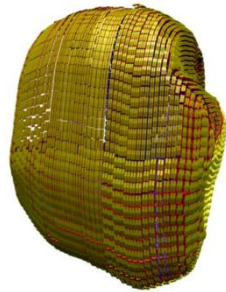
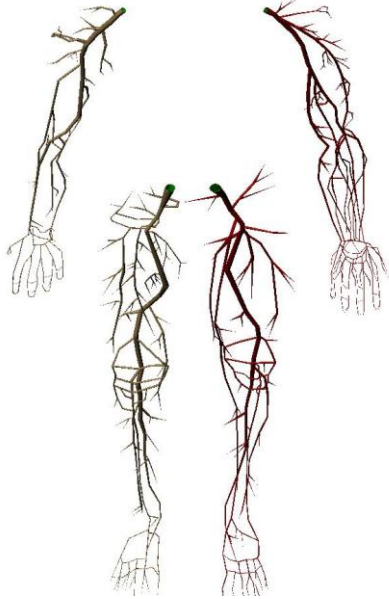




Prof. Dr. Leandro Alves Neves



Sumário

1. Gráficos

■ OpenGL

- Definição
- Pipeline
- Sistema de Coordenadas
- Morfologia das Funções
- Exemplos de Aplicações
- Instalação
- Desenvolvendo Aplicações
 - Estrutura

OpenGL: Definição



■ Bibliotecas:

- ❑ GLU, GLUT, FREEGLUT, GLEW, GLUI...

- Exemplos

- ❑ GLUT (OpenGL Utility Toolkit)
 - ❑ GLU (OpenGL Utility Library)

■ Cabeçalho incluído no programa em C.

```
#include <gl/glut.h>
```

```
#include <gl/glu.h>
```

OpenGL: Definição

- Não é uma Linguagem de Programação
- É um sistema gráfico (API - *Application Programming Interface*) para criação de aplicações gráficas 2D e 3D
 - ❑ Implementada em C, composta por aproximadamente 250 funções e/ou comandos
- Inicialmente: desenvolvida pela *Silicon Graphics (SGI)*

<http://www.opengl.org/>



OpenGL: Definição

- São rotinas de baixo nível
 - Usando a linguagem C
 - `#include <gl/glut.h>` (*Utility Toolkit*)
 - GLUT foi escrita por [Mark J. Kilgard](#)
 - Gerencia a execução da aplicação (I/O) com o S.O.

- Exemplo de função:
 - `glutMainLoop()`

OpenGL: Definição

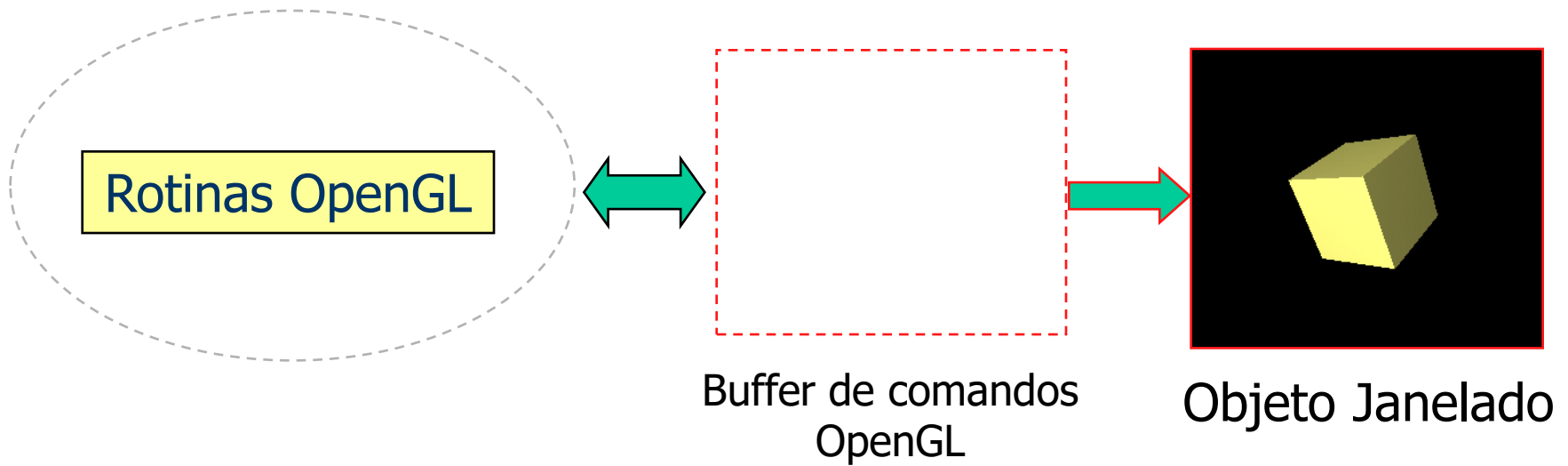
- São rotinas de baixo nível
 - Usando a linguagem C
 - `#include <gl/glu.h>` (*Utility Library*)
 - Mapeamento do sistema de coordenadas e ambiente 2D/3D da aplicação
- Exemplos
 - `gluOrtho2D`, `gluPerspective...`

OpenGL: Pipeline

- **pipeline** simplificado do OpenGL: descreve um processo que pode ter dois ou mais passos distintos



OpenGL: Pipeline



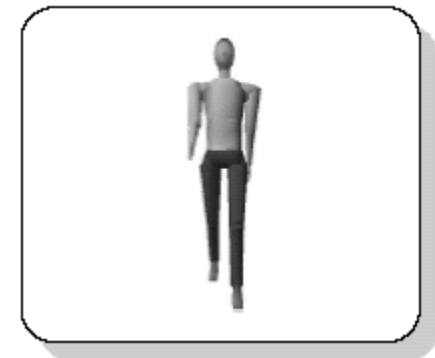
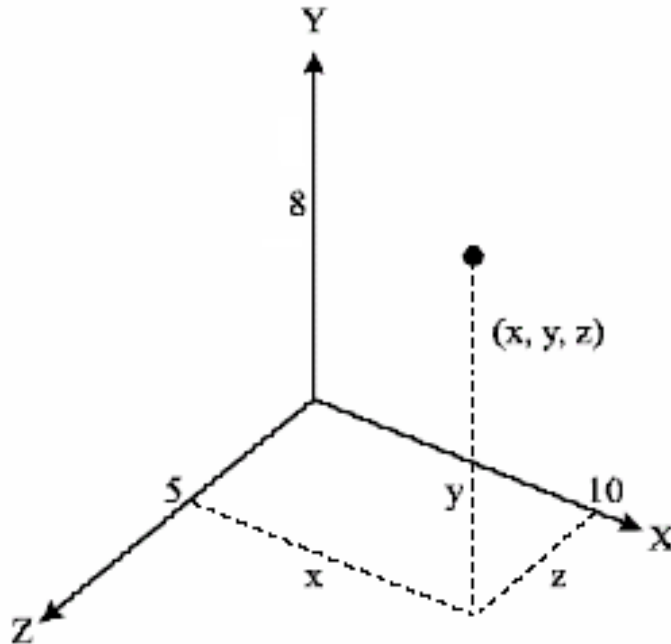
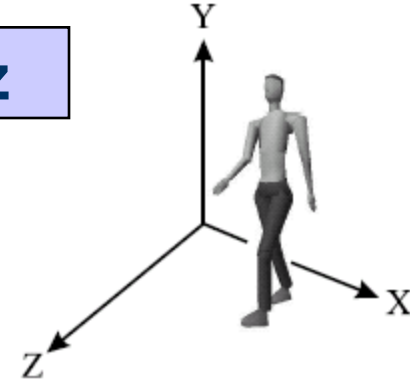
OpenGL: Sistema de coordenadas

```
glVertex3f (10.0, 8.0, 5.0);
```

x


y

z



OpenGL: Morfologia das Funções

- glVertex3f (10.0, 8.0, 5.0);

gl  é a biblioteca;

Vertex  é o nome da função;

3  é quantidade de variáveis (3D);

2  é quantidade de variáveis (2D);

f  é o tipo dessas variáveis (float).

A maioria das funções em OpenGL segue essa morfologia:

**Biblioteca + nome da função + número de variáveis +
tipo de variáveis (arg 1, arg 2, ...arg n);**

OpenGL: Morfologia das Funções

`glVertex3f ()`

*Número de
componentes*

2 - (x,y)
3 - (x,y,z)

Tipo de dado

b - byte
ub - unsigned byte
s - short
us - unsigned short
i - int
ui - unsigned int
f - float
d - double

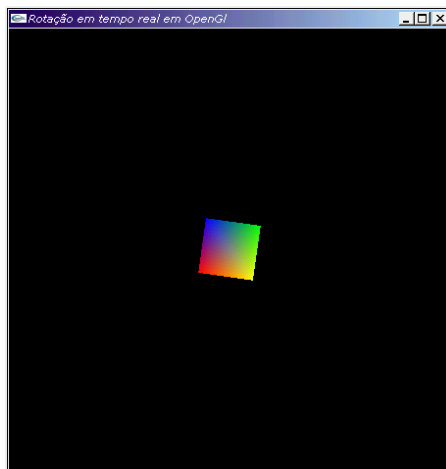
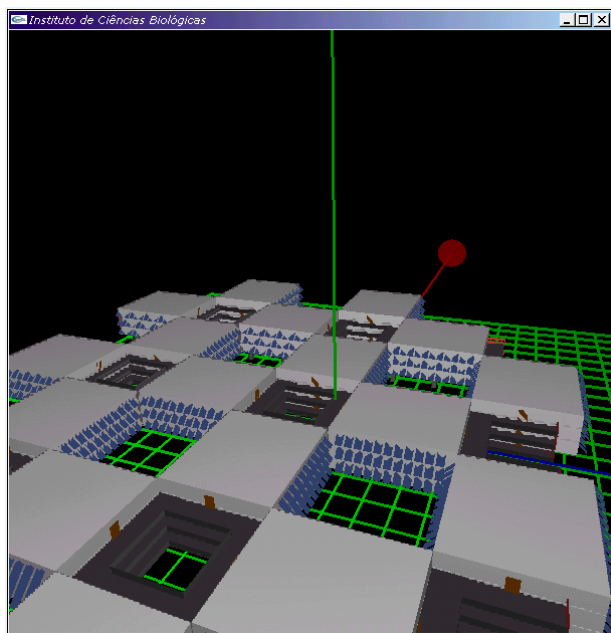
OpenGL: Morfologia das Funções

- Cuidado!!! Algumas exceções.
- Exemplo:

`glClearColor(variáveis)`

OpenGL: Exemplos de Aplicações

- Exemplos de aplicações geradas com a OpenGL.



OpenGL com Linux

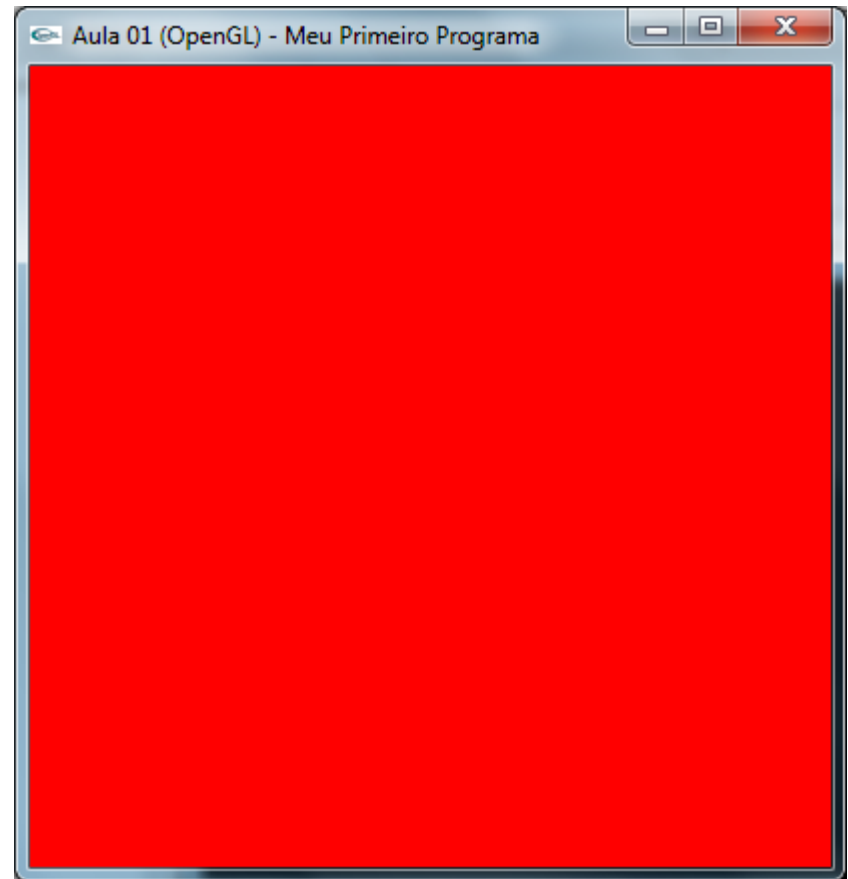
- Compilar código fonte em linguagem C e OpenGL:
 - Instalar o pacote freeglut3, se necessário:
 - # Debian e derivados (ubuntu):
 - *apt-get install freeglut3*
- Compilar o código:
 - *gcc <fonte.c> -o <fonte> -lglut -lGL -lGLU -lm*
- *Executar*
 - *./fonte*

OpenGL: Máquina de Estado?

- OpenGL é uma máquina de estado!
- Determina o estado do sistema:
 - se está rotacionado;
 - a cor para a renderização e etc
 - próximo passo é renderizar
- Consequentemente a cada alteração de estado a cena é novamente renderizada

OpenGL: Criando janelas

- Todo programa que usa OpenGL tem alguns elementos no código fonte que são obrigatórios:



Exemplo de janela de aplicação Windows

OpenGL: Estrutura (A mais básica)

```

9
10 PrimeiroPrograma - Prof. Leandro Alves Neves
11 Um programa OpenGL simples que abre uma janela GLUT
12 Este código está baseado no Simple.c, exemplo
13 disponível no livro "OpenGL SuperBible" e "OpenGL: uma abordagem prática e objetiva" ,
14 2nd Edition, de Richard S. e Wright Jr.*/
15
16 #include <stdlib.h>
17 #include <stdio.h>
18 #include <GL/freeglut.h>
19
20 void DISPLAY ();
21
22 // Inicializa parâmetros de rendering
23 void DISPLAY ()
24 {
25     // Define a cor de fundo da janela de visualização como preta
26     glClearColor(1, 0, 0, 0);
27     glMatrixMode(GL_PROJECTION); //Ativa matriz de projeção
28     glLoadIdentity(); // "Limpa" ou "transforma" a matriz em identidade, reduzindo possíveis erros.
29     gluOrtho2D(-200,200,-200,200); //Define tipo de projeção (2D) e o tamanho
30     glMatrixMode(GL_MODELVIEW); //Ativa matriz de visualização
31     glLoadIdentity(); // "Limpa" ou "transforma" a matriz em identidade, reduzindo possíveis erros.
32     //Limpa a janela de visualização com a cor de fundo especificada
33     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); /* "limpa" um buffer particular ou combinações de buffers,
34                                                         onde buffer é uma área de armazenamento para informações da imagem.
35                                                         Nesse caso, está "limpando os buffers para suportarem animações */
36     //Chamada para Função ou funções para desenhar o objeto/cena...
37     //-----
38     glutSwapBuffers();
39
40 }

```

OpenGL: Estrutura (A mais básica)

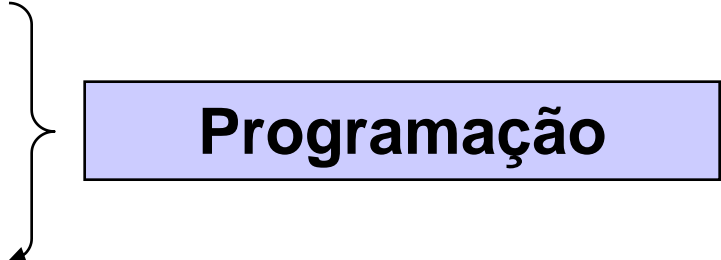
```
44  
45 int main(int argc, char **argv)  
46 {  
47     glutInit(&argc, argv); // Initializes glut  
48  
49     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);  
50     /*Avisa a GLUT que modo de exibição deve ser usado quando a janela é criada.  
51     // Nesse caso, permite: uma única janela; cores compostas por Verm. Verde e Azul; e, */  
52     glutInitWindowSize(400,400);  
53     glutInitWindowPosition(100,100);  
54     glutCreateWindow("Aula 01 (OpenGL) - Meu Primeiro Programa");  
55     glutDisplayFunc(DISPLAY);  
56     glutMainLoop();  
57     return 0;  
58 }
```

Exercícios

Altere os parâmetros numéricos das funções:

1. `glutInitWindowPosition` com: 0,0; 200,400
2. `glutInitWindowSize` com: 10x10; 800x600
 - Quando utilizar os valores 800x600, considere `glutInitWindowPosition` 0,0
3. `glClearColor` com: somente azul; somente vermelho; somente verde;
4. Modifique o nome da barra de títulos para `Aula_01_OpenGl – Criar Janelas`

Conhecimentos Necessários para Aula

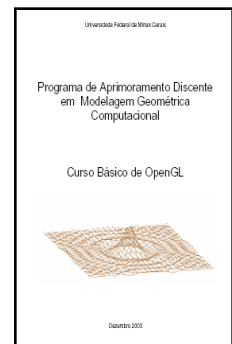
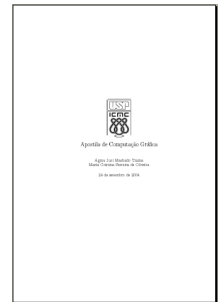
- **Linguagem C**
 - **Modularização (Funções em Linguagem C, com passagem de parâmetros)**
 - **Buffer**
 - **Imagem**
- 
- A diagram consisting of a light blue rectangular box with a black border containing the word "Programação". To the left of this box, a curly bracket groups the bullet points "Linguagem C" and "Modularização (Funções em Linguagem C, com passagem de parâmetros)". An arrow points from the bottom of the bracket to the box.

Conhecimentos Adquiridos com a Aula



- **Utilidade de um Sistema Gráfico**
 - **Instalação do OpenGL**
- **Máquina de Estados**
- **Entendimento sobre Imagens e Buffer**
- **Construção e Manipulação de Janelas**

Bibliografia

1. Azevedo, E., Conci, A. Computação Gráfica: Teoria e Prática. Rio de Janeiro: Elsevier, 2003. González, R. C., Woods, R. E.
2. Traina, A. J. M., Oliveira, M. C. F. Apostila de Computação Gráfica. Disponível em: www.icmc.sc.usp.br/gbdi, 2004.
3. Takahashi, R. et al. Apostila: Curso Básico de OpenGL: Programa de Aprimoramento Discente em Modelagem Geométrica Computacional, UFMG, 2003
4. Cohen, M., Manssour, I. H. OpenGL: uma abordagem prática. São Paulo, Novatec, 2006.






Anexo: OpenGL – definições dos comandos

- **glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)**
Inicializa as combinações de matrizes para escrita com cores, onde buffer é uma área de armazenamento para informações da imagem. Os componentes RGB são geralmente referenciados como color buffer ou pixel buffer.
- **glutSwapBuffers();**  faz com que qualquer comando OpenGL não executado seja executado.
- **glClearColor(0.0f, 0.0f, 1.0f, 0.0f)**  é a função que determina a cor utilizada para limpar a janela. O intervalo para cada componente red, green, blue é de 0 a 1. O componente alfa é usado para efeitos especiais, tal como transparência.


Anexo: OpenGL – definições dos comandos

- **glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);** Define o modo de exibição usado na janela criada. Nesse caso, cria uma janela com suporte para exibir cores compostas por Verm. Verde e Azul, além da técnica utilizada para escrita Depth
 - Argumentos indicam a criação de uma janela *double-buffered* (GLUT_DOUBLE) com o modo de cores RGB (GLUT_RGB). Uma janela *double-buffered* define que os comandos de desenho são executados para criar uma cena fora da tela (antecede a exibição) para depois rapidamente colocá-la na *view* (ou janela de visualização). Este método é geralmente utilizado para produzir efeitos de animação. O modo de cores RGB significa que as cores são especificadas através do fornecimento de intensidades dos componentes *red*, *green* e *blue* separadas. O GLUT_DEPTH define que a técnica utilizada para escrita na janela (matriz) será a partir de duas matrizes.

Anexo: OpenGL – definições dos comandos

- **glutInitWindowSize(450,450)**  permite definir uma janela com um tamanho de x=450 e y=450 (comprimento e altura em pixel)
- **glutInitWindowPosition(100,100)**  permite posicionar a janela em x=100 e y=100
- **glutCreateWindow("Primeiro Programa")**  é o comando da biblioteca GLUT que cria a janela. Neste caso, é criada uma janela com o nome "Primeiro Programa". Este argumento corresponde a legenda para a barra de título da janela.

Anexo: OpenGL – definições dos comandos

- **glutMainLoop()**  é a função que inicializa a execução da “máquina de estados” e processa todas as mensagens específicas do sistema operacional, tais como teclas e botões do mouse pressionados, até que o programa termine. Mantém a máquina de estados em funcionamento.