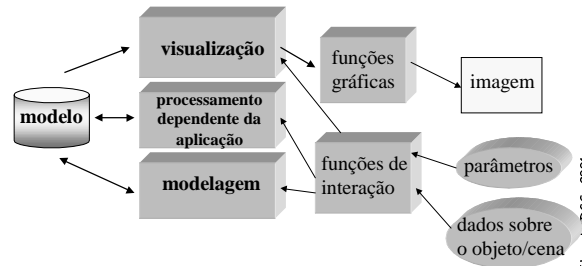


Visualização 2D e 3D

Conceitos e introdução à programação com OpenGL

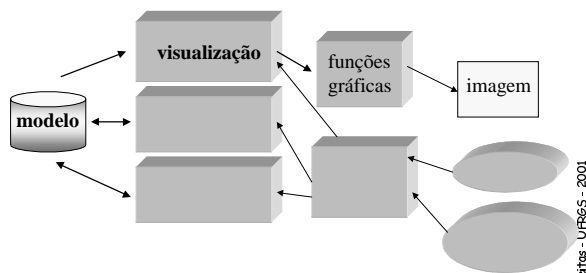
Carla Freitas - UFRGS - 2001

Estrutura geral de sistemas gráficos



Carla Freitas - UFRGS - 2001

Modelo clássico de visualização em CG



Carla Freitas - UFRGS - 2001

Modelo clássico

- As figuras são produzidas vetorialmente, envolvendo pontos e outras primitivas geométricas
- As primitivas que compõem as figuras são descritas num sistema de coordenadas universo
- A biblioteca gráfica funciona como uma caixa preta, convertendo as posições para o sistema de coordenadas de dispositivo

Carla Freitas - UFRGS - 2001

Programação gráfica

- Uso de um conjunto de funções para gerar imagens de objetos cujos modelos estão representados numa estrutura de dados específica da aplicação
 - objetos 2D e 3D requerem tratamento diferenciado em certos momentos
 - OpenGL é uma biblioteca gráfica que oferece funções para criação de imagens de objetos 2D e 3D

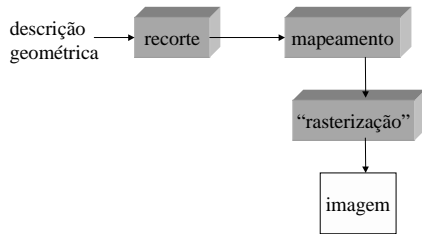
Carla Freitas - UFRGS - 2001

Visualização 2D

- Para objetos 2D, uma biblioteca gráfica deve oferecer funções de:
 - especificação da janela no universo
 - *viewing rectangle*, *clipping rectangle*
 - especificação da *viewport* na tela
 - em termos da janela de tela definida através pelo sistema de janelas

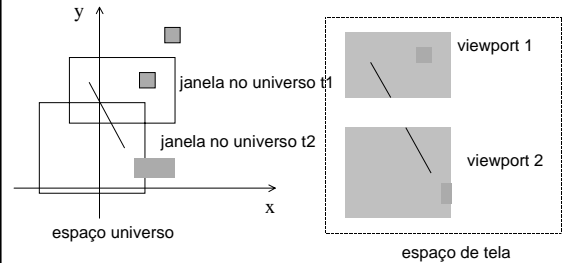
Carla Freitas - UFRGS - 2001

Pipeline de visualização 2D



Carla Freitas - UFRGS - 2001

Visualização de objetos 2D

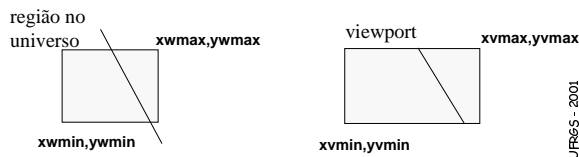


Carla Freitas - UFRGS - 2001

Exibindo uma região do espaço universo

■ Recorte

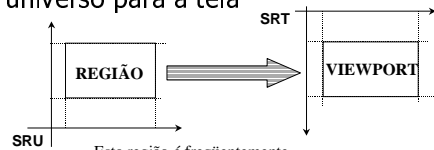
- determinação das partes visíveis dos objetos, segundo a região de interesse no universo corrente



Carla Freitas - UFRGS - 2001

Transformação de visualização

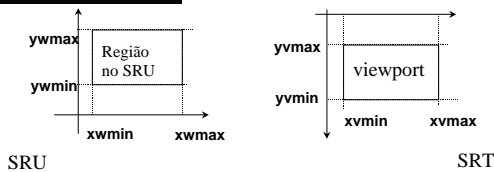
- Corresponde ao mapeamento das coordenadas dos pontos do objeto no universo para a tela



Esta região é frequentemente denominada "window", mas não corresponde a coordenadas de tela.

Carla Freitas - UFRGS - 2001

Transformação de visualização



Carla Freitas - UFRGS - 2001

Mapeamento para a viewport



$$xt = xvmin + (xu - xwmin) * \frac{xvmax - xvmin}{xwmax - xwmin}$$

$$yt = yvmin + (yu - ywmin) * \frac{yvmax - yvmin}{ywmax - ywmin}$$

Carla Freitas - UFRGS - 2001

Visualização 2D usando uma API gráfica

- Especificação da região no universo que contém os objetos de interesse
 - se for API 2D, função tipo "set_window"
 - se for API 3D, função que define volume
- Especificação da viewport, região da tela
- Chamada de funções de controle de atributos
- Chamada de funções de saída (primitivas geométricas)

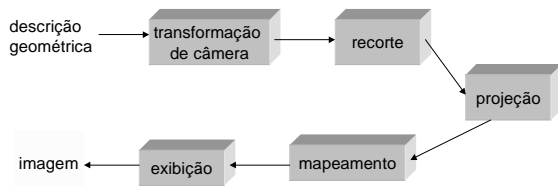
Carla Freitas - UFRGS - 2001

Visualização 3D: câmera sintética

- Corresponde ao *pipeline* clássico de visualização de objetos 3D
- Câmera é localizada e orientada no SRU
- Coordenadas dos objetos são transformadas para o SRC
- Coordenadas são projetadas no plano de projeção
- Coordenadas são mapeadas para a *viewport*

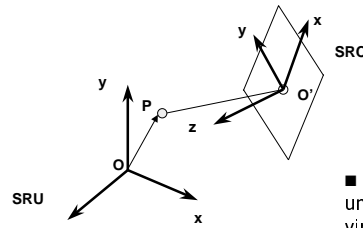
Carla Freitas - UFRGS - 2001

Pipeline de visualização 3D



Carla Freitas - UFRGS - 2001

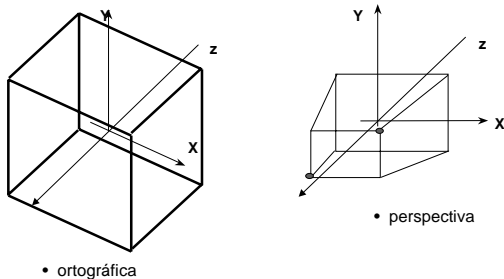
Visualização de objetos 3D



- O observador "carrega" uma câmera fotográfica virtual.
- O plano XY do SRC é o plano de projeção

Carla Freitas - UFRGS - 2001

Visualização 3D: especificando o volume de visualização

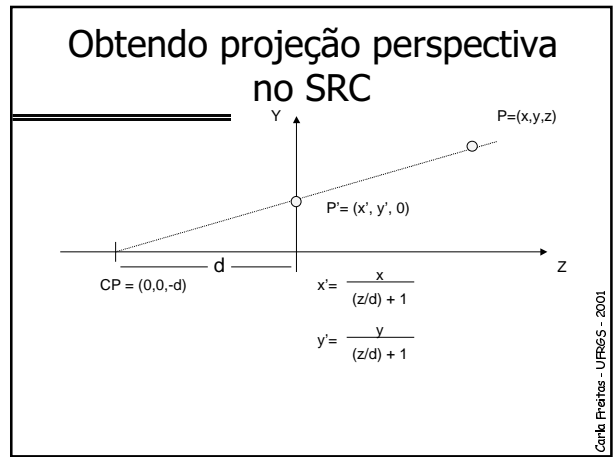
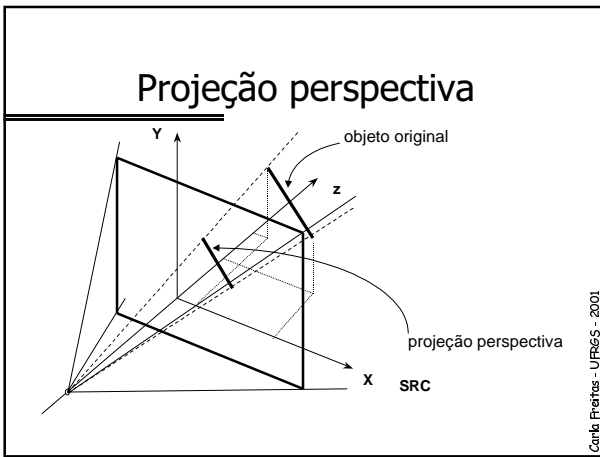
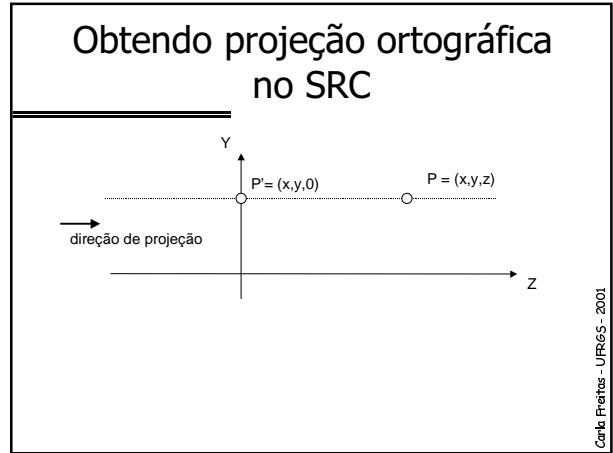
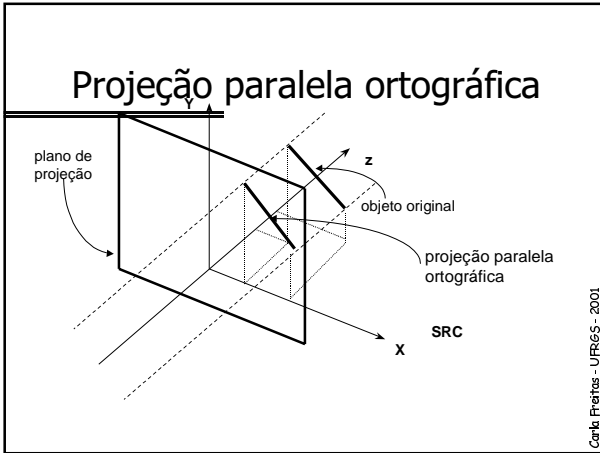


Carla Freitas - UFRGS - 2001

Projeções

- Paralelas
 - oblíquas ou ortográficas
 - é necessário conhecer a direção de projeção
- Perspectiva
 - é necessário conhecer a posição do centro de projeção e o ângulo de visão

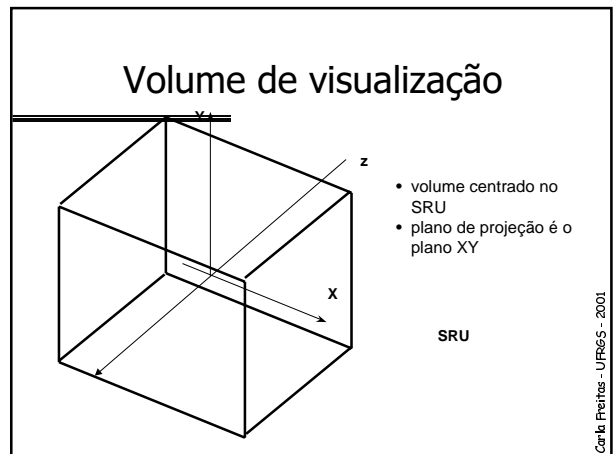
Carla Freitas - UFRGS - 2001



Visualização: 2D como caso especial de 3D

- Utilizando uma biblioteca 3D para visualizar um mundo 2D
 - assumir que o mundo 2D equivale ao plano $z=0$ do sistema de coordenadas 3D
 - definir o volume de visualização
 - definir a projeção como **ortográfica**

Carla Freitas - UFRGS - 2001



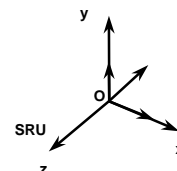
Visualização (2D) em OpenGL

- 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, distâncias do observador no SRU
- Não se especifica posição de observador

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: 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_MODEL_VIEW);
glLoadIdentity();
gluLookAt (px,py,pz, ax,ay,az, vx,vy,vz);
/* observador, alvo, vertical */
```

Carla Freitas - UFRGS - 2001

## Definindo a projeção

- Ortográfica

```
glMatrixMode (GL_PROJECTION);
glLoadIdentity();
glOrtho (left, right, bottom, top, near, far);
glMatrixMode (GL_MODELVIEW);
glLoadIdentity();
```

Carla Freitas - UFRGS - 2001

## Visualização: matrizes

- A especificação de visualização em geral inicializa uma matriz interna das bibliotecas gráficas
- No caso de OpenGL, é necessário indicar explicitamente a matriz:

```
glMatrixMode (GL_PROJECTION);
glLoadIdentity();
gluOrtho (...);
```

Carla Freitas - UFRGS - 2001

## Perspectiva em OpenGL

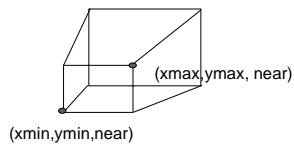
- Perspectiva

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

...
glMatrixMode (GL_MODELVIEW);
glLoadIdentity();
```

Carla Freitas - UFRGS - 2001

## Perspectiva



Carla Freitas - UFRGS - 2001

## Projeção perspectiva

- O centro de projeção (CP) é assumido como localizado na origem
- near e far definem os planos do volume de visualização ao longo do eixo z
  - são especificados como distâncias da origem
  - near é o plano de projeção

Carla Freitas - UFRGS - 2001

## Projeção perspectiva

- Outra forma de especificar:

```
glMatrixMode (GL_PROJECTION);
glLoadIdentity();
gluPerspective (fovy, aspect, near, far);
...
```

  - fovy é o ângulo de visão em y
  - o volume de visualização considera o CP na origem
  - aspect = w/h é a relação entre as dimensões da janela sobre o plano *near*

Carla Freitas - UFRGS - 2001