

# Procedura generale per disegnare un robot 3D.

La procedura viene spiegata facendo riferimento al programma SCARA.pde, che rappresenta in effetti un esempio applicativo della stessa.

## La funzione `box(float dx, float dy, float dz)`

Ciascun link del robot viene disegnato con la funzione **`box(dx, dy, dz)`**, che disegna un parallelepipedo di dimensioni dx, dy e dz con dx lungo l'asse x corrente (che all'inizio è l'asse orizzontale da sinistra a destra), dy lungo y (che all'inizio è l'asse verticale verso il basso) e dz (che all'inizio è l'asse uscente dallo schermo - N.B. la terna non è destrorsa). Il parallelepipedo viene disegnato con l'origine della terna corrente nel suo centro di simmetria.

## La procedura

Si descrive la procedura spiegando le linee di codice del robot SCARA.

Si disegna prima la base, cioè **il link 0**, che va disegnato nel punto (xBase,yBase) individuato con un click di mouse: a tal fine si fa il `translate(xBase,yBase)` e si disegna il parallelepipedo.

```
// Link 0 (base)
translate(xBase,yBase);
box(d0x,d0y,d0z);
```

Si passa quindi al **link 1**. Il link ruota col giunto 1 di `theta[0]` rispetto all'asse y. Per questo si fa il **`rotateY(theta[0])`**. In questo modo l'asse x viene ruotato di `theta[0]` gradi (in senso orario se visto da y) e con esso anche l'asse z mentre l'asse y resta fermo (verticale verso il basso). Il **`translate`** successivo serve per portare l'origine al centro del link da disegnare, traslazione che viene fatta nel nuovo sistema di coordinate, quindi di  $(d1x-d0x)/2$  rispetto al nuovo asse x e di  $-(d0y+d1y)/2$  rispetto al nuovo asse y (che coincide in realtà col vecchio).

```
// Link 1 (si muove con theta1 = theta[0])
rotateY(theta[0]);
translate((d1x-d0x)/2,-(d0y+d1y)/2,0);
box(d1x,d1y,d1z);
```

Per il **link 2** si muove l'origine sull'asse del giunto 2 che è il perno della rotazione successiva. A tal fine si fa un **`translate`** di  $(d1x-d2z)/2$  rispetto all'asse x corrente (che è quello lungo cui è puntato il link 1). Il fatto che c'è `d2z` è perché si vuole fare in modo che quando il link 2 per esempio ha ruotato di 90 gradi non sporga dal link precedente ma sia esattamente alla fine di esso (vedere programma in azione). A questo punto si ruota di `theta2` rispetto all'asse y (sempre verticale) ruotando quindi l'asse x e z. Il successivo **`translate`** serve per portare l'origine al centro del link da disegnare, traslazione che viene fatta nel nuovo sistema di coordinate, quindi di  $(d2x-d2z)/2$  rispetto al nuovo asse x.

```
// Link 2 (si muove con theta2 = theta[1])
translate((d1x-d2z)/2,0,0);
rotateY(theta[1]);
translate((d2x-d2z)/2,0,0);
box(d2x,d2y,d2z);
```

Il **link 3** viene disegnato spostando l'origine di  $(d2x-d3x)/2$  lungo l'asse x corrente (cioè alla fine del link precedente) e di  $\theta[2]$  (cioè  $d3$ ) in verticale (il giunto 3 è prismatico). Una traslazione positiva porta l'organo terminale verso il basso (y positiva è verso il basso).

```
// Link 3 (si muove con  $d3 = \theta[2]$ )  
translate((d2x-d3x)/2,theta[2],0);  
box(d3x,d3y,d3z);
```

Le funzioni **pushMatrix()** e **popMatrix()** non sarebbero necessarie se le scritte di testo fossero messe all'inizio di draw. Infatti in ogni ciclo di **draw()** il sistema di riferimento viene comunque riportato al suo valore di default.