

Computação Gráfica e Interfaces (LEIC) Sistemas Gráficos (LEEC)

Enunciados de exercícios para as aulas práticas

4. Transformações Geométricas

O objectivo deste grupo de exercícios é aplicar os conceitos de Transformações Geométricas, no contexto da tecnologia OpenGL.

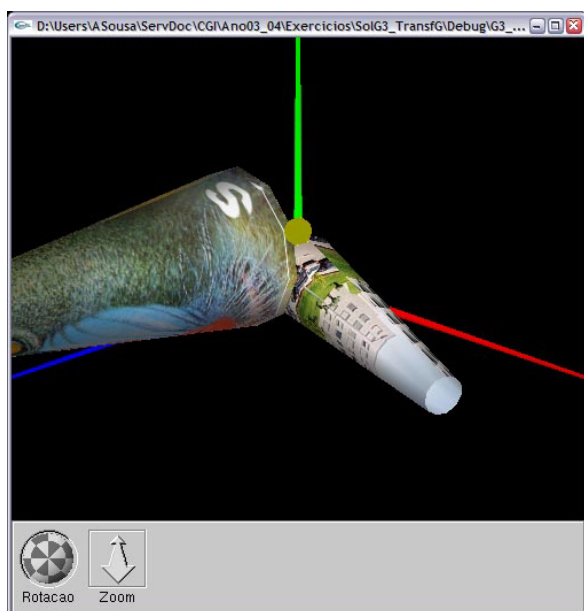


Fig. 1 - Cena 3D em OpenGL

A cena 3D da Fig. 1 constitui-se de, além dos três eixos e da fonte de luz (com respectiva esfera simbólica), de dois cilindros (cones) ocios, gerados com a instrução `gluCylinder()`. O cilindro 2 é o que se cobre com a textura da imagem do mandril e é fixo; o cilindro 1 é o que se cobre com a textura da imagem da FEUP e é animado.

O centro da maior base do cilindro 1 encontra-se nas coordenadas $(0, 2, 3)$ e o seu eixo é paralelo ao eixo dos Z . O centro da maior base do cilindro 2 coincide, inicialmente, com a origem das coordenadas e o seu eixo coincide com o eixo dos Z .

Pretende-se animar o cilindro 1 para que ele faça uma rotação completa sobre o eixo dos Y , ficando coincidente com o cilindro 2. A animação deve assim compor-se de translações, rotações e escalamentos.

No arquivo `G3_TransfG` é fornecido um conjunto de ficheiros com parte do código para a criação daquela cena. Na zona de declarações globais do programa encontram-se as principais grandezas utilizadas, nomeadamente as dimensões, coordenadas e transformações geométricas dos dois cilindros. Existe uma função `void myUpdateTransforms(int dummy)` que é chamada regularmente em intervalos de `mili_secs` e que é responsável por actualizar os valores das transformações geométricas.

1. A posição actual do cilindro 2 não é a desejada. Corrija-a.
2. Complete o conteúdo da função `myUpdateTransforms(int dummy)`, de forma a que o cilindro 1
 - a) Faça uma rotação completa em volta do eixo dos Y .
 - b) Suba em Y , para que o seu eixo fique coincidente com o eixo do cilindro 2.
 - c) Seja escalado até ficar com as mesmas dimensões do cilindro 2.
 - d) Seja deslocado até ficar coincidente com o cilindro 2.
3. Acrescente, na função `display()`, as chamadas necessárias a funções de transformações geométricas para que as alíneas anteriores fiquem completas (esta questão pode, eventualmente, ser concretizada em simultâneo com a questão anterior).
4. Verifique a necessidade (ou não) da existência das chamadas a `glPushMatrix()` e `glPopMatrix()` que se encontram comentadas na função `display()` entre os desenhos dos dois cilindros.