**4.22 藝術造型編織籃**

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": "x4\_pos","displayName": "x4 position","type": "float","rangeMin": 0,"rangeMax": 100,"default": 12},

{ "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":18},

{ "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": 15},

{ "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": "division","displayName": "division number","type": "int","rangeMin": 1,"rangeMax": 16,"default": 3},

{ "id": "cubic\_length" ,"displayName": "length of small cubic","type": "float","rangeMin": 0.1,"rangeMax": 3,"default":2},

{ "id": "sides", "displayName": "NumOfSides", "type": "int", "rangeMin": 4, "rangeMax": 360, "default": 25 },

{ "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 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 tr4 = x4\_pos;

var tr3 = x3\_pos;

var tr2 = x2\_pos;

var tr1 = x1\_pos;

var cubic\_length=params.cubic\_length;

var ch=cubic\_length/2; //取ch為cubic\_length之一半

var division=params.division;

var sides = params.sides;

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

var mesh = new Mesh3D();

var side1a=[];

var side1b=[];

var side1c=[];

var side1d=[];

var side1e=[];

var side1f=[];

var side1g=[];

var side1h=[];

var side2a=[];

var side2b=[];

var side2c=[];

var side2d=[];

var side2e=[];

var side2f=[];

var side2g=[];

var side2h=[];

var side3a=[];

var side3b=[];

var side3c=[];

var side3d=[];

var side3e=[];

var side3f=[];

var side3g=[];

var side3h=[];

var side4a=[];

var side4b=[];

var side4c=[];

var side4d=[];

var side4e=[];

var side4f=[];

var side4g=[];

var side4h=[];

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

{

var angle1=ch/tr1; //取cubic角度增加

var angle2=ch/tr2;

var angle3=ch/tr3;

var angle4=ch/tr4;

//cubic 1 8點x座標

var tx1a = (tr1-ch) \* Math.cos(i \* angle-angle1);

var tx1b = (tr1+ch) \* Math.cos(i \* angle-angle1);

var tx1c = (tr1+ch) \* Math.cos(i \* angle-angle1);

var tx1d = (tr1-ch) \* Math.cos(i \* angle-angle1);

var tx1e = (tr1-ch) \* Math.cos(i \* angle+angle1);

var tx1f = (tr1+ch) \* Math.cos(i \* angle+angle1);

var tx1g = (tr1+ch) \* Math.cos(i \* angle+angle1);

var tx1h = (tr1-ch) \* Math.cos(i \* angle+angle1);

//cubic 1 8點y座標

var ty1a = (tr1-ch) \* Math.sin(i \* angle-angle1);

var ty1b = (tr1+ch) \* Math.sin(i \* angle-angle1);

var ty1c = (tr1+ch) \* Math.sin(i \* angle-angle1);

var ty1d = (tr1-ch) \* Math.sin(i \* angle-angle1);

var ty1e = (tr1-ch) \* Math.sin(i \* angle+angle1);

var ty1f = (tr1+ch) \* Math.sin(i \* angle+angle1);

var ty1g = (tr1+ch) \* Math.sin(i \* angle+angle1);

var ty1h = (tr1-ch) \* Math.sin(i \* angle+angle1);

//cubic 2 8點x座標

var tx2a = (tr2-ch) \* Math.cos(i \* angle-angle2);

var tx2b = (tr2+ch) \* Math.cos(i \* angle-angle2);

var tx2c = (tr2+ch) \* Math.cos(i \* angle-angle2);

var tx2d = (tr2-ch) \* Math.cos(i \* angle-angle2);

var tx2e = (tr2-ch) \* Math.cos(i \* angle+angle2);

var tx2f = (tr2+ch) \* Math.cos(i \* angle+angle2);

var tx2g = (tr2+ch) \* Math.cos(i \* angle+angle2);

var tx2h = (tr2-ch) \* Math.cos(i \* angle+angle2);

//cubic 2 8點y座標

var ty2a = (tr2-ch) \* Math.sin(i \* angle-angle2);

var ty2b = (tr2+ch) \* Math.sin(i \* angle-angle2);

var ty2c = (tr2+ch) \* Math.sin(i \* angle-angle2);

var ty2d = (tr2-ch) \* Math.sin(i \* angle-angle2);

var ty2e = (tr2-ch) \* Math.sin(i \* angle+angle2);

var ty2f = (tr2+ch) \* Math.sin(i \* angle+angle2);

var ty2g = (tr2+ch) \* Math.sin(i \* angle+angle2);

var ty2h = (tr2-ch) \* Math.sin(i \* angle+angle2);

//cubic 3 8點x座標

var tx3a = (tr3-ch) \* Math.cos(i \* angle-angle3);

var tx3b = (tr3+ch) \* Math.cos(i \* angle-angle3);

var tx3c = (tr3+ch) \* Math.cos(i \* angle-angle3);

var tx3d = (tr3-ch) \* Math.cos(i \* angle-angle3);

var tx3e = (tr3-ch) \* Math.cos(i \* angle+angle3);

var tx3f = (tr3+ch) \* Math.cos(i \* angle+angle3);

var tx3g = (tr3+ch) \* Math.cos(i \* angle+angle3);

var tx3h = (tr3-ch) \* Math.cos(i \* angle+angle3);

//cubic 3 8點y座標

var ty3a = (tr3-ch) \* Math.sin(i \* angle-angle3);

var ty3b = (tr3+ch) \* Math.sin(i \* angle-angle3);

var ty3c = (tr3+ch) \* Math.sin(i \* angle-angle3);

var ty3d = (tr3-ch) \* Math.sin(i \* angle-angle3);

var ty3e = (tr3-ch) \* Math.sin(i \* angle+angle3);

var ty3f = (tr3+ch) \* Math.sin(i \* angle+angle3);

var ty3g = (tr3+ch) \* Math.sin(i \* angle+angle3);

var ty3h = (tr3-ch) \* Math.sin(i \* angle+angle3);

//cubic 4 8點x座標

var tx4a = (tr4-ch) \* Math.cos(i \* angle-angle4);

var tx4b = (tr4+ch) \* Math.cos(i \* angle-angle4);

var tx4c = (tr4+ch) \* Math.cos(i \* angle-angle4);

var tx4d = (tr4-ch) \* Math.cos(i \* angle-angle4);

var tx4e = (tr4-ch) \* Math.cos(i \* angle+angle4);

var tx4f = (tr4+ch) \* Math.cos(i \* angle+angle4);

var tx4g = (tr4+ch) \* Math.cos(i \* angle+angle4);

var tx4h = (tr4-ch) \* Math.cos(i \* angle+angle4);

//cubic 4 8點y座標

var ty4a = (tr4-ch) \* Math.sin(i \* angle-angle4);

var ty4b = (tr4+ch) \* Math.sin(i \* angle-angle4);

var ty4c = (tr4+ch) \* Math.sin(i \* angle-angle4);

var ty4d = (tr4-ch) \* Math.sin(i \* angle-angle4);

var ty4e = (tr4-ch) \* Math.sin(i \* angle+angle4);

var ty4f = (tr4+ch) \* Math.sin(i \* angle+angle4);

var ty4g = (tr4+ch) \* Math.sin(i \* angle+angle4);

var ty4h = (tr4-ch) \* Math.sin(i \* angle+angle4);

side1a.push([x\_pos+tx1a, y\_pos+ty1a, z\_pos+z1\_pos-ch]);

side1b.push([x\_pos+tx1b, y\_pos+ty1b, z\_pos+z1\_pos-ch]);

side1c.push([x\_pos+tx1c, y\_pos+ty1c, z\_pos+z1\_pos+ch]);

side1d.push([x\_pos+tx1d, y\_pos+ty1d, z\_pos+z1\_pos+ch]);

side1e.push([x\_pos+tx1e, y\_pos+ty1e, z\_pos+z1\_pos-ch]);

side1f.push([x\_pos+tx1f, y\_pos+ty1f, z\_pos+z1\_pos-ch]);

side1g.push([x\_pos+tx1g, y\_pos+ty1g, z\_pos+z1\_pos+ch]);

side1h.push([x\_pos+tx1h, y\_pos+ty1h, z\_pos+z1\_pos+ch]);

side2a.push([x\_pos+tx2a, y\_pos+ty2a, z\_pos+z2\_pos-ch]);

side2b.push([x\_pos+tx2b, y\_pos+ty2b, z\_pos+z2\_pos-ch]);

side2c.push([x\_pos+tx2c, y\_pos+ty2c, z\_pos+z2\_pos+ch]);

side2d.push([x\_pos+tx2d, y\_pos+ty2d, z\_pos+z2\_pos+ch]);

side2e.push([x\_pos+tx2e, y\_pos+ty2e, z\_pos+z2\_pos-ch]);

side2f.push([x\_pos+tx2f, y\_pos+ty2f, z\_pos+z2\_pos-ch]);

side2g.push([x\_pos+tx2g, y\_pos+ty2g, z\_pos+z2\_pos+ch]);

side2h.push([x\_pos+tx2h, y\_pos+ty2h, z\_pos+z2\_pos+ch]);

side3a.push([x\_pos+tx3a, y\_pos+ty3a, z\_pos+z3\_pos-ch]);

side3b.push([x\_pos+tx3b, y\_pos+ty3b, z\_pos+z3\_pos-ch]);

side3c.push([x\_pos+tx3c, y\_pos+ty3c, z\_pos+z3\_pos+ch]);

side3d.push([x\_pos+tx3d, y\_pos+ty3d, z\_pos+z3\_pos+ch]);

side3e.push([x\_pos+tx3e, y\_pos+ty3e, z\_pos+z3\_pos-ch]);

side3f.push([x\_pos+tx3f, y\_pos+ty3f, z\_pos+z3\_pos-ch]);

side3g.push([x\_pos+tx3g, y\_pos+ty3g, z\_pos+z3\_pos+ch]);

side3h.push([x\_pos+tx3h, y\_pos+ty3h, z\_pos+z3\_pos+ch]);

side4a.push([x\_pos+tx4a, y\_pos+ty4a, z\_pos+z4\_pos-ch]);

side4b.push([x\_pos+tx4b, y\_pos+ty4b, z\_pos+z4\_pos-ch]);

side4c.push([x\_pos+tx4c, y\_pos+ty4c, z\_pos+z4\_pos+ch]);

side4d.push([x\_pos+tx4d, y\_pos+ty4d, z\_pos+z4\_pos+ch]);

side4e.push([x\_pos+tx4e, y\_pos+ty4e, z\_pos+z4\_pos-ch]);

side4f.push([x\_pos+tx4f, y\_pos+ty4f, z\_pos+z4\_pos-ch]);

side4g.push([x\_pos+tx4g, y\_pos+ty4g, z\_pos+z4\_pos+ch]);

side4h.push([x\_pos+tx4h, y\_pos+ty4h, z\_pos+z4\_pos+ch]);

}

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

{

mesh.quad(side1a[i], side1e[i],side1f[i],side1b[i]); //cubic最下面封口

mesh.quad(side4c[i], side4g[i],side4h[i],side4d[i]); //cubic最上面封口

//cubic內側面

mesh.quad(side1a[i], side1d[i],side1h[i],side1e[i]);

mesh.quad(side2a[i], side2d[i],side2h[i],side2e[i]);

mesh.quad(side3a[i], side3d[i],side3h[i],side3e[i]);

mesh.quad(side4a[i], side4d[i],side4h[i],side4e[i]);

//cubic外側面

mesh.quad(side1c[i], side1b[i],side1f[i],side1g[i]);

mesh.quad(side2c[i], side2b[i],side2f[i],side2g[i]);

mesh.quad(side3c[i], side3b[i],side3f[i],side3g[i]);

mesh.quad(side4c[i], side4b[i],side4f[i],side4g[i]);

if(i!=sides-1)

{

//上下斜通cubic

//1~2

mesh.quad(side1d[i], side2a[i+1],side2e[i+1],side1h[i]); //內側面

mesh.quad(side1c[i], side1g[i],side2f[i+1],side2b[i+1]); //外側面

mesh.quad(side1c[i], side2b[i+1],side2a[i+1],side1d[i]); //前面

mesh.quad(side2e[i+1], side2f[i+1],side1g[i],side1h[i]); //後面

//2~3

mesh.quad(side2d[i], side3a[i+1],side3e[i+1],side2h[i]); //內側面

mesh.quad(side2c[i], side2g[i],side3f[i+1],side3b[i+1]); //外側面

mesh.quad(side2c[i], side3b[i+1],side3a[i+1],side2d[i]); //前面

mesh.quad(side3e[i+1], side3f[i+1],side2g[i],side2h[i]); //後面

//3~4

mesh.quad(side3d[i], side4a[i+1],side4e[i+1],side3h[i]); //內側面

mesh.quad(side3c[i], side3g[i],side4f[i+1],side4b[i+1]); //外側面

mesh.quad(side3c[i], side4b[i+1],side4a[i+1],side3d[i]); //前面

mesh.quad(side4e[i+1], side4f[i+1],side3g[i],side3h[i]); //後面

//沿圓周通

//1

mesh.quad(side1h[i], side1d[i+1],side1a[i+1],side1e[i]); //內側面

mesh.quad(side1c[i+1], side1g[i],side1f[i],side1b[i+1]); //外側面

mesh.quad(side1g[i], side1c[i+1],side1d[i+1],side1h[i]); //上面

mesh.quad(side1f[i], side1e[i],side1a[i+1],side1b[i+1]); //下面

//2

mesh.quad(side2h[i], side2d[i+1],side2a[i+1],side2e[i]); //內側面

mesh.quad(side2c[i+1], side2g[i],side2f[i],side2b[i+1]); //外側面

mesh.quad(side2g[i], side2c[i+1],side2d[i+1],side2h[i]); //上面

mesh.quad(side2f[i], side2e[i],side2a[i+1],side2b[i+1]); //下面

//3

mesh.quad(side3h[i], side3d[i+1],side3a[i+1],side3e[i]); //內側面

mesh.quad(side3c[i+1], side3g[i],side3f[i],side3b[i+1]); //外側面

mesh.quad(side3g[i], side3c[i+1],side3d[i+1],side3h[i]); //上面

mesh.quad(side3f[i], side3e[i],side3a[i+1],side3b[i+1]); //下面

//4

mesh.quad(side4h[i],side4d[i+1],side4a[i+1],side4e[i]); //內側面

mesh.quad(side4c[i+1], side4g[i],side4f[i],side4b[i+1]); //外側面

mesh.quad(side4g[i], side4c[i+1],side4d[i+1],side4h[i]); //上面

mesh.quad(side4f[i], side4e[i],side4a[i+1],side4b[i+1]); //下面

}

}

//最後四邊形掃邊封口

//上下斜通cubic

//1~2

mesh.quad(side1d[sides-1], side2a[0],side2e[0],side1h[sides-1]); //內側面

mesh.quad(side1c[sides-1], side1g[sides-1],side2f[0],side2b[0]); //外側面

mesh.quad(side1c[sides-1], side2b[0],side2a[0],side1d[sides-1]); //前面

mesh.quad(side2e[0], side2f[0],side1g[sides-1],side1h[sides-1]); //後面

//2~3

mesh.quad(side2d[sides-1], side3a[0],side3e[0],side2h[sides-1]); //內側面

mesh.quad(side2c[sides-1], side2g[sides-1],side3f[0],side3b[0]); //外側面

mesh.quad(side2c[sides-1], side3b[0],side3a[0],side2d[sides-1]); //前面

mesh.quad(side3e[0], side3f[0],side2g[sides-1],side2h[sides-1]); //後面

//3~4

mesh.quad(side3d[sides-1], side4a[0],side4e[0],side3h[sides-1]); //內側面

mesh.quad(side3c[sides-1], side3g[sides-1],side4f[0],side4b[0]); //外側面

mesh.quad(side3c[sides-1], side4b[0],side4a[0],side3d[sides-1]); //前面

mesh.quad(side4e[0], side4f[0],side3g[sides-1],side3h[sides-1]); //後面

//沿圓周通

//1

mesh.quad(side1h[sides-1], side1d[0],side1a[0],side1e[sides-1]); //內側面

mesh.quad(side1c[0], side1g[sides-1],side1f[sides-1],side1b[0]); //外側面

mesh.quad(side1g[sides-1], side1c[0],side1d[0],side1h[sides-1]); //上面

mesh.quad(side1f[sides-1], side1e[sides-1],side1a[0],side1b[0]); //下面

//2

mesh.quad(side2h[sides-1], side2d[0],side2a[0],side2e[sides-1]); //內側面

mesh.quad(side2c[0], side2g[sides-1],side2f[sides-1],side2b[0]); //外側面

mesh.quad(side2g[sides-1], side2c[0],side2d[0],side2h[sides-1]); //上面

mesh.quad(side2f[sides-1], side2e[sides-1],side2a[0],side2b[0]); //下面

//3

mesh.quad(side3h[sides-1], side3d[0],side3a[0],side3e[sides-1]); //內側面

mesh.quad(side3c[0], side3g[sides-1],side3f[sides-1],side3b[0]); //外側面

mesh.quad(side3g[sides-1], side3c[0],side3d[0],side3h[sides-1]); //上面

mesh.quad(side3f[sides-1], side3e[sides-1],side3a[0],side3b[0]); //下面

//4

mesh.quad(side4h[sides-1], side4d[0],side4a[0],side4e[sides-1]); //內側面

mesh.quad(side4c[0], side4g[sides-1],side4f[sides-1],side4b[0]); //外側面

mesh.quad(side4g[sides-1], side4c[0],side4d[0],side4h[sides-1]); //上面

mesh.quad(side4f[sides-1], side4e[sides-1],side4a[0],side4b[0]); //下面

return Solid.make(mesh);

}