**4.15 9點可變旋轉體**

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": "x9\_pos","displayName": "x9 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 30},

{ "id": "z9\_pos","displayName": "z9 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 90 },

{ "id": "x8\_pos","displayName": "x8 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 50},

{ "id": "z8\_pos","displayName": "z8 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 80 },

{ "id": "x7\_pos","displayName": "x7 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 30},

{ "id": "z7\_pos","displayName": "z7 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 70 },

{ "id": "x6\_pos","displayName": "x6 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 20},

{ "id": "z6\_pos","displayName": "z6 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 60 },

{ "id": "x5\_pos","displayName": "x5 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 40},

{ "id": "z5\_pos","displayName": "z5 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 50 },

{ "id": "x4\_pos","displayName": "x4 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 20},

{ "id": "z4\_pos","displayName": "z4 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 40 },

{ "id": "x3\_pos","displayName": "x3 position","type": "float","rangeMin": 0,"rangeMax": 100,"default":20},

{ "id": "z3\_pos","displayName": "z3 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 30},

{ "id": "x2\_pos","displayName": "x2 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 30},

{ "id": "z2\_pos","displayName": "z2 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 20 },

{ "id": "x1\_pos","displayName": "x1 position","type": "float","rangeMin": 0,"rangeMax": 100,"default":10},

{ "id": "z1\_pos","displayName": "z1 position","type": "float","rangeMin": 0,"rangeMax": 100,"default":10 },

{ "id": "height\_top", "displayName": "Height\_top", "type": "length", "rangeMin": 0, "rangeMax": 100, "default": 100 },

{ "id": "height\_bottom", "displayName": "Height\_bottom", "type": "length", "rangeMin": 0, "rangeMax": 100, "default": 0 },

{ "id": "sides", "displayName": "NumOfSides", "type": "int", "rangeMin": 3, "rangeMax": 360, "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 x9\_pos = params.x9\_pos;

var z9\_pos = params.z9\_pos;

var x8\_pos = params.x8\_pos;

var z8\_pos = params.z8\_pos;

var x7\_pos = params.x7\_pos;

var z7\_pos = params.z7\_pos;

var x6\_pos = params.x6\_pos;

var z6\_pos = params.z6\_pos;

var x5\_pos = params.x5\_pos;

var z5\_pos = params.z5\_pos;

var x4\_pos = params.x4\_pos;

var z4\_pos = params.z4\_pos;

var x3\_pos = params.x3\_pos;

var z3\_pos = params.z3\_pos;

var x2\_pos = params.x2\_pos;

var z2\_pos = params.z2\_pos;

var x1\_pos = params.x1\_pos;

var z1\_pos = params.z1\_pos;

var x\_pos = params.x\_pos;

var y\_pos = params.y\_pos;

var z\_pos = params.z\_pos;

var tr9 = x9\_pos;

var tr8 = x8\_pos;

var tr7 = x7\_pos;

var tr6 = x6\_pos;

var tr5 = x5\_pos;

var tr4 = x4\_pos;

var tr3 = x3\_pos;

var tr2 = x2\_pos;

var tr1 = x1\_pos;

var height\_top = params.height\_top;

var height\_bottom = params.height\_bottom;

var sides = params.sides;

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

var mesh = new Mesh3D();

var side1=[];

var side2=[];

var side3=[];

var side4=[];

var side5=[];

var side6=[];

var side7=[];

var side8=[];

var side9=[];

var top = [];

var bottom = [];

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

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

for (var i = 0; i < sides; i++) //設定各點座標

{

var tx1 = tr1 \* Math.cos(i \* angle);

var ty1 = tr1 \* Math.sin(i \* angle);

var tx2 = tr2 \* Math.cos(i \* angle);

var ty2 = tr2 \* Math.sin(i \* angle);

var tx3 = tr3 \* Math.cos(i \* angle);

var ty3 = tr3 \* Math.sin(i \* angle);

var tx4 = tr4 \* Math.cos(i \* angle);

var ty4 = tr4 \* Math.sin(i \* angle);

var tx5 = tr5 \* Math.cos(i \* angle);

var ty5 = tr5 \* Math.sin(i \* angle);

var tx6 = tr6 \* Math.cos(i \* angle);

var ty6 = tr6 \* Math.sin(i \* angle);

var tx7 = tr7 \* Math.cos(i \* angle);

var ty7 = tr7 \* Math.sin(i \* angle);

var tx8 = tr8 \* Math.cos(i \* angle);

var ty8 = tr8 \* Math.sin(i \* angle);

var tx9 = tr9 \* Math.cos(i \* angle);

var ty9 = tr9 \* Math.sin(i \* angle);

side1.push([x\_pos+tx1, y\_pos+ty1, z\_pos+z1\_pos]);

side2.push([x\_pos+tx2, y\_pos+ty2, z\_pos+z2\_pos]);

side3.push([x\_pos+tx3, y\_pos+ty3, z\_pos+z3\_pos]);

side4.push([x\_pos+tx4, y\_pos+ty4, z\_pos+z4\_pos]);

side5.push([x\_pos+tx5, y\_pos+ty5, z\_pos+z5\_pos]);

side6.push([x\_pos+tx6, y\_pos+ty6, z\_pos+z6\_pos]);

side7.push([x\_pos+tx7, y\_pos+ty7, z\_pos+z7\_pos]);

side8.push([x\_pos+tx8, y\_pos+ty8, z\_pos+z8\_pos]);

side9.push([x\_pos+tx9, y\_pos+ty9, z\_pos+z9\_pos]);

}

for ( i = 0; i < sides-1; i++)   
 //用三角形與四邊形掃邊(不包含最後封口)

{

mesh.triangle (peak, side9[i], side9[i+1]); //頂部三角形

//掃邊

mesh.quad(side1[i], side1[i+1],side2[i+1],side2[i]);

mesh.quad(side2[i], side2[i+1],side3[i+1],side3[i]);

mesh.quad(side3[i], side3[i+1],side4[i+1],side4[i]);

mesh.quad(side4[i], side4[i+1],side5[i+1],side5[i]);

mesh.quad(side5[i], side5[i+1],side6[i+1],side6[i]);

mesh.quad(side6[i], side6[i+1],side7[i+1],side7[i]);

mesh.quad(side7[i], side7[i+1],side8[i+1],side8[i]);

mesh.quad(side8[i], side8[i+1],side9[i+1],side9[i]);

mesh.triangle (base, side1[i+1], side1[i]); //底部三角形

}

mesh.triangle (peak, side9[sides-1], side9[0]); //最後頂部封口

mesh.quad(side1[sides-1], side1[0],side2[0],side2[sides-1]);

//最後四邊形掃邊封口

mesh.quad(side2[sides-1], side2[0],side3[0],side3[sides-1]);

mesh.quad(side3[sides-1], side3[0],side4[0],side4[sides-1]);

mesh.quad(side4[sides-1], side4[0],side5[0],side5[sides-1]);

mesh.quad(side5[sides-1], side5[0],side6[0],side6[sides-1]);

mesh.quad(side6[sides-1], side6[0],side7[0],side7[sides-1]);

mesh.quad(side7[sides-1], side7[0],side8[0],side8[sides-1]);

mesh.quad(side8[sides-1], side8[0],side9[0],side9[sides-1]);

mesh.triangle (base, side1[0], side1[sides-1]);

//最後底部封口

return Solid.make(mesh);

}