**4.2 中心線為基準之基準立方體**

var Conversions = Core.Conversions;

var Debug = Core.Debug;

var Path2D = Core.Path2D;

var Point2D = Core.Point2D;

var Point3D = Core.Point3D;

var Matrix2D = Core.Matrix2D;

var Matrix3D = Core.Matrix3D;

var Mesh3D = Core.Mesh3D;

var Plugin = Core.Plugin;

var Tess = Core.Tess;

var Sketch2D = Core.Sketch2D;

var Solid = Core.Solid;

var Vector2D = Core.Vector2D;

var Vector3D = Core.Vector3D;

params =

[

{ "id": "x",

"displayName": "x size",

"type": "float",

"rangeMin": 0,

"rangeMax": 200, //桌面最大為200mm，因此物件最大設為200mm。

"default": 20 //xyz方向尺寸皆預設為20mm

},

{ "id": "y",

"displayName": "y size",

"type": "float",

"rangeMin": 0,

"rangeMax": 200, "default": 20

},

{ "id": "z",

"displayName": "z size",

"type": "float",

"rangeMin": 0,

"rangeMax": 200,

"default": 20

},

{ "id": "x\_pos",

"displayName": "x position",

"type": "float",

"rangeMin": -200,

"rangeMax": 200,

"default": 0

//預設100是因為將圖形放到桌面中央時，調整x、y position  
 //為零時，可將圖中心線形移往桌面左下角。

},

{ "id": "y\_pos",

"displayName": "y position",

"type": "float",

"rangeMin": -200,

"rangeMax": 200,

"default": 0

},

{ "id": "z\_pos",

"displayName": "z position",

"type": "float",

"rangeMin": -200,

"rangeMax": 200,

"default": 0

}

];

// Shape Generator

function process(params)

{

var x = params.x;

var y = params.y;

var z = params.z;

var x\_pos = params.x\_pos;

var y\_pos = params.y\_pos;

var z\_pos = params.z\_pos;

var A0 = [x\_pos-0.5\*x,y\_pos-0.5\*y,z\_pos-0.5\*z];   
 //圖形變化由中心點向兩端延伸

var A1 = [x\_pos+0.5\*x,y\_pos-0.5\*y,z\_pos-0.5\*z];

var A2 = [x\_pos+0.5\*x,y\_pos+0.5\*y,z\_pos-0.5\*z];

var A3 = [x\_pos-0.5\*x,y\_pos+0.5\*y,z\_pos-0.5\*z];

var B0 = [x\_pos-0.5\*x,y\_pos-0.5\*y,z\_pos+0.5\*z];

var B1 = [x\_pos+0.5\*x,y\_pos-0.5\*y,z\_pos+0.5\*z];

var B2 = [x\_pos+0.5\*x,y\_pos+0.5\*y,z\_pos+0.5\*z];

var B3 = [x\_pos-0.5\*x,y\_pos+0.5\*y,z\_pos+0.5\*z];

var mesh = new Mesh3D();

mesh.triangle(A0, A2, A1);

mesh.triangle(A0, A2, A3);

mesh.triangle(B0, B1, B2);

mesh.triangle(B0, B2, B3);

mesh.triangle(A0, A1, B1);

mesh.triangle(A0, B1, B0);

mesh.triangle(A1, B1, B2);

mesh.triangle(A1, B2, A2);

mesh.triangle(A2, B2, B3);

mesh.triangle(A2, B3, A3);

mesh.triangle(A3, A0, B0);

mesh.triangle(A3, B0, B3);

var solid =Solid.make(mesh); //以mesh做出立體

return solid;

}