

## TP<sub>2</sub> : Hiérarchie Transformation et Animation

Dans ce module, nous utiliserons *Code::Blocks* sous Linux (Débian) avec les bibliothèques *gl* (*libgl1-mesa-glx*), *freeglut* (*freeglut3-dev*), *libjpeg* (*libjpeg62-turbo-dev*) et le débogueur *gdb*.

### Créer un projet avec Code::Blocks

Démarrer sous Debian, lancer Code::Blocks,

Créer un projet vide [créer un projet Code::Blocks](#)

Placer le fichier [cube.cpp](#) dans le répertoire de votre projet



### Modélisation du bras du robot

A partir de l'arbre complet réalisé en TD, utiliser :

- les fonctions `glutSolid...` pour dessiner les primitives,
- `glPushMatrix()`,
- les transformations,
- la projection orthogonale `glOrtho(...)` pour agrandir le volume de vision

pour coder le bras du robot :

- Socle de taille ( $L_{socle} \times H_{socle} \times l_{socle}$ )
- Rotule1 de rayon  $R_{rotule}$
- Bras cylindrique de taille  $L_{bras}$  et de rayon  $R_{bras}$
- Rotule2 de rayon  $R_{rotule}$
- AvantBras cylindrique de taille  $L_{avbras}$  et de rayon  $R_{avbras}$
- Pince constituée
  - d'une base parallélépipédique de taille ( $L_{main} \times H_{main} \times l_{main}$ )
  - de deux doigts parallélépipédiques symétriques par rapport au plan axial de la base de taille ( $L_{doigt} \times H_{doigt} \times l_{doigt}$ )

A titre indicatif les dimensions pour tester votre programme :

$L_{socle} = 2$ ;  $H_{socle} = 1/2$ ;  $l_{socle} = 1$ ;

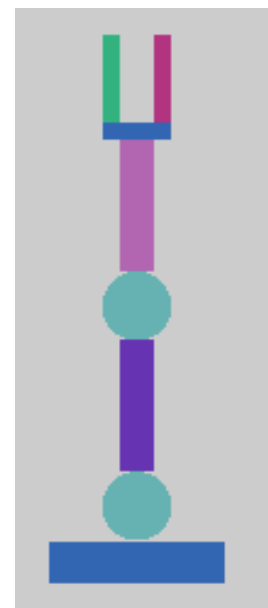
$L_{bras} = 2$ ;  $R_{bras} = 0.3$ ;

$L_{avbras} = 1.5$ ;  $R_{avbras} = 0.2$ ;

$L_{main} = 0.8$ ;  $H_{main} = 0.2$ ;  $l_{main} = 0.8$ ;

$L_{doigt} = 0.2$ ;  $H_{doigt} = 1$ ;  $l_{doigt} = 0.8$ ;

$R_{rotule} = 0.4$ .

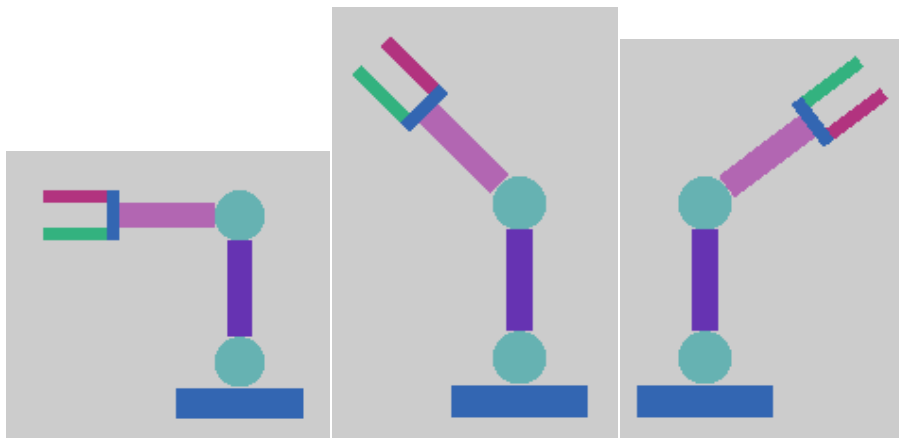


### Animation du bras de robot articulé et pince

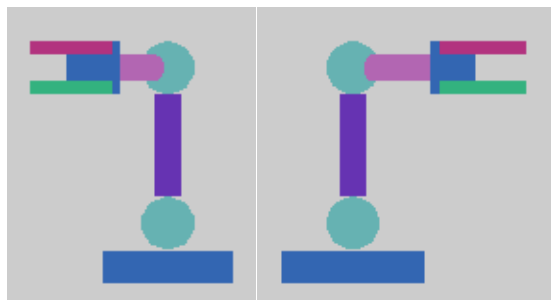
1. Rotation de l'avant-bras sur lui-même automatiquement (idle).



2. Rotation de l'avant-bras autour de la rotule de l'avant-bras d'angle droit d'un côté jusqu'à l'autre côté et inversement grâce aux touches r et R.



3. Rotation automatique de la caméra autour du bras du robot



... Enchaîner avec le TD3 puis le TP3