**4.33 多中空旋轉柱**

var Debug = Core.Debug;

var Mesh3D = Core.Mesh3D;

var Path2D = Core.Path2D;

var Plugin = Core.Plugin;

var Tess = Core.Tess;

var Solid = Core.Solid;

var chosenNames = ["中央多一旋轉柱" ,"中央不要旋轉柱" ];

var chosenValues = ["yes", "no"];

params = [

{ "id": "centerAdd",

"displayName": "中央是否多一旋轉柱",

"type": "list",

"listLabels": chosenNames ,

"listValues": chosenValues,

"default": "no"

},

{ "id": "dd","displayName": "方柱繞z軸分身個數","type": "int", "rangeMin": 1,"rangeMax": 8, "default": 5 },

{ "id": "start\_x","displayName": "起始公轉旋轉體中心x座標","type": "float","rangeMin": 0,"rangeMax": 100,"default":60},

{ "id": "x8\_pos","displayName": "x8 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 10},

{ "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": 12},

{ "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": 14},

{ "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": 18},

{ "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": 16},

{ "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":14},

{ "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": 12},

{ "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":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 centerAdd= params["centerAdd"];

var dd=params.dd;

var start\_x = params.start\_x;

var x9\_pos = params.x1\_pos; //第九層取與第一層同

var z9\_pos = params.z1\_pos; //第九層取與第一層同

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 bx1 ; //宣告柱中心點公轉xy座標

var by1 ;

var sides = params.sides;

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

var mesh = new Mesh3D();

//宣告各層各點座標

var side1=[];

var side2=[];

var side3=[];

var side4=[];

var side5=[];

var side6=[];

var side7=[];

var side8=[];

var side9=[];

for (var j = 0 ; j <dd+1 ;j++) {

side1[j]=[];

side2[j]=[];

side3[j]=[];

side4[j]=[];

side5[j]=[];

side6[j]=[];

side7[j]=[];

side8[j]=[];

side9[j]=[];

}

//暫用座標宣告

var tx1 ;

var ty1;

var tx2 ;

var ty2;

var tx3 ;

var ty3 ;

var tx4 ;

var ty4 ;

var tx5;

var ty5 ;

var tx6;

var ty6 ;

var tx7 ;

var ty7 ;

var tx8 ;

var ty8;

var tx9 ;

var ty9 ;

var anglen=2\*Math.PI/dd; //分身每次旋轉角度

var big\_radius=start\_x; //公轉半徑

var ddAdd; //是否加一選轉柱

if(centerAdd=="yes") ddAdd=1;

else ddAdd=0;

for (var n=0; n< dd+ddAdd; n++) //對所有柱子

{

//以柱中心點公轉xy座標

bx1 = big\_radius\*Math.cos(n\*anglen);

by1 =big\_radius\*Math.sin(n\*anglen);

if (n==dd){

bx1=0;

by1=0;}

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

{

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

side1[n].push([x\_pos+bx1+tx1, y\_pos+by1+ty1, z\_pos+z1\_pos]);

side2[n].push([x\_pos+bx1+tx2, y\_pos+by1+ty2, z\_pos+z2\_pos]);

side3[n].push([x\_pos+bx1+tx3, y\_pos+by1+ty3, z\_pos+z3\_pos]);

side4[n].push([x\_pos+bx1+tx4, y\_pos+by1+ty4, z\_pos+z4\_pos]);

side5[n].push([x\_pos+bx1+tx5, y\_pos+by1+ty5, z\_pos+z5\_pos]);

side6[n].push([x\_pos+bx1+tx6, y\_pos+by1+ty6, z\_pos+z6\_pos]);

side7[n].push([x\_pos+bx1+tx7, y\_pos+by1+ty7, z\_pos+z7\_pos]);

side8[n].push([x\_pos+bx1+tx8, y\_pos+by1+ty8, z\_pos+z8\_pos]);

side9[n].push([x\_pos+bx1+tx9, y\_pos+by1+ty9, z\_pos+z9\_pos]);

}

for ( i = 0; i < sides-1; i++) //用四邊形掃邊(不包含最後封口)

{

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

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

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

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

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

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

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

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

}

//最後四邊形掃邊封口

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

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

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

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

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

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

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

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

}

return Solid.make(mesh);

}