

# INF01047 - Fundamentos de Computação Gráfica

## Prática 3 - Visualização 3D

### Objetivos:

Exercitar o uso de funções de manipulação de câmera e visualização 3D (diferentes projeções) em OpenGL .

**Código fonte para download:** [camera.cpp](#)

Esta aula prática reforça os conceitos de visualização 3D, usando OpenGL.

A **especificação da câmera** e a **escolha da projeção** são exemplificadas abaixo, considerando projeção paralela ortográfica:

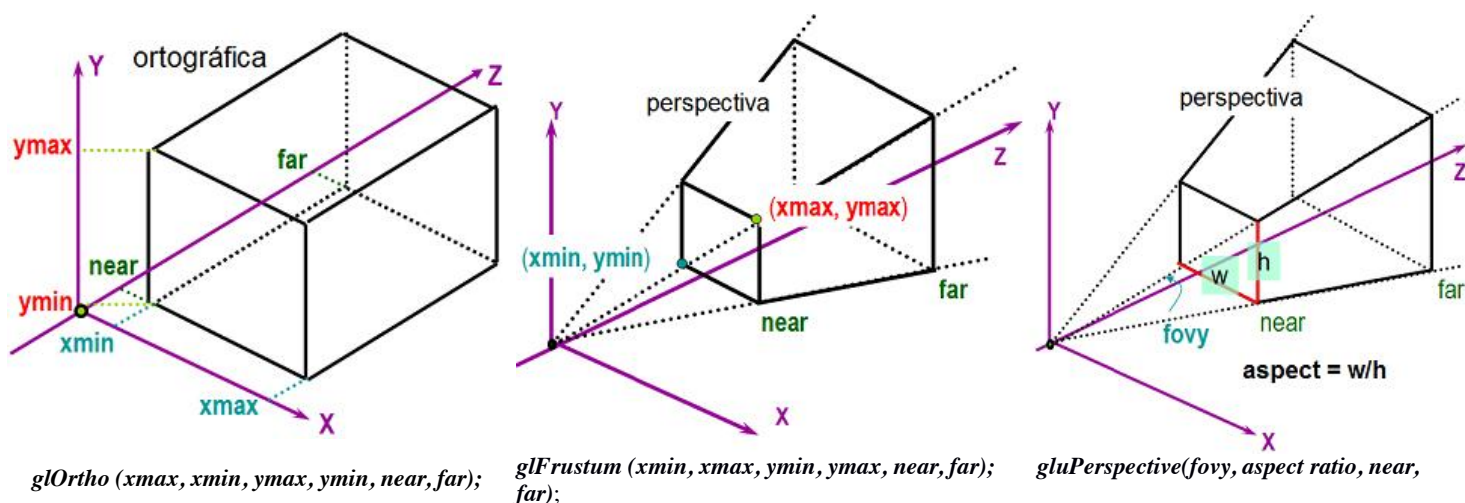
```
....
glMatrixMode(GL_PROJECTION); // indica que será alterada a matriz que contém os parâmetros de projeção
glLoadIdentity(); // carrega a matriz de projeção com a identidade
glOrtho(left, right, bottom, top, near, far); // instrui parâmetros de projeção paralela ortográfica, com valores relativos à posição da câmera.
// near é o plano de projeção que será posteriormente mapeado para a imagem na janela
glMatrixMode(GL_MODELVIEW); // indica que as próximas chamadas afetarão a matriz de modelagem-visualização (câmera e
transformações geométricas)
glLoadIdentity(); // carrega a matriz model_view com a matriz identidade
gluLookAt (Camera1.atx,Camera1.aty,Camera1.atz,Camera1.tox,Camera1.toy,Camera1.toz, 0,1,0); // indica a posição, orientação e vertical
da câmera
....
```

### Exercício 1:

Com o programa **Camera** monte um projeto.

1. Identifique no código fonte (e descreva) a localização da câmera em relação ao objeto, confrontando com o resultado da visualização. **Veja que os eixos estão definidos como vermelho (R) para o eixo X, verde (G) para o Y e azul para Z.**

Neste programa, usamos as duas formas de projeção: ortográfica e perspectiva. Para especificar essas projeções em OpenGL, usamos funções que nos permitem especificar os volumes de visualização.



**Identifique, no programa, os tipos de projeção previstos e respectivos parâmetros iniciais, e o tamanho e a localização do objeto no SRU. Estabeleça a relação entre o objeto no SRU, a câmera e volume de visualização, também no SRU.**

2. Exercite a ampliação dos limites dos planos near e far, para efeito de recorte em profundidade. Por exemplo, coloque near = 120.0, e veja o efeito na imagem final. Explique o que aconteceu. Lembre que esses parâmetros são DISTÂNCIAS do ponto onde está localizada a câmera.

4. Observe a implementação de zoom in (visão de detalhes, "aproximação") e zoom out (visão geral, "afastamento"). Explique a causa desse efeito no caso da projeção ortográfica. O que ocorre se você aciona várias vezes o botão esquerdo do mouse, para obter zoom in? Explique.

5. Com os parâmetros de projeção de volta aos originais, altere a posição da câmera, mudando um parâmetro de cada vez, para ter controle sobre

as alterações. Também convém ressaltar que os parâmetros da função **glOrtho ( )** são relativos à câmera. **Observe e explique o que ocorre se as variáveis minX e maxX não são simétricas (por exemplo, minxX=10, maxX =110).**

## Exercício 2:

Com o programa agora gerando projeção **perspectiva** (para isso, mude o **#DEFINE** no início do programa).

1. Observe o efeito de mudança dos parâmetros da perspectiva e da câmera, alterando um de cada vez e descrevendo o resultado. Altere, inclusive, os planos **near** e **far**. Explique o que aconteceu.

2. Com os parâmetros de volta aos originais, reconfigurar a câmera (parâmetros "at", "to") para gerar imagem do objeto:

- do lado esquerdo, a 45 graus;
- de trás, a 45 graus;
- de cima;
- de baixo.

3. Em cada um dos casos, observe o enquadramento do objeto no volume e o efeito da função de *zoom in* e *zoom out*.

4. O que acontece se você altera o parâmetro vertical da câmera?

## Exercício 3:

Escolha uma das formas de projeção e implemente a exibição dos seguintes objetos, nas situações especificadas. Buscar os parâmetros das funções na documentação da GLUT.

1. O teapot de tamanho 20, localizado na posição (10,10,10), rotacionado 30° em relação à sua orientação original.

2. Um cone localizado na origem do sistema, com base 10 e altura 50.

Observe que os objetos devem ser independentes, ou seja, as transformações geométricas aplicadas a um deles não devem afetar o outro. Para testar se tudo está correto, **mude o foco para qualquer outra janela da tela e retorne à janela OpenGL: se a imagem não é mais a originalmente planejada, explique porque.**

## Entrega desta prática:

Você NAO PRECISA entregar os resultados, MAS os exercicios ajudam o trabalho final...

*Prática elaborada por Carla Freitas e Luciana Nedel. Modificada por Marcelo Walter (outubro de 2010) e Carla Freitas (maio de 2011; outubro de 2012; setembro de 2014)*