

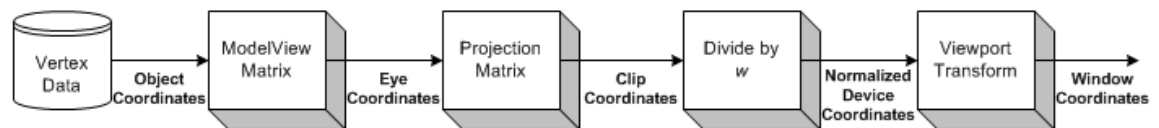
## P03. PROJEÇÕES – VISORES (VIEWPORTS)

### SUMÁRIO

Matrizes no OpenGL	16
Utilização de projeções	16
Utilização de visores ( <i>viewports</i> )	18

### Matrizes no OpenGL

Transformação da informação geométrica (vértices, normais, ...) no *pipeline* do OpenGL



OpenGL utiliza matrizes [4 x 4] para as transformações

OpenGL tem 4 tipos diferentes de matrizes:

`GL_MODELVIEW`, `GL_PROJECTION`, `GL_TEXTURE` e `GL_COLOR`

Mudança do tipo: `glMatrixMode()`

`glMatrixMode(GL_MODELVIEW)`

OpenGL mantém uma pilha que manipula as matrizes correspondentes a cada um dos modos

`MODELVIEW`, `PROJECTION`, `TEXTURE` e `COLOR`

Matriz atual (topo da pilha) tem o produto cumulativo das multiplicações das várias matrizes existentes na pilha

Normalmente, `MODELVIEW` tem uma profundidade de 32 matrizes

`glGetIntegerv (GL_MAX_MODELVIEW_STACK_DEPTH, stackSize)`

Retorna um inteiro com o tamanho da pilha

Normalmente, os outros modos têm uma profundidade de 2

`glGetIntegerv (GL_MAX_PROJECTION_STACK_DEPTH, stackSize)`

`glGetIntegerv (GL_MAX_TEXTURE_STACK_DEPTH, stackSize)`

`glGetIntegerv (GL_MAX_COLOR_STACK_DEPTH, stackSize)`

Retorno do número de matrizes que estão atualmente na pilha

`glGetIntegerv (GL_MODELVIEW_STACK_DEPTH, numMats)`

### Utilização de projeções

Geração de imagens em R2 é tratada pelo OpenGL como um caso especial de visualização em R3

Mesmo para desenhar uma linha em R2, o OpenGL processa as operações de visualização em R3

Necessidade de definir o tipo de projeção

Paralela Ortogonal

`glMatrixMode (GL_PROJECTION);`

`gluOrtho2D (0.0 , 200.0 , 0.0 , 150.0)`

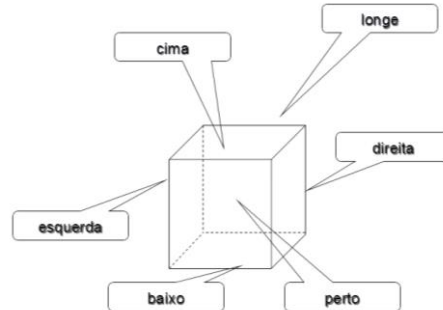
Projeção do conteúdo de uma janela (WCS)

$$0.0 \leq x \leq 200.0$$

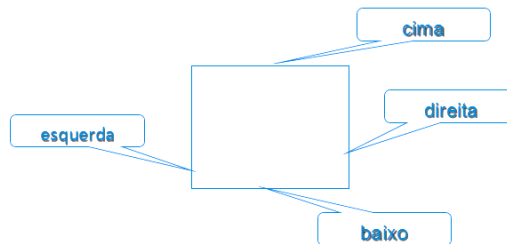
$$0.0 \leq y \leq 150$$

**Projeção paralela ortogonal**

**glOrtho** (esquerda, direita, baixo, cima, perto, longe)



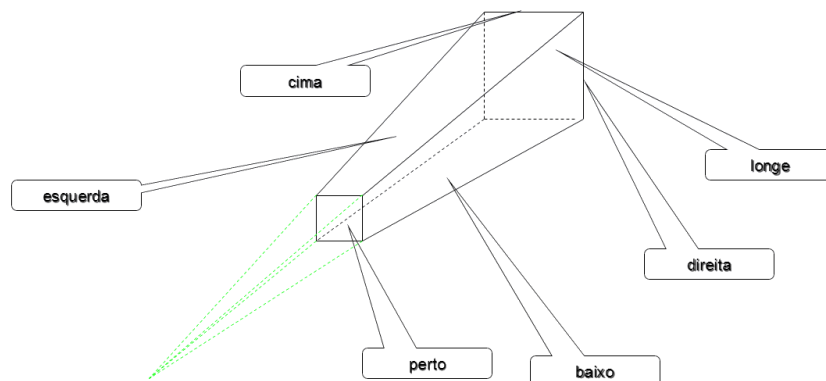
**gluOrtho2D** (esquerda, direita, baixo, cima)



**Projeção perspectiva**

**glFrustum** (esquerda, direita, baixo, cima, perto, longe)

**gluPerspective** (ângulo, distorção, perto, longe)



Exemplo de uso da função gluOrtho2D

```
void display( void )
{
    glClear( GL_COLOR_BUFFER_BIT );
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluOrtho2D (0.0, 500.0, 0.0, 500.0); ... }
```

## Utilização de visores (*viewports*)

Utilização das funções **gluOrtho2D** e **glViewport**

Especificação do retângulo de recorte

**gluOrtho2D** (xesq, xdir, ybaixo, ycima)

Definida na função **display** ou numa função de inicialização

Controlar a posição da informação na janela

**glViewport** (xmin, ymin, largura, altura)

Definida na função ***display***, antes de desenhar

Exemplo de uso da função **gluOrtho2D** e **glViewport**, com função **Init**

```
void display( void )
{
    glClear( GL_COLOR_BUFFER_BIT );
    glViewport(50.0, 50.0, 200.0, 200.0);
    glBegin(GL_POLYGON);
        .... ;
    ...
}
```