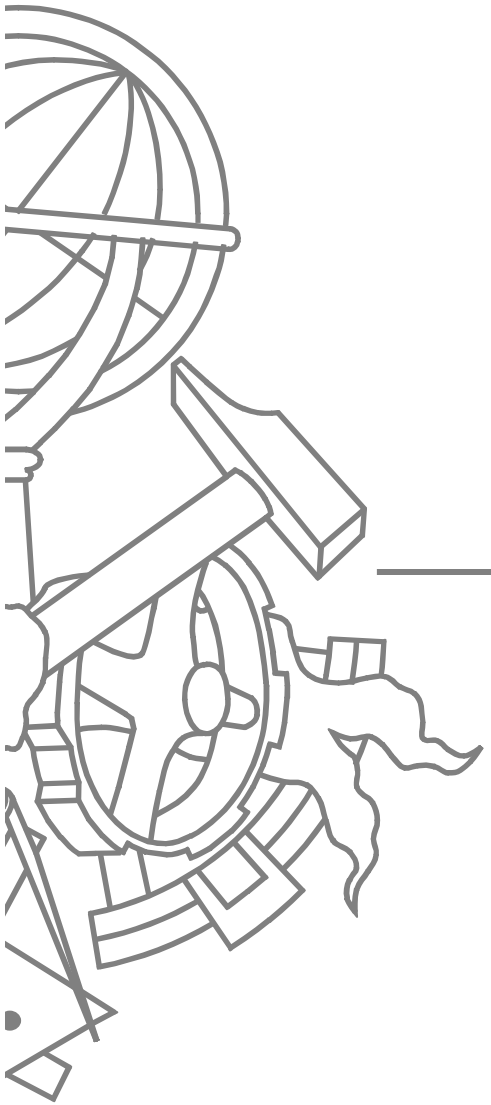


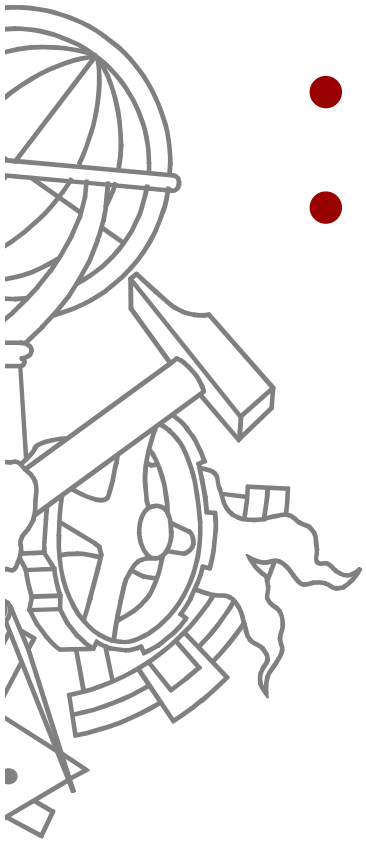
Projecções e modelos hierárquicos

Aula 5

Sistemas Gráficos e Interactivos
Instituto Superior de Engenharia do Porto

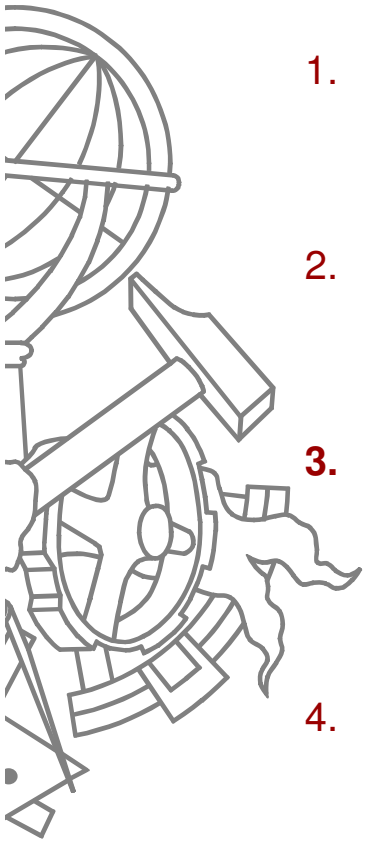
Paulo Gandra de Sousa
psousa@dei.isep.ipp.pt

Conteúdo

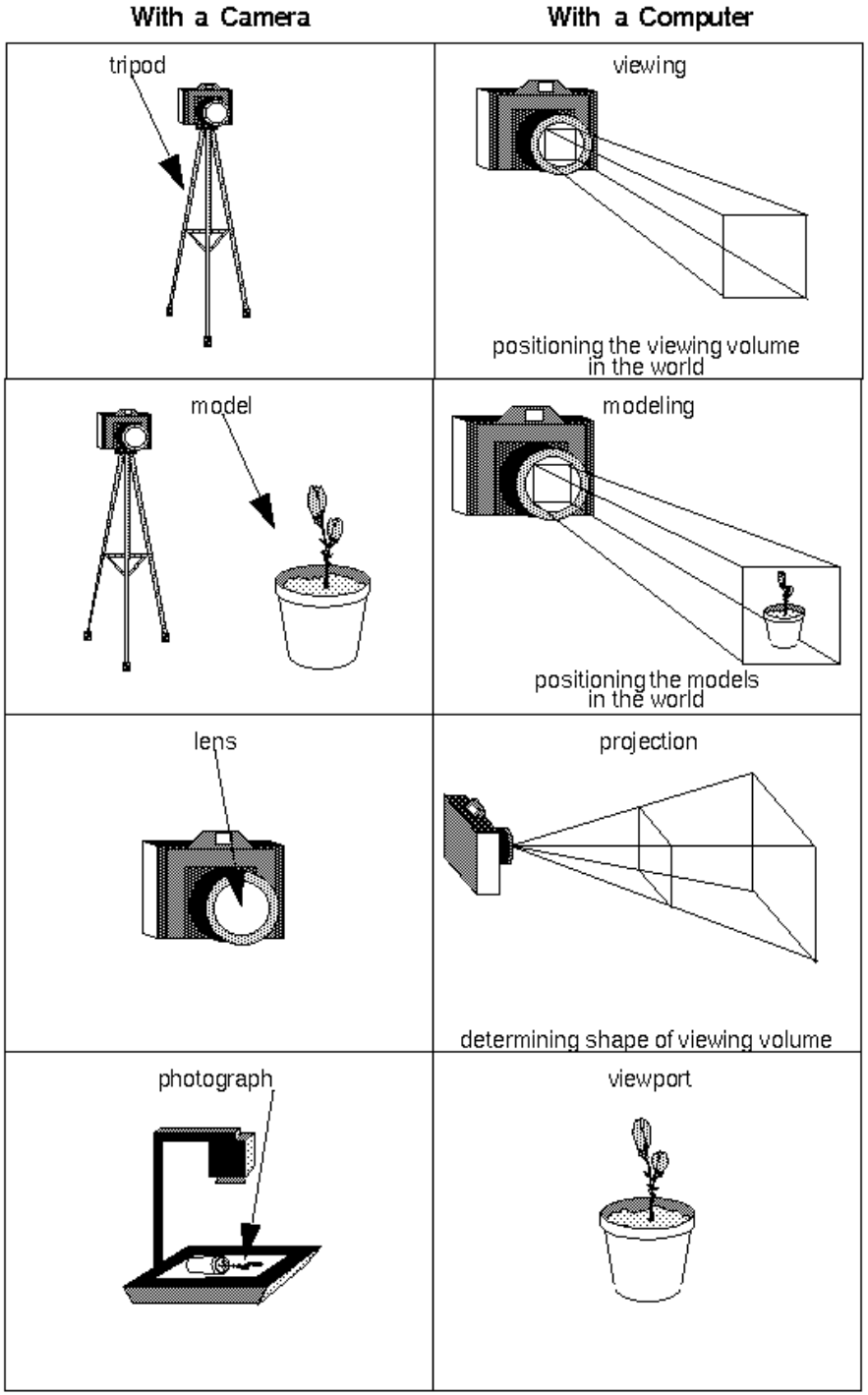


- Projecções
- Modelos hierárquicos de objectos

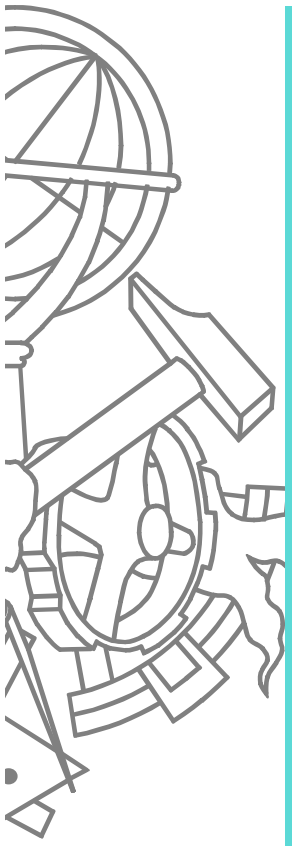
Relembrando...



1. Apontar a câmara à cena (viewing transformation).
2. Compor a cena (modeling transformation).
3. **Escolher o tipo de lente e acertar o zoom (projection transformation).**
4. Determinar o tamanho físico da cena (viewport transformation).

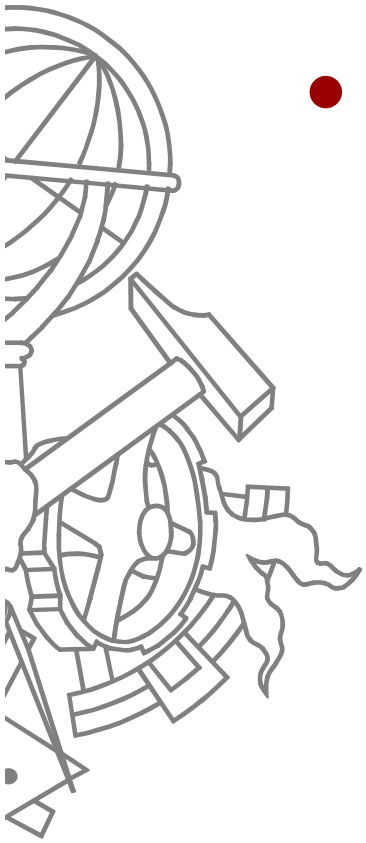


Esqueleto de código



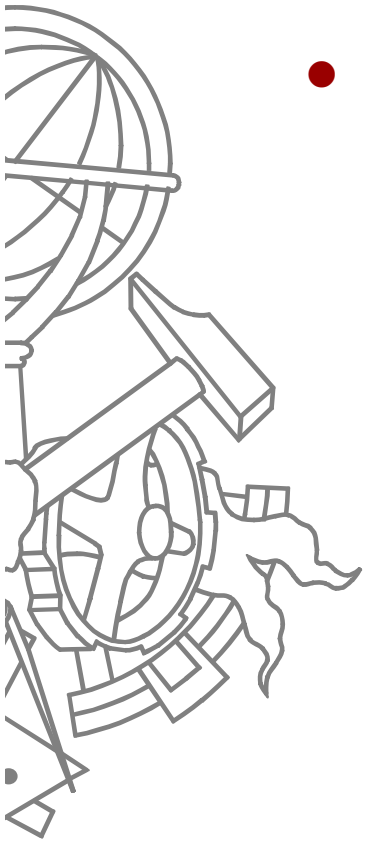
```
void reshape(int w, int h) {
   // viewport transformation
   glViewport(0, 0, w, h);
   // projection transformation
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   projeccao();
   ...
}

void display() {
   // modelview transformation
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
   // posicionamento da câmara
   camara();
   // transformações do modelo
   ...
}
```

O que é a transformação de projecção?

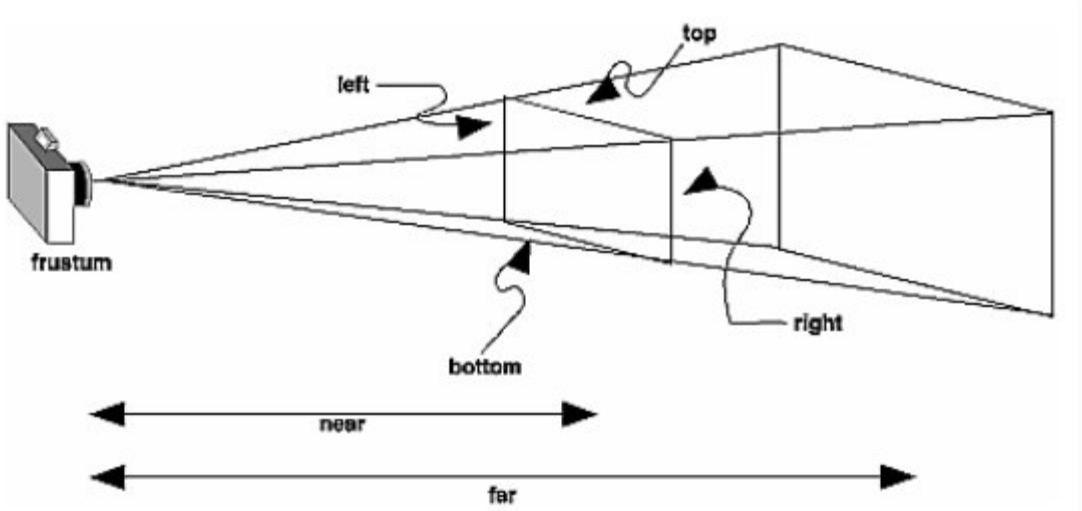


- The purpose of the projection transformation is to define a *viewing volume*, which is used in two ways.
  - The viewing volume determines how an object is projected onto the screen (that is, by using a perspective or an orthographic projection), and
  - defines which objects or portions of objects are clipped out of the final image.

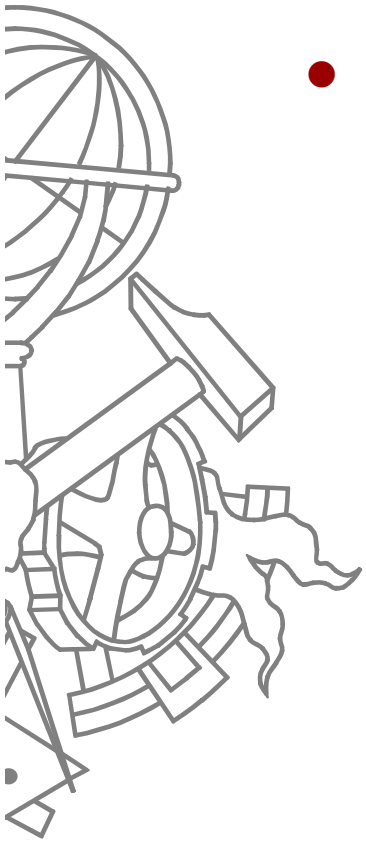
Perspectiva



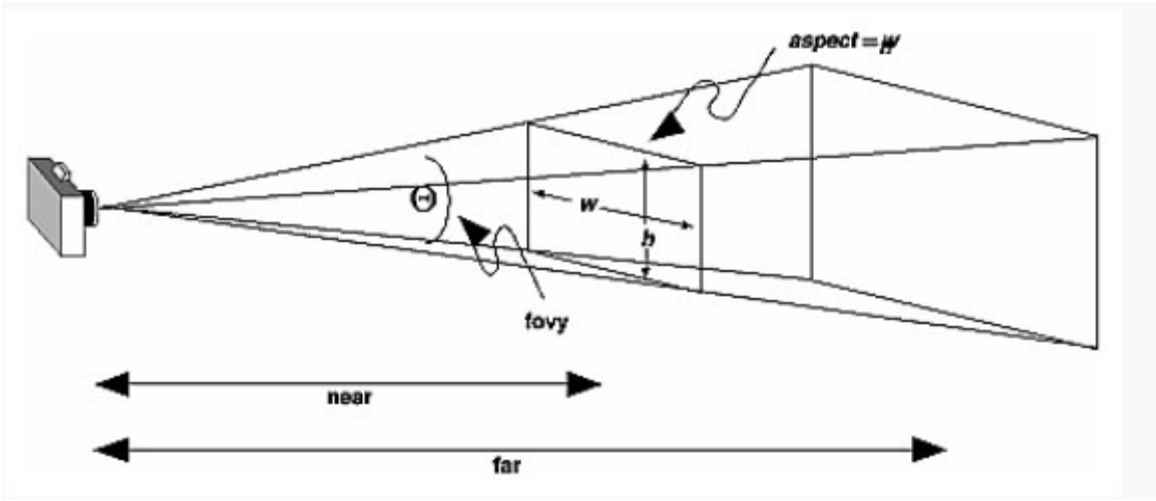
- `void glFrustum(`
  `GLdouble left, GLdouble right,`
  `GLdouble bottom, GLdouble top,`
  `GLdouble zNear, GLdouble zFar`
  `)`



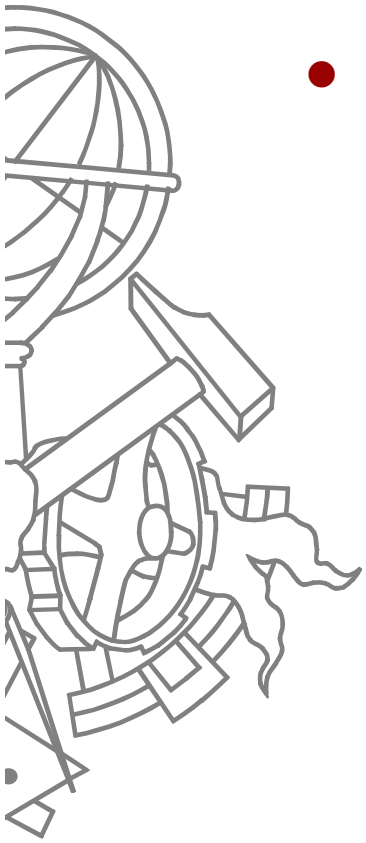
Perspectiva



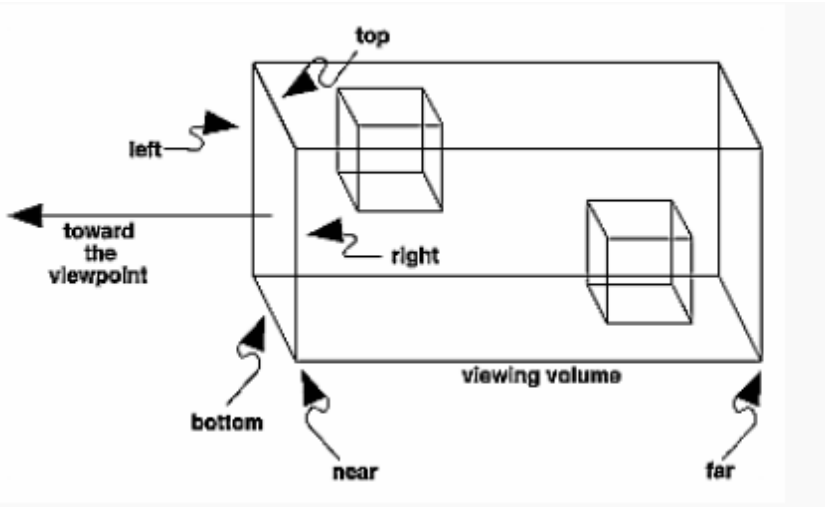
- `void gluPerspective(`
  `GLdouble fovy, GLdouble aspect,`
  `GLdouble zNear, GLdouble zFar`
  `);`



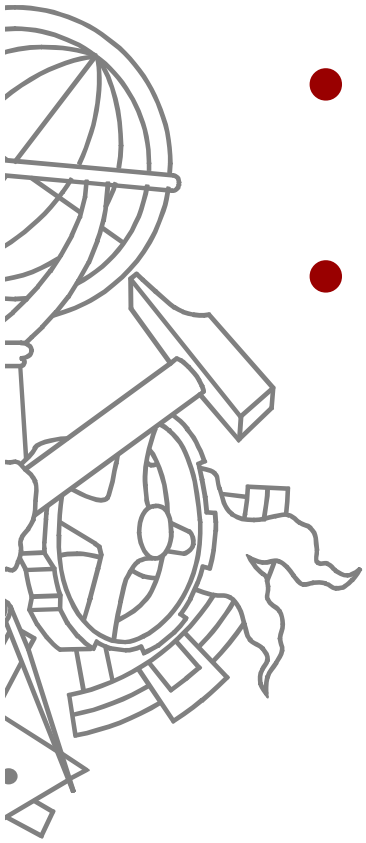
Ortográfica



- `void glOrtho(`
  `GLdouble left, GLdouble right,`
  `GLdouble bottom, GLdouble top,`
  `GLdouble zNear, GLdouble zFar`
  `);`

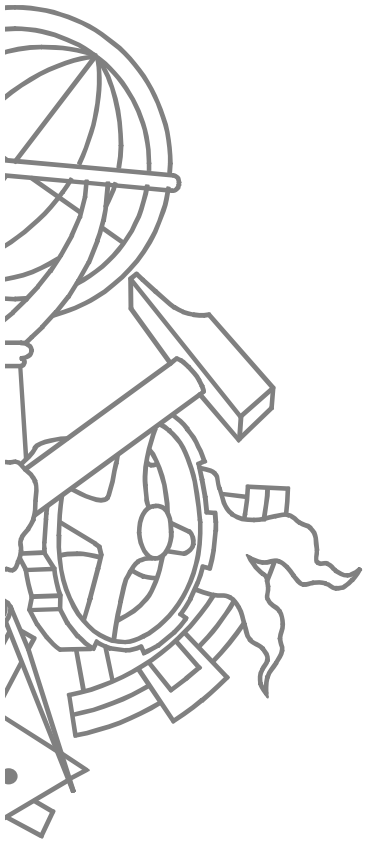


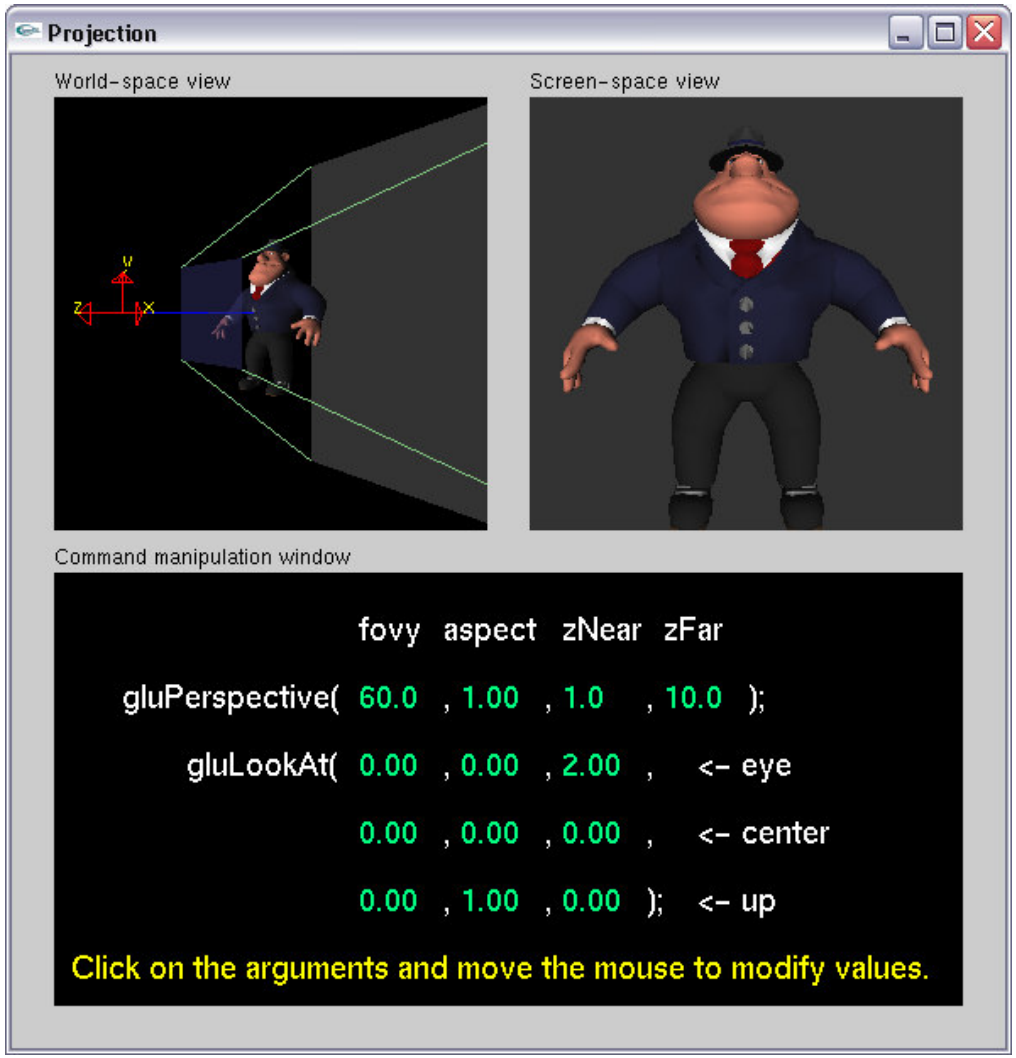
zNear & zFar



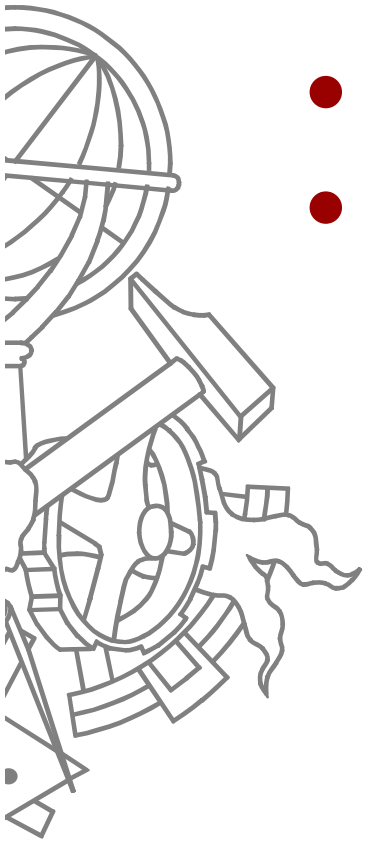
- zNear e zFar definem os planos de corte (*clipping*) **relativos** à posição da câmara
- Exemplo:
  - zNear = 1 & zFar = 3
  - Câmara (0, 0, 0)
    - Só são visíveis objectos com coordenada z no intervalo [-1, -3]
  - Mover câmara para (0, 0, 3)
    - Só são visíveis objectos com coordenada z no intervalo [2, 0]

Demo



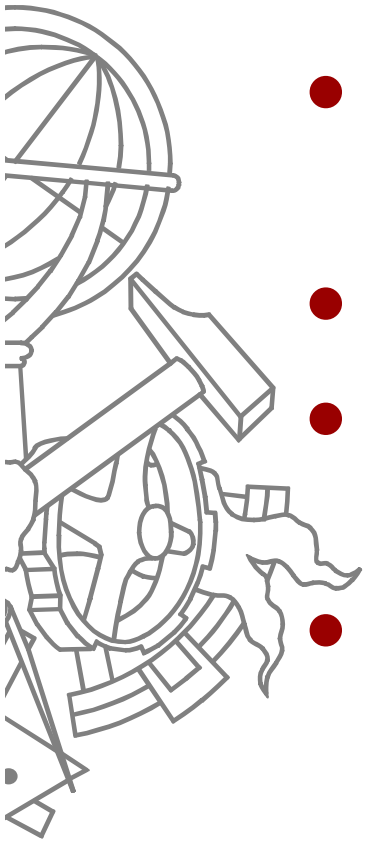

Projection
World-space view
Screen-space view
Command manipulation window
fovy aspect zNear zFar
gluPerspective(60.0 , 1.00 , 1.0 , 10.0);
gluLookAt(0.00 , 0.00 , 2.00 , <- eye
0.00 , 0.00 , 0.00 , <- center
0.00 , 1.00 , 0.00); <- up
Click on the arguments and move the mouse to modify values.

Modelos hierárquicos



- Objectos compostos por vários objectos
- Com “articulações”
  - Exemplos:
    - braço de um robot
    - Sistema solar

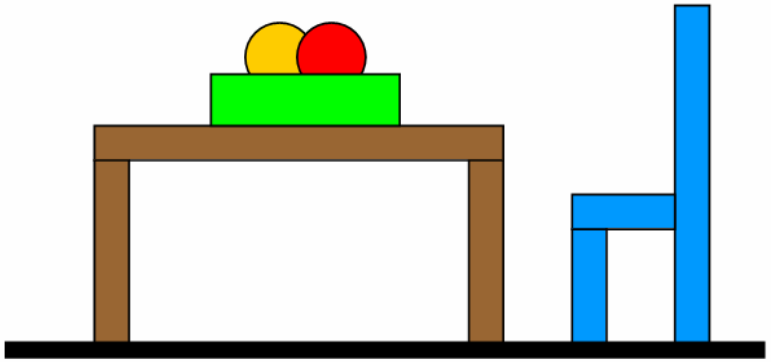
Modelos hierarquicos

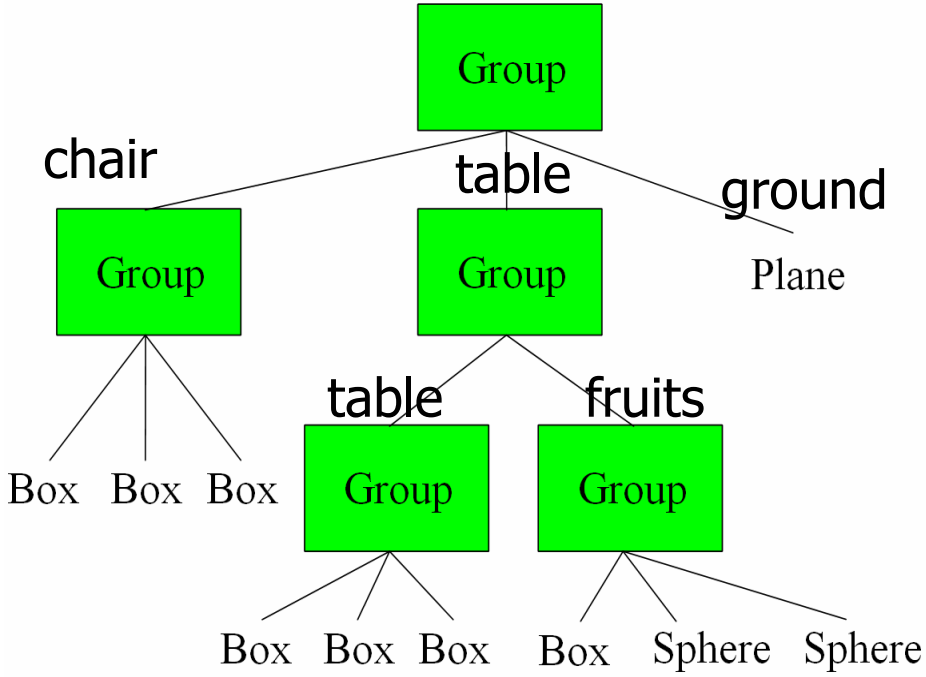


- Estruturas em árvore para representar o modelo
- Cada nó é uma transformação ou objecto
- Nós de um mesmo ramo representam transformações acumuladas
- Nós de ramos diferentes são transformações independentes

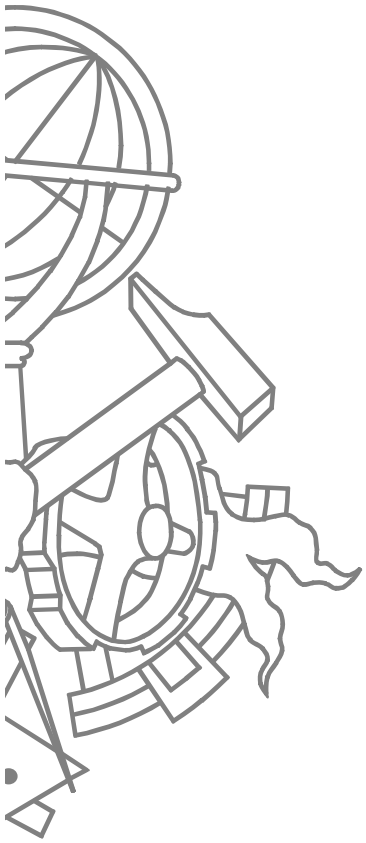
Hierarchical Grouping of Objects

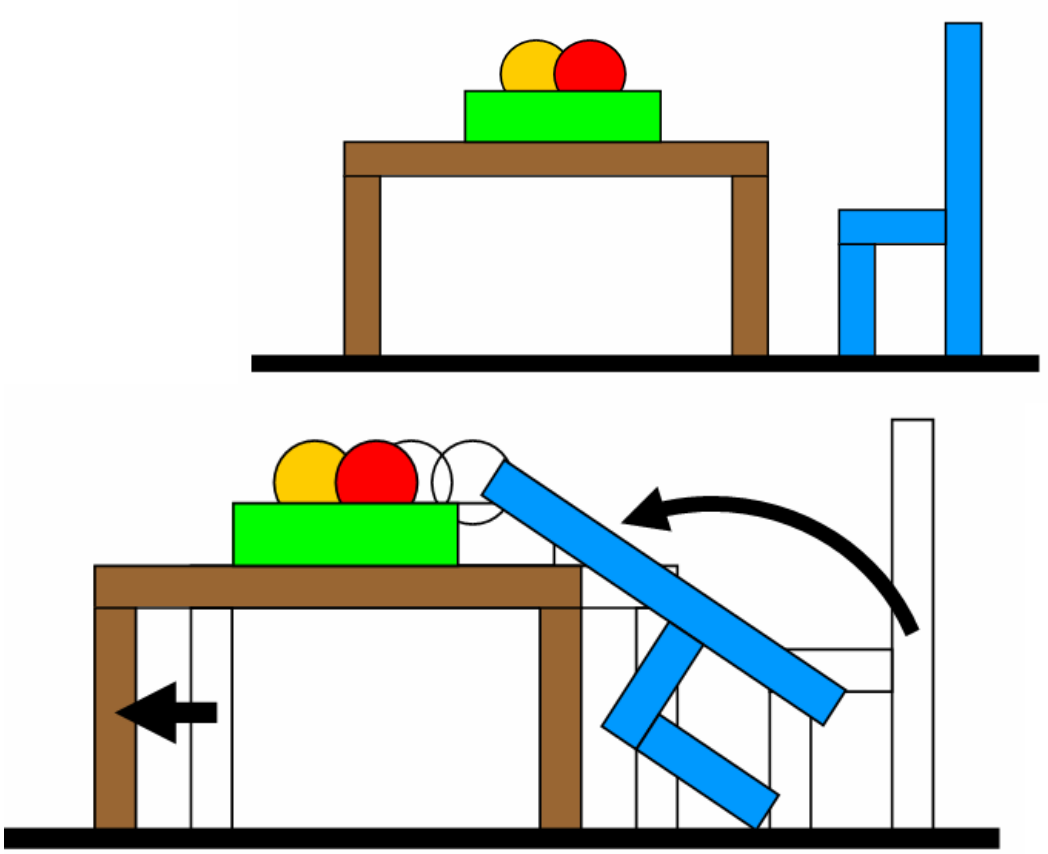
- Logical organization of scene





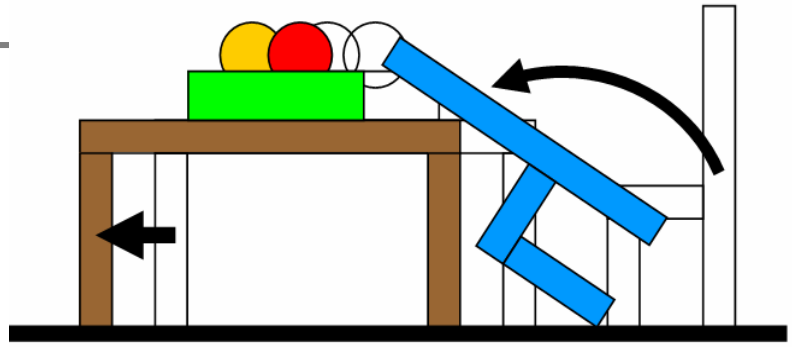
Adding Transformations

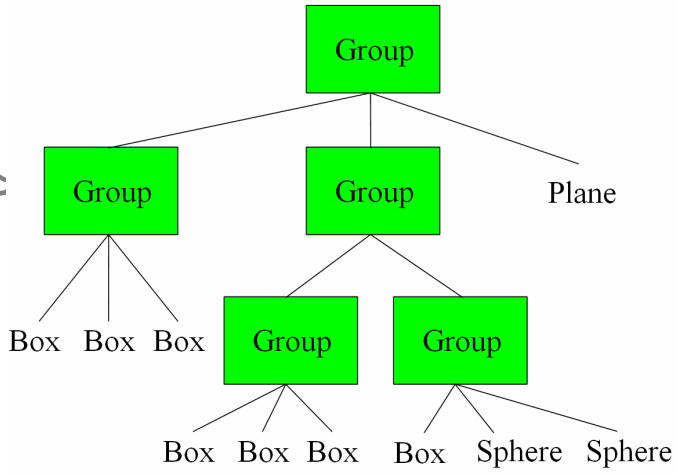


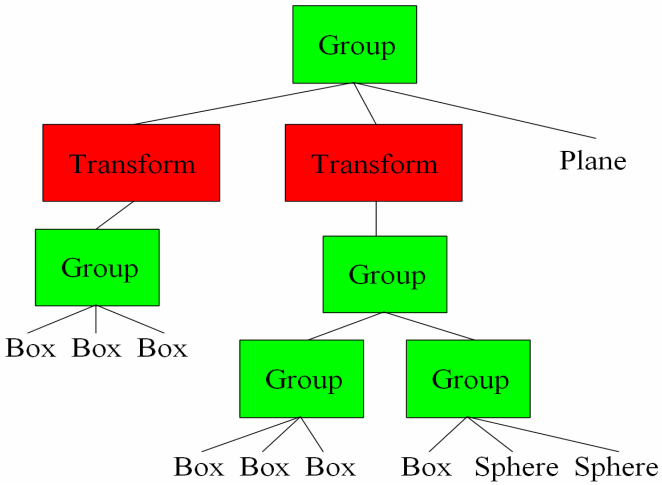


Hierarchical Transformation of Objects

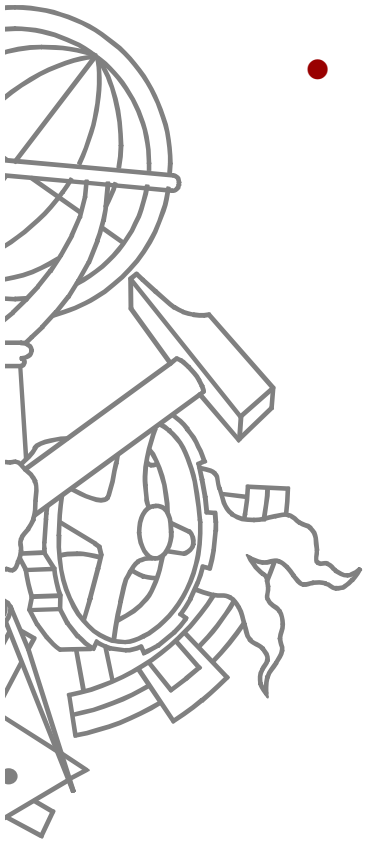
- Transforms position logical groupings of objects within the scene



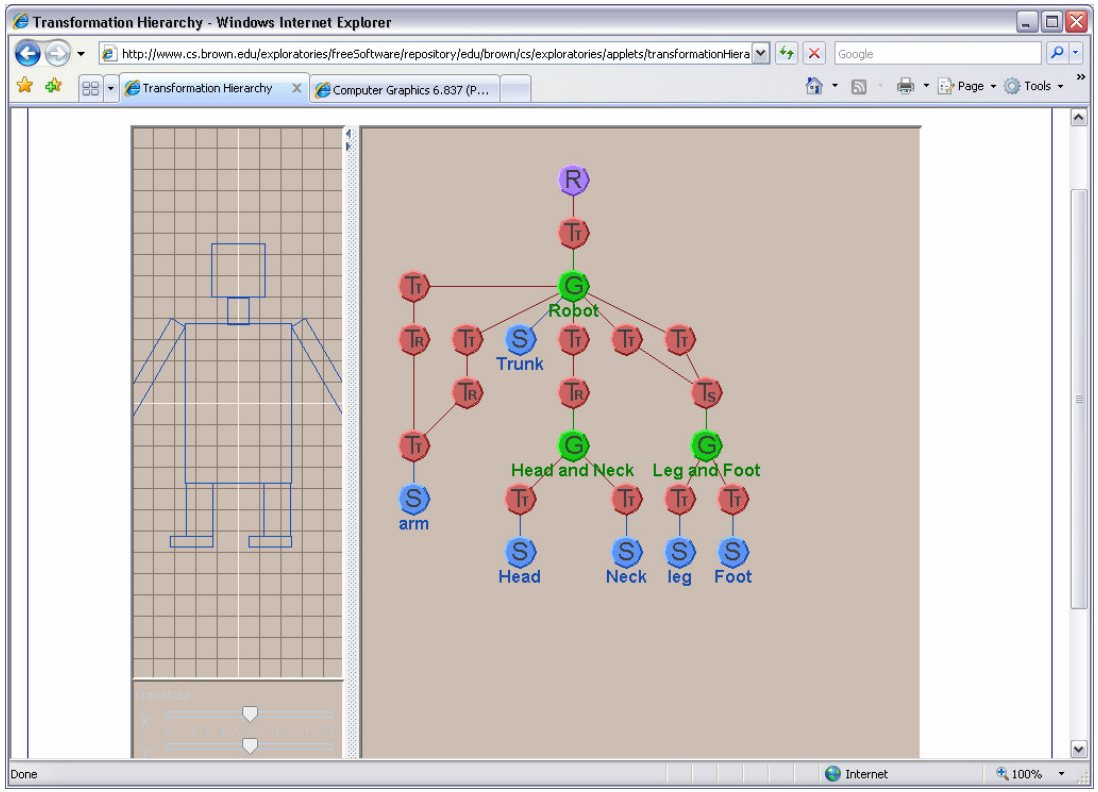




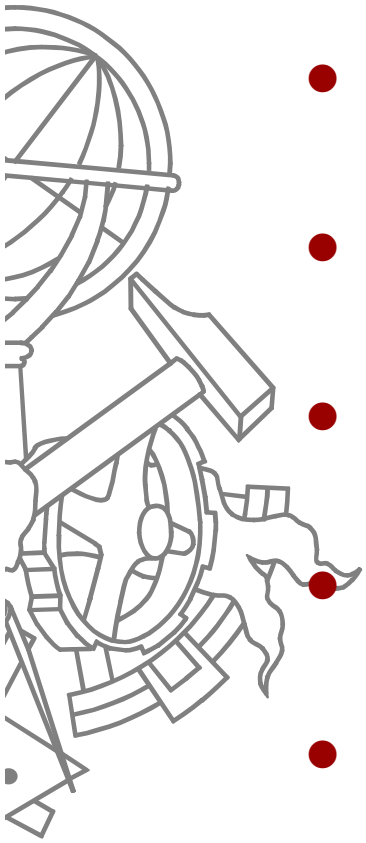
Demo



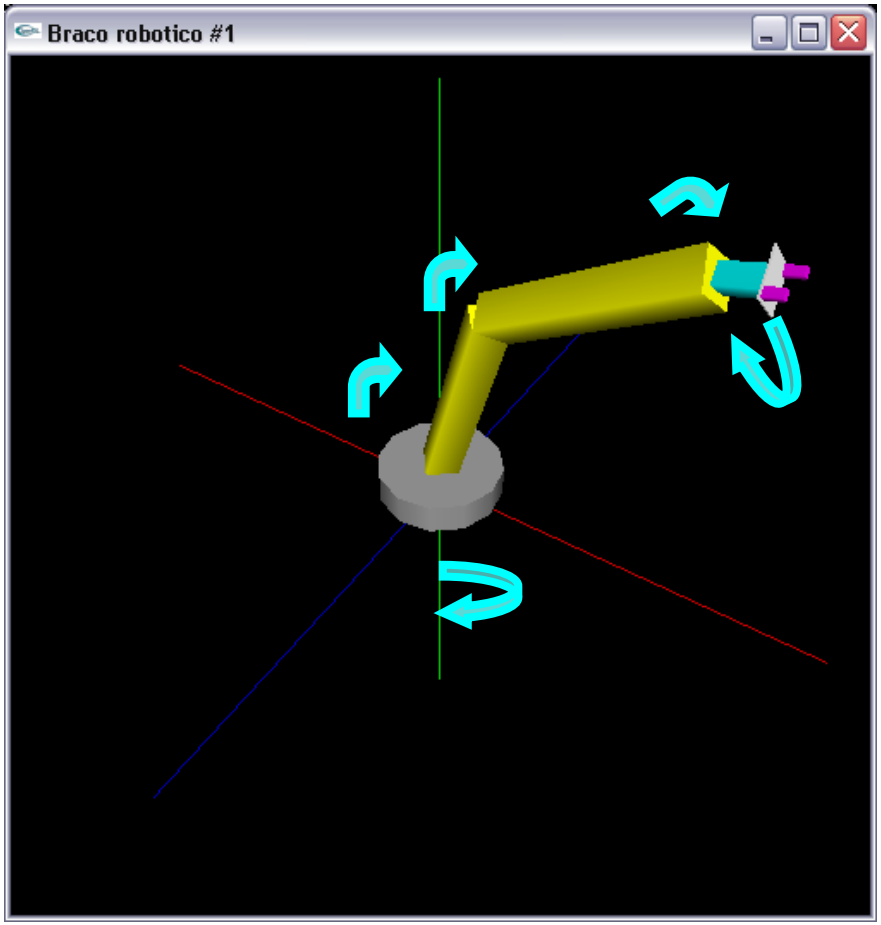
- http://www.cs.brown.edu/exploratories/freeSoftware/repository/edu/brown/cs/exploratories/applets/transformationHierarchy/transformation_hierarchy_guide.html



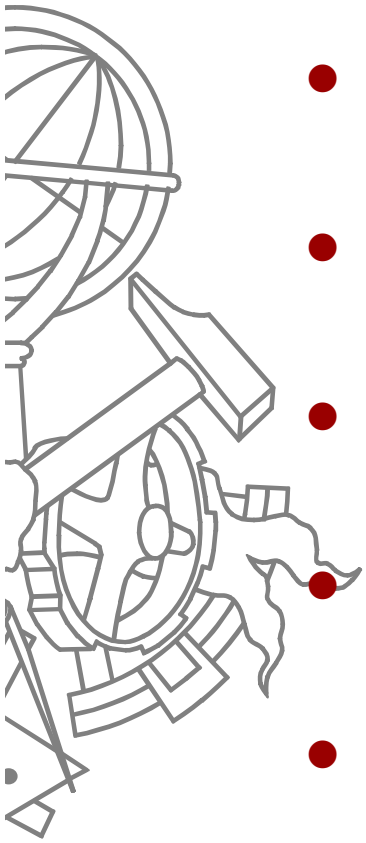
Exemplo: braço robot



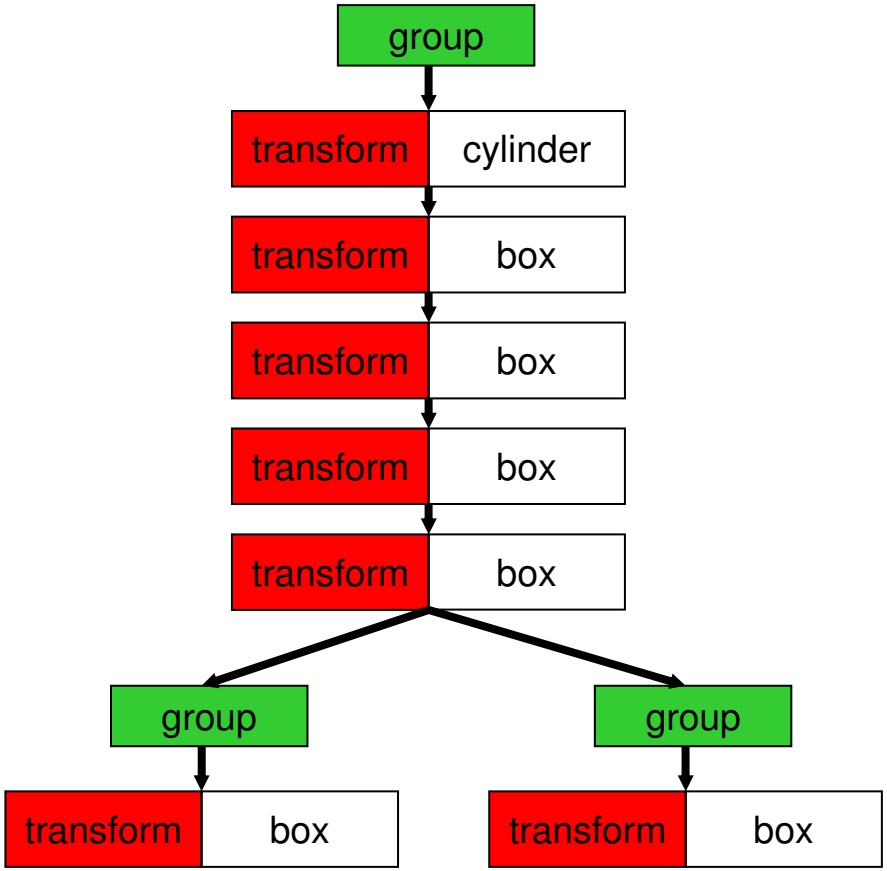
- Base
  - Rotação no plano
- 1º segmento
  - Rotação no eixo
- 2º segmento
  - Rotação no eixo
- Pulso
  - Rotação no eixo
- Garra
  - rotação no plano
  - Fecha/abre



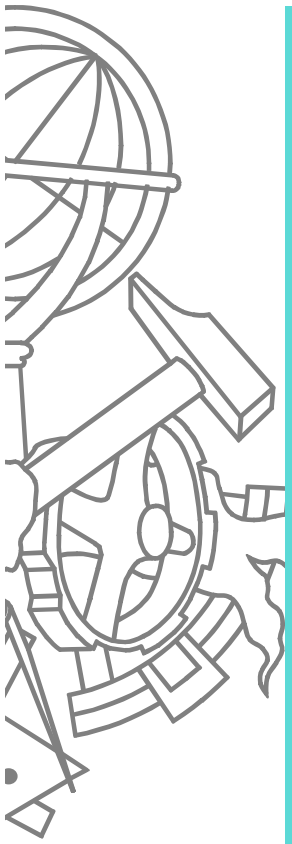
Exemplo: braço robot



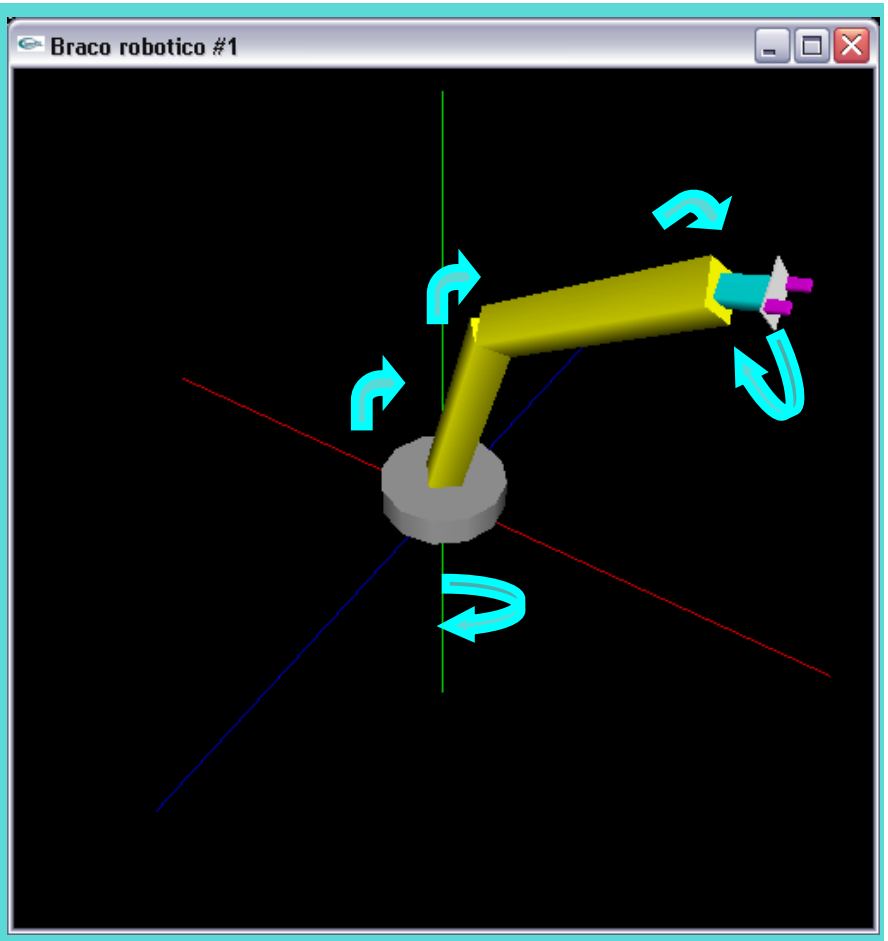
- Base
  - Rotação no plano
- 1º segmento
  - Rotação no eixo
- 2º segmento
  - Rotação no eixo
- Pulso
  - Rotação no eixo
- Garra
  - Rotação no plano
  - Fecha/abre



Classe RobotArm

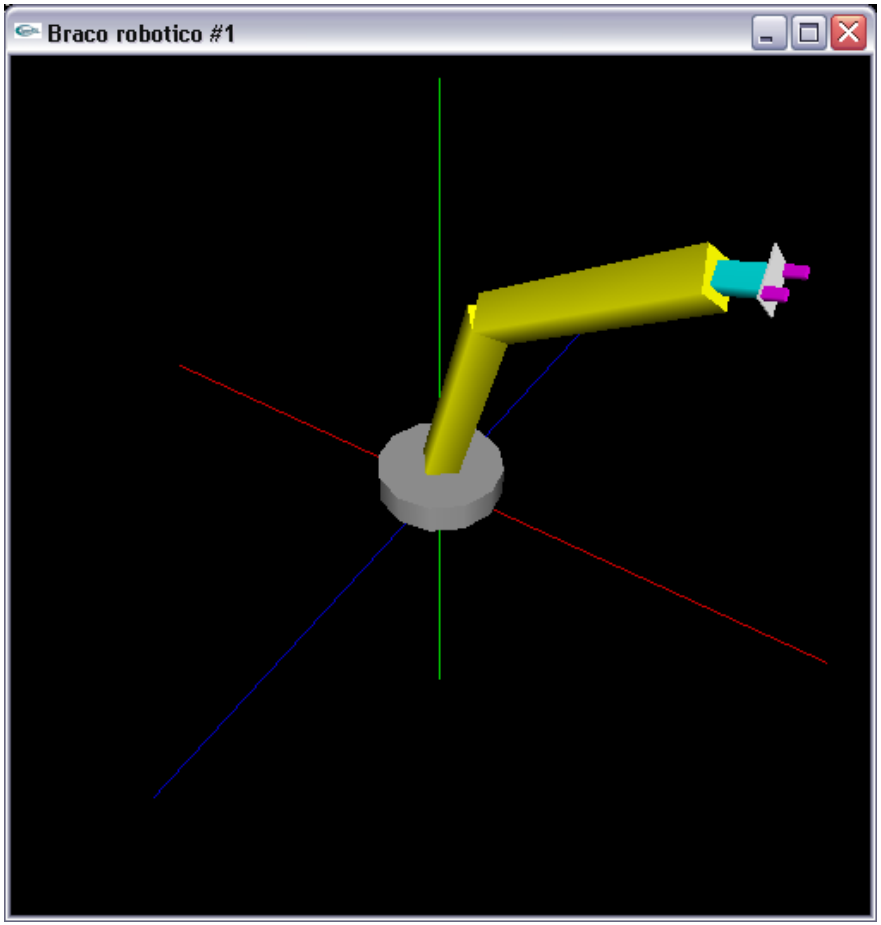


```
class RobotArm
{
   GLfloat rotBase;
   GLfloat rotSeg1;
   GLfloat rotSeg2;
   GLfloat rotWrist;
   GLfloat rotClaw;
   bool clawOpened;

   ...
}
```

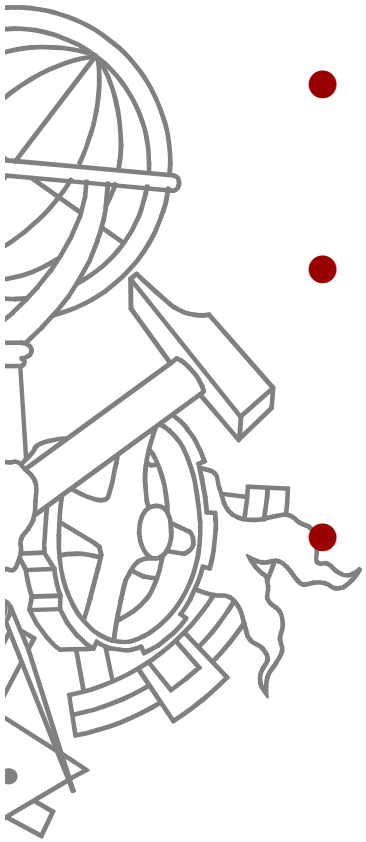


```
glPushMatrix();
      //base
      glRotatef(rotBase, 0, 0, 1);
      cylinderWithTopAndBottom(mode, BASE_RADIUS, BASE_HEIGHT, 12, 2);
      //segmento 1
      glTranslatef(0, 0, BASE_HEIGHT);
      glRotatef(rotSeg1, 1, 0, 0);
      box(mode, SEG1_WIDTH, SEG1_LENGTH);
      //segmento 2
      glTranslatef(0, 0, SEG1_LENGTH);
      glRotatef(rotSeg2, 1, 0, 0);
      box(mode, SEG2_WIDTH, SEG2_LENGTH);
      //pulso
      glTranslatef(0, 0, SEG2_LENGTH);
      glRotatef(rotWrist, 1, 0, 0);
      box(mode, WRIST_WIDTH, WRIST_LENGTH);
      //garra
      glTranslatef(0, 0, WRIST_LENGTH);
      glRotatef(rotClaw, 0, 0, 1);
      box(mode, CLAW_BASE_WIDTH, CLAW_BASE_LENGTH);
      // pinças
      glTranslatef(0, 0, CLAW_BASE_LENGTH);
      float d = (clawOpened ? CLAW_BASE_WIDTH/2 : CLAW_WIDTH/2);
      // pinça "direita"
      glPushMatrix();
            glTranslatef(-d, 0, 0);
            box(mode, CLAW_WIDTH, CLAW_LENGTH);
      glPopMatrix();
      // pinça "esquerda"
      glPushMatrix();
            glTranslatef(+d, 0, 0);
            box(mode, CLAW_WIDTH, CLAW_LENGTH);
      glPopMatrix();
glPopMatrix();
```

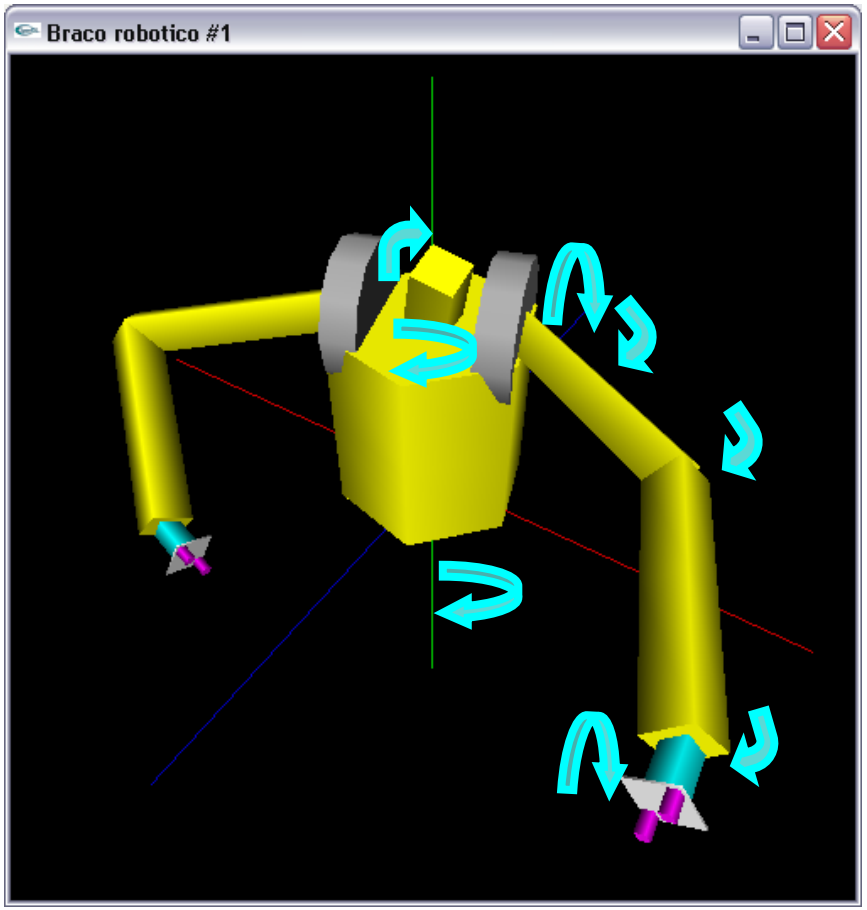
Demo



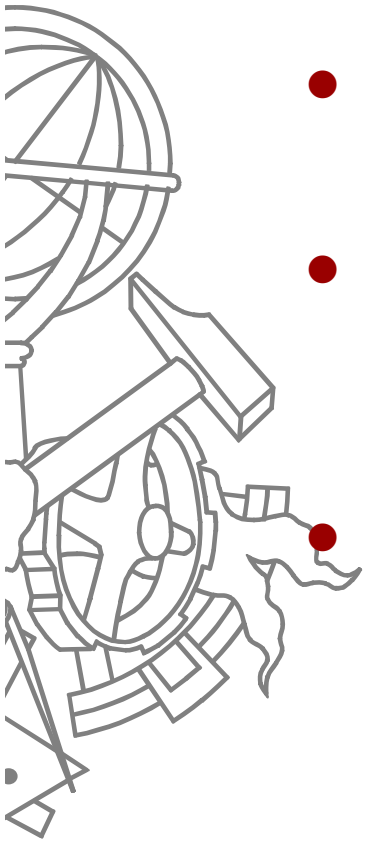
Exemplo: corpo robot



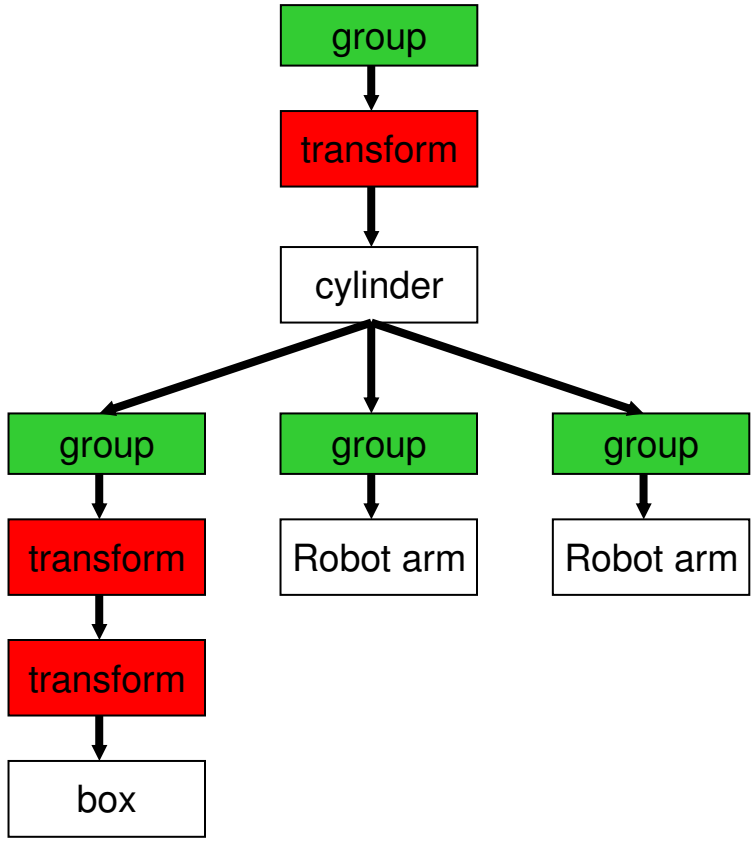
- Tronco
  - Rotação no plano
- Pescoço
  - Rotação no plano
  - Rotação no eixo
- Dois braços



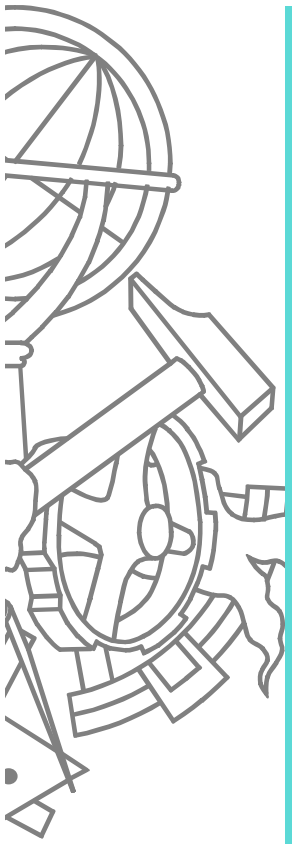
Exemplo: corpo robot



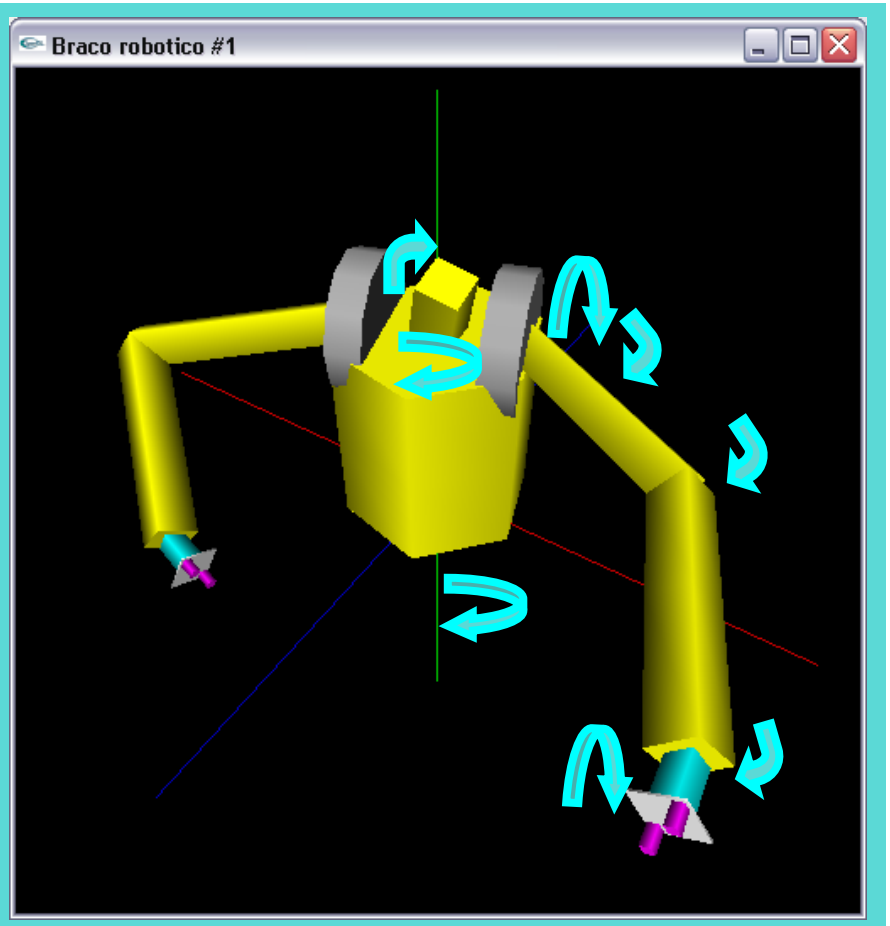
- Tronco
  - Rotação no plano
- Pescoço
  - Rotação no plano
  - Rotação no eixo
- Dois braços



Classe Robot

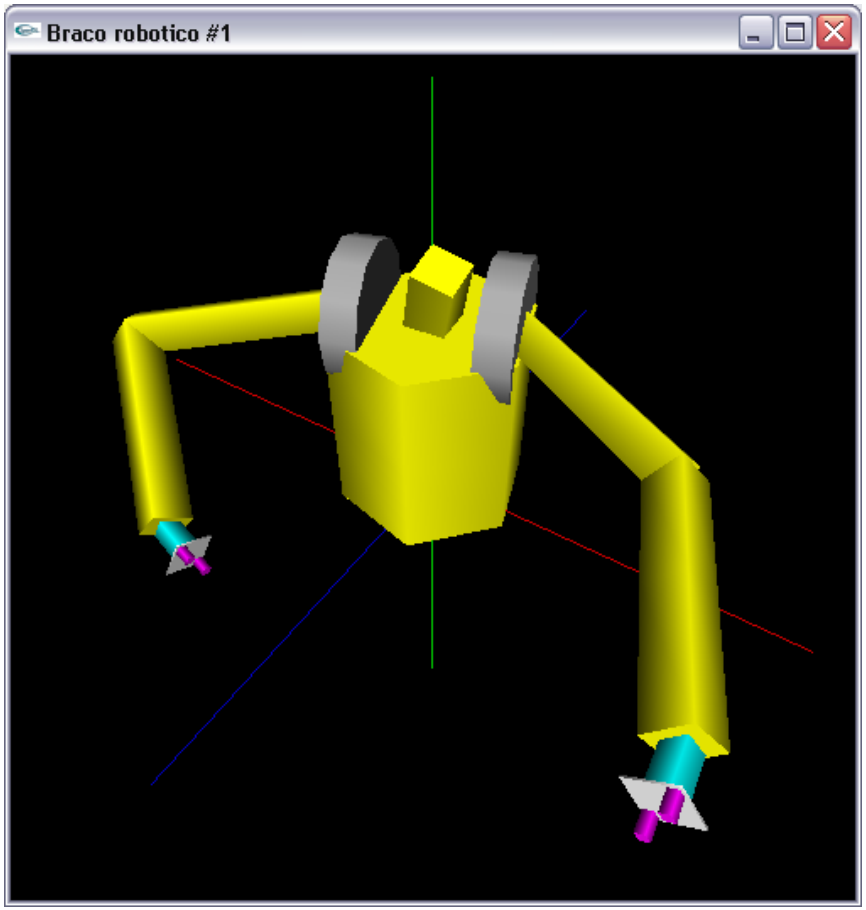


```
class Robot
{
   GLfloat rotTorso;
   GLfloat rotKneck;
   GLfloat rotHead;

   RobotArm left;
   RobotArm right;

   ...
}
```

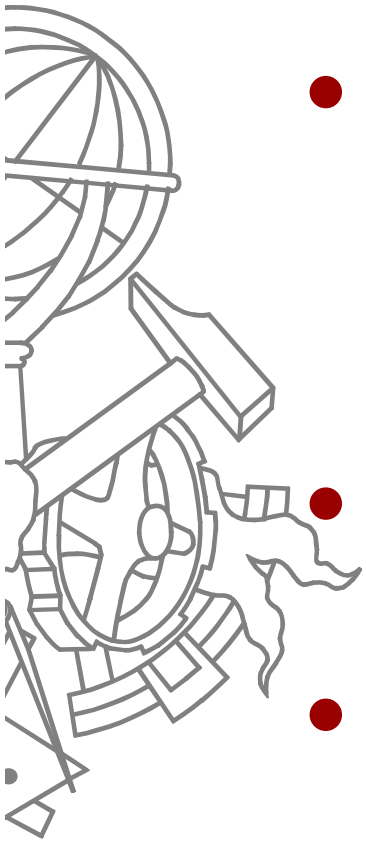


```
glPushMatrix();
      // torso
      glRotatef(rotTorso, 0, 0, 1);
      cylinderWithTopAndBottom(mode, TORSO_WIDTH, TORSO_HEIGHT, 6, 2);

      glPushMatrix();
               // kneck
               glTranslatef(0, 0, TORSO_HEIGHT);
               glRotatef(rotKneck, 0, 0, 1);

               // head
               glRotatef(rotHead, 0, 1, 0);
               box(mode, HEAD_WIDTH, HEAD_HEIGTH);
      glPopMatrix();

      // left arm
      glPushMatrix();
               glTranslatef(-TORSO_WIDTH/2, 0, TORSO_HEIGHT);
               glRotatef(-90, 0, 1, 0);
               left.Display(mode, showAxis);
      glPopMatrix();

      // rigth arm
      glPushMatrix();
               glTranslatef(+TORSO_WIDTH/2, 0, TORSO_HEIGHT);
               glRotatef(+90, 0, 1, 0);
               right.Display(mode, showAxis);
      glPopMatrix();
glPopMatrix();
```

Demo



Implementação genérica



- Utilizar uma estrutura em árvore para descrever os modelos
  - Nós do tipo transformação ou elemento
  - Polimorfismo OO
- Rotina genérica para percorrer a árvore e desenhar o objecto
- Ler modelo de ficheiro

Exemplo

```
class No {
   List<No> descendentes;
   void virtual Draw();
}
class Transformacao : No { ... }
class Translacao : Transformacao { ... }
class Rotacao : Transformacao { ... }
class Escala : Transformacao { ... }
class Elemento : No { ... }
class Esfera : Elemento { ... }
class Cubo : Elemento { ... }
class Paralelepipedo : Elemento { ... }
```