

## Programação OpenGL

### Funções de visualização

Carla Freitas - UFRGS - 2001

## Tipos de dados em OpenGL

- GLbyte = inteiro de 8 bits
- GLshort = inteiro de 16 bits
- GLint, GLsizei = inteiro de 32 bits
- GLfloat = float de 32 bits
- GLdouble = float de 64 bits
- GLubyte, GLboolean = inteiro s/sinal de 8 bits
- GLushort = inteiro s/sinal de 16 bits
- GLuint, GLenum = inteiro s/sinal de 32 bits

Carla Freitas - UFRGS - 2001

## Definição da janela OpenGL

- A janela para exibição das imagens produzidas pelas chamadas das funções OpenGL é uma das janelas do sistema de janelas
- A interface entre a biblioteca gráfica e o sistema de janelas/sistema operacional pode ser complexa.

Carla Freitas - UFRGS - 2001

## Funções de controle para visualização

- A GLUT é uma biblioteca de funções para interface entre a GL e o sistema de janelas, fornecendo, portanto, independência de plataforma.
- Outra alternativa é usar as funções do sistema base (Windows GDI, p. exemplo)

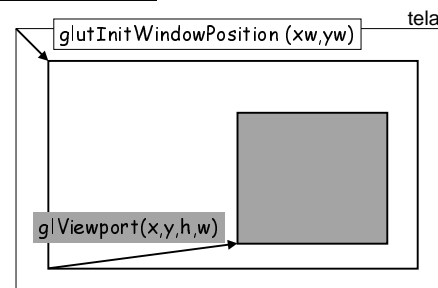
Carla Freitas - UFRGS - 2001

## Inicialização da janela

- Janelas possuem como atributos largura e altura, as quais são medidas em termos de coordenadas de janela.
- Janelas em OpenGL tem a origem no canto inferior esquerdo.
- Os sistemas de janelas devolvem posições do mouse relativas ao canto superior esquerdo

Carla Freitas - UFRGS - 2001

## Relação *window*, *viewport* e *tela*



Carla Freitas - UFRGS - 2001

## Inicialização de janela

```
glutInitWindowSize (480, 640);  
glutInitWindowPosition (0,0);  
- janela de 480 x 640 pixels,  
- no canto superior esquerdo da tela  
id = glutCreateWindow ("título");  
- criação efetiva da janela com determinado  
título  
- devolve o identificador da janela
```

Carla Freitas - UFRGS - 2001

## Inicialização da janela

- `glutInitDisplayMode (GLUT_RGB | GLUT_SINGLE);`  
especifica que a
  - cor será dada em RGB,
  - será utilizado um único buffer de imagem
- `glutSetWindow (id)`  
- torna "id" a janela corrente

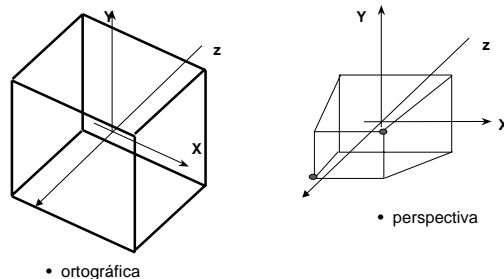
Carla Freitas - UFRGS - 2001

## Inicialização de janela

```
glViewport (GLint x, GLint y, GLsizei w,  
GLsizei h)  
- determina uma área dentro da window  
- x, y, w e h em pixels  
w/h = razão de aspecto
```

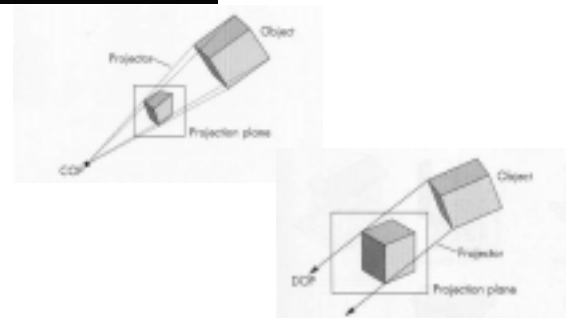
Carla Freitas - UFRGS - 2001

## Especificando a região de interesse no universo



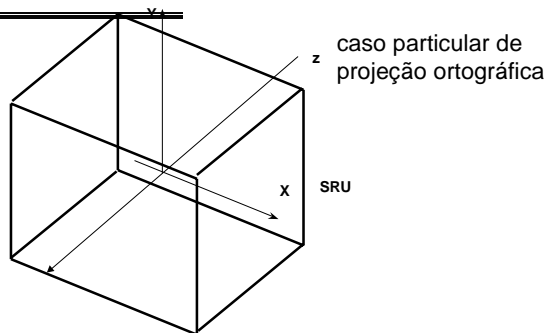
Carla Freitas - UFRGS - 2001

## Projeção cônica x paralela



Carla Freitas - UFRGS - 2001

## Visualização 2D



Carla Freitas - UFRGS - 2001

## Visualização (2D) em OpenGL

- Utilizando a GLU
  - gluOrtho2D (left, right, bottom, top);
    - todos os parâmetros são *double*
- Esta função chama
  - glOrtho (left, right, bottom, top, near, far);
    - near = -1.0 e far = 1.0
    - especificando limites do volume no SRU
- Não se especifica posição de observador

Carla Freitas - UFRGS - 2001

## Visualização: matriz de projeção

- A especificação de visualização inicializa uma matriz interna de transformação, a matriz de projeção
- No caso de OpenGL, é necessário indicar explicitamente a matriz:
  - glMatrixMode (GL\_PROJECTION);
  - glLoadIdentity();
  - glOrtho (...);

Carla Freitas - UFRGS - 2001

## Funções de controle de visualização

- É necessário especificar
  - que função do programa é responsável pelo desenho na janela
    - quando a janela permanecer oculta por certo tempo e precisar ser redesenhada
  - que função trata a alteração de tamanho da janela
    - adequação da *viewport*
    - redesenho das primitivas geométricas

Carla Freitas - UFRGS - 2001

## Controle de visualização

- glutDisplayFunc (funcao-de-desenho);
  - especifica a função *callback* para a janela corrente
- glutReshapeFunc (funcao,largura,altura);
  - especifica a função *callback* de redimensionamento para a janela corrente

Carla Freitas - UFRGS - 2001

## Controle de visualização

- glutMainLoop();
  - inicia o loop principal de espera e tratamento de eventos
- glutPostRedisplay ();
  - determina que a janela corrente deve ser redesenhada
- glFlush();
  - determina a espera pelo término da execução de todos os comandos OpenGL

Carla Freitas - UFRGS - 2001

## Funções OpenGL (pontos)

- Controle de atributos
  - glPointSize (size) /\* diâmetro em pixels \*/
- Obtendo tamanhos disponíveis
  - glFloat sizes [2];
  - glFloat step;
  - glGetFloatv (GL\_PONT\_SIZE\_RANGE, sizes);
  - glGetFloatv (GL\_PONT\_SIZE\_GRANULARITY, &step);
    - o tamanho do ponto em geral fica entre 0.5 e 10
    - o intervalo na implementação Microsoft é 0.125
- glBegin (GL\_POINTS); .... glEnd();

Carla Freitas - UFRGS - 2001

## Funções OpenGL (linhas)

- Controle de atributos
  - `glLineWidth (width) /* largura em pixels */`
  - `glLineStipple (fator, padrão) /* padrão da linha */`
  - `glEnable (GL_LINE_STIPPLE);`
- Obtendo tamanhos disponíveis
  - `GLfloat sizes [2];`
  - `GLfloat step;`
  - `glGetFloatv (GL_LINE_WIDTH_RANGE, sizes);`
  - `glGetFloatv (GL_LINE_WIDTH_GRANULARITY, &step);`
    - a largura da linha em geral fica entre 0.5 e 10
    - o intervalo na implementação Microsoft é 0.125

Carla Freitas - UFRGS - 2001

## Funções OpenGL (linhas)

- Especificando padrão
  - `GLint fator;`
  - `GLushort padrão;`
  - o padrão indica a configuração do segmento de reta com uma correspondência bit-pixel
  - o fator amplia o número de pixels que devem obedecer a configuração de um bit
    - padrão = 0X00FF indica tracejado 8 pixels ligados e 8 desligados (o padrão começa com o menos significativo)
    - um fator de 5 indicaria 5 pixels correspondendo a 1 bit

Carla Freitas - UFRGS - 2001

## Funções OpenGL (cor)

- Especificando cor
  - `glColor3f (c_vermelho, c_verde, c_azul);`
  - `glColor4f (c_vermelho, c_verde, c_azul, c_alpha);`
  - os componentes vermelho, verde e azul podem ser especificados em float ou qualquer outro tipo numérico (substitui-se o *f* na chamada)
  - o componente alpha corresponde à transparência da cor

Carla Freitas - UFRGS - 2001

## Funções primitivas geométricas

- Baseadas em vértices
  - `glBegin (GL_LINES); ..... glEnd();`
- Outras
  - `glRect (x1,y1,x2,y2)` e variações
  - `glutSolidCone, glutWireCone`
    - cone orientado ao longo do eixo z
    - raio da base, altura, nro de divisões verticais e transversais
  - `glutSolidCube, glutWireCube (tamanho)`
  - `glutSolidTeapot, glutWireTeapot (tamanho) /* raio`
  - ...

Carla Freitas - UFRGS - 2001

## Transformações geométricas

- Compostas na matriz de modelagem
  - `glRotatef`
  - `glTranslatef`
  - `glScalef`
- A matriz de modelagem também representa alterações na posição do observador

Carla Freitas - UFRGS - 2001

## Transformações na matriz MODELVIEW

- `glMatrixMode (GL_MODELVIEW);`
- `glLoadIdentity();`
- ... /\* transformações geométricas \*/
- Estas transformações são aplicadas aos vértices antes da projeção

Carla Freitas - UFRGS - 2001

## Controlando a matriz de transformação

- `glPushMatrix ();`
  - empilha a matriz de transformação corrente
- `glPopMatrix ();`
  - desempilha a matriz no topo da pilha tornando-a a matriz corrente

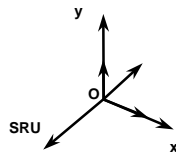
Carla Freitas - UFRGS - 2001

## Exemplo e exercícios

Carla Freitas - UFRGS - 2001

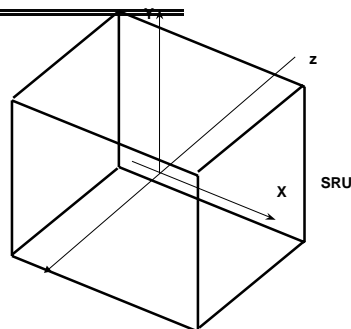
## Visualização 3D em OpenGL

- Posição default da câmera
  - na origem (0,0,0)
  - orientada para o eixo -z
- Volume default para projeção ortográfica
  - 2 x 2 x 2
  - objetos atrás do observador também são projetados



Carla Freitas - UFRGS - 2001

## Visualização 3D



Carla Freitas - UFRGS - 2001

## Visualização 3D em OpenGL

- Projeção paralela ortográfica
  - ...
  - `glMatrixMode (GL_PROJECTION);`
  - `glLoadIdentity();`
  - `gluOrtho (left, right, bottom, top, near, far);`
  - `glMatrixMode (GL_MODELVIEW);`
  - ...

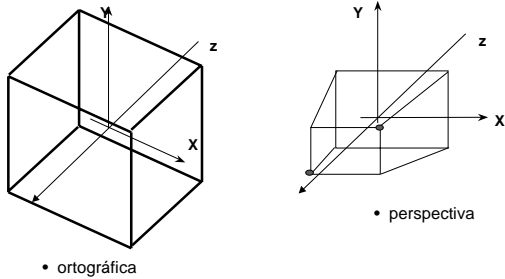
Carla Freitas - UFRGS - 2001

## Visualização 3D: posicionamento da câmera

- Função da GLU
  - indicação da posição do observador, do ponto-alvo e de um vetor que indica a "vertical" da câmera
  - `glMatrixMode (GL_MODELVIEW);`
  - `gluLookAt (px,py,pz, ax,ay,az, vx,vy,vz);`  
/\* observador, alvo, vertical \*/
  - `glMatrixMode (GL_PROJECTION);`
  - `glOrtho (...);`
  - ...

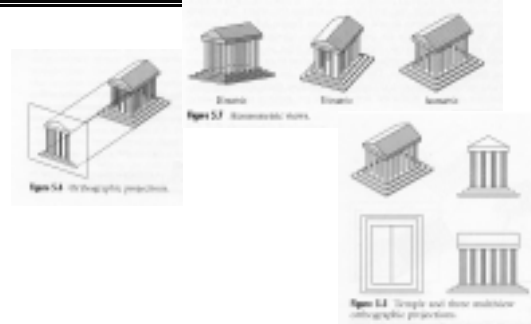
Carla Freitas - UFRGS - 2001

## Especificando a região de interesse no universo



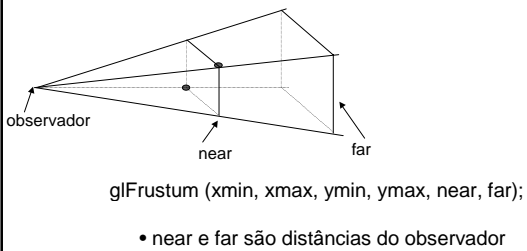
Carla Freitas - UFRGS - 2001

## Projeções paralelas ortográficas



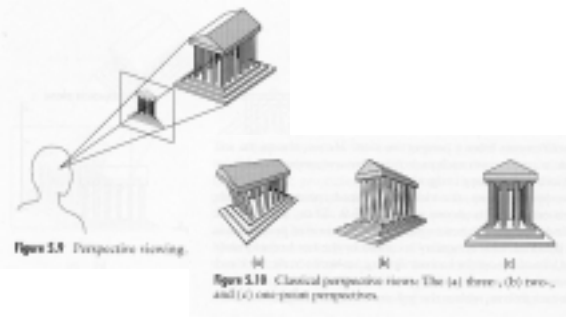
Carla Freitas - UFRGS - 2001

## Perspectiva



Carla Freitas - UFRGS - 2001

## Projeções em perspectiva



Carla Freitas - UFRGS - 2001

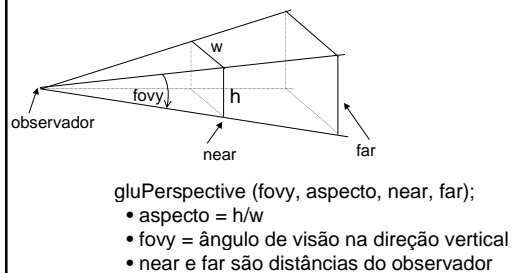
## Perspectiva em OpenGL

### ■ Perspectiva

```
glMatrixMode (GL_PROJECTION);
glLoadIdentity();
glFrustum (xmin, xmax, ymin, ymax, near, far);
...
glMatrixMode (GL_MODELVIEW);
glLoadIdentity();
```

Carla Freitas - UFRGS - 2001

## Perspectiva



Carla Freitas - UFRGS - 2001

## Perspectiva em OpenGL

- Forma de especificação mais intuitiva:  
glMatrixMode (GL\_PROJECTION);  
glLoadIdentity();  
gluPerspective (fovy, aspect, near, far);  
...  
– o volume de visualização considera o observador como centro de projeção, na origem

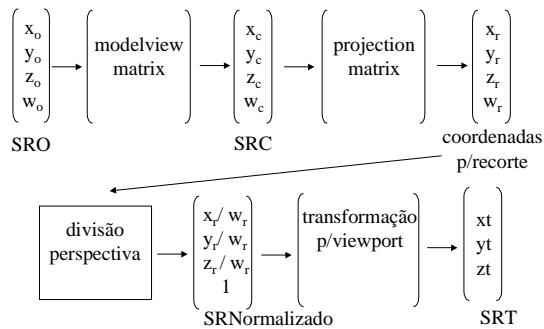
Carla Freitas - UFRGS - 2001

## Visualização 3D em OpenGL

- A especificação de volume de visualização perspectiva inicializa a matriz de projeção
  - Esta matriz, aplicada aos vértices, transforma-os de modo que possa ser aplicado o recorte contra o volume
- Após o recorte, é realizada a projeção efetivamente
  - divisão perspectiva, que produz coordenadas normalizadas
- A transformação para viewport também é expressa internamente por matriz

Carla Freitas - UFRGS - 2001

## Seqüência de transformações em OpenGL



Carla Freitas - UFRGS - 2001

## Exemplo e exercícios

Carla Freitas - UFRGS - 2001