

mesh.triangle (peak, top[i], top[i+1]);

mesh.quad(bottom[i], bottom[i+1],top[i+1],top[i]);

mesh.triangle (base, bottom[i+1], bottom[i]);

}

mesh.triangle (peak, top[sides-1], top[0]);

mesh.quad(bottom[sides-1], bottom[0],top[0],top[sides-1]);

mesh.triangle (base, bottom[0], bottom[sides-1]);

return Solid.make(mesh);

}