



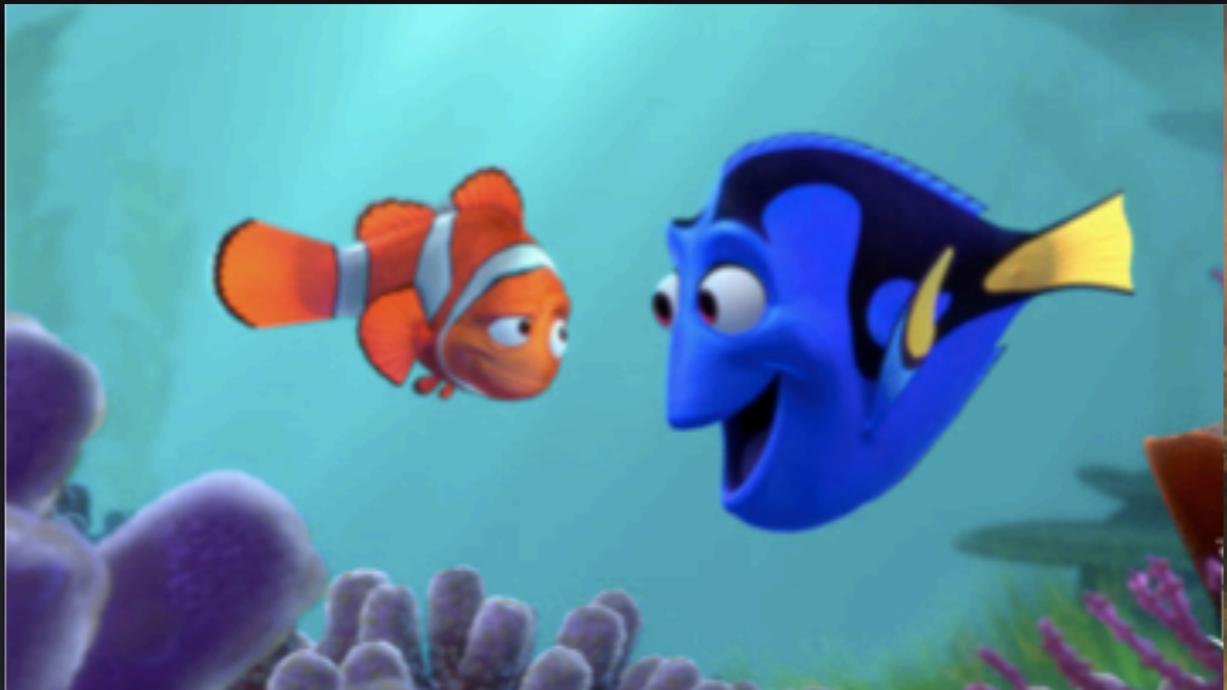
www.ime.usp.br/dcc

MAC-420: Introdução à Computação Gráfica

Marcel P. Jackowski
mjack@ime.usp.br

Aula #1: Objetivos e avaliação

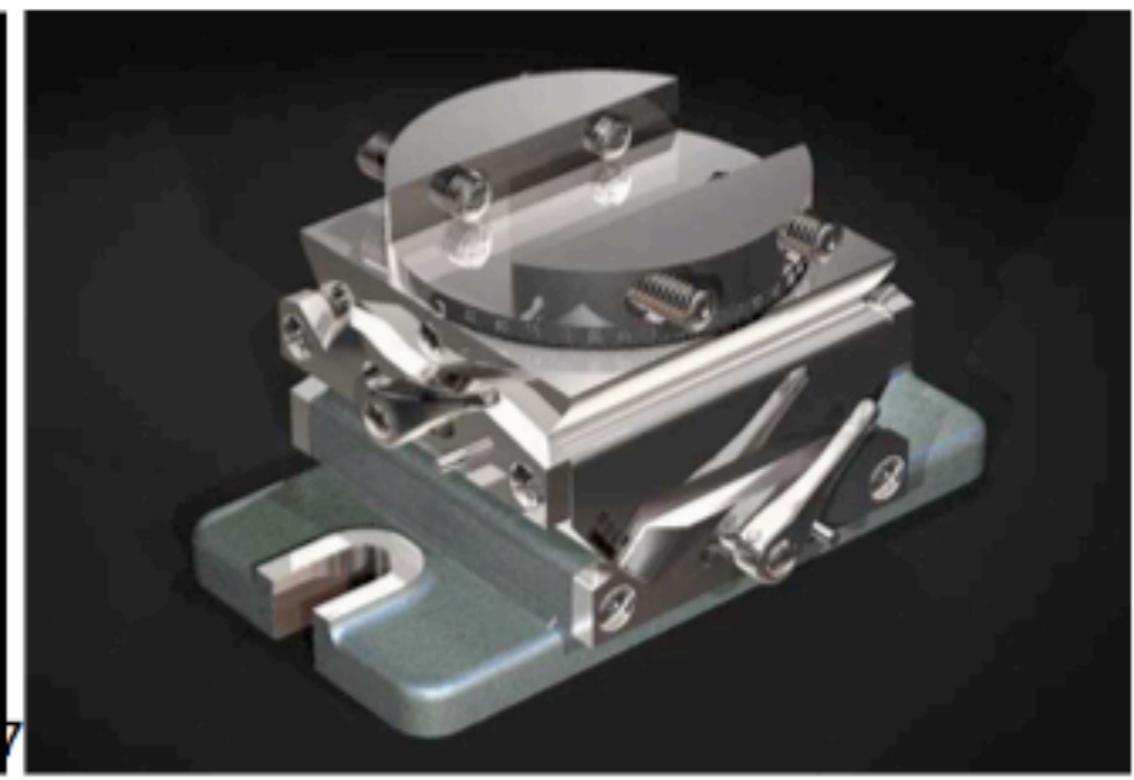
Filmes



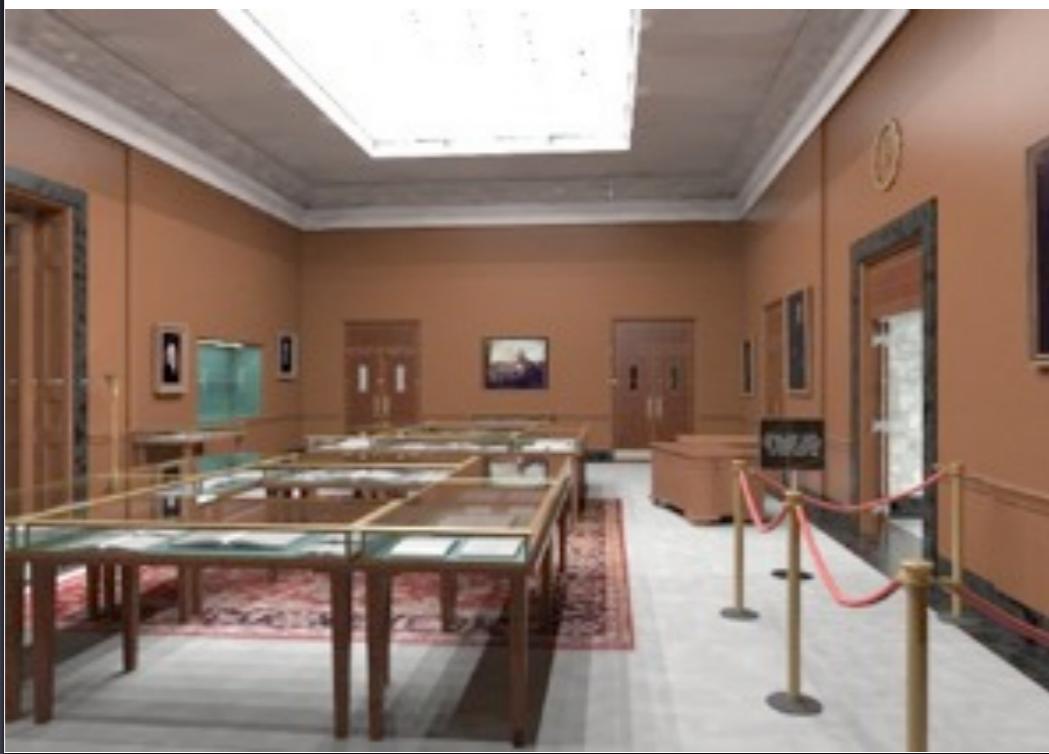
Simulações



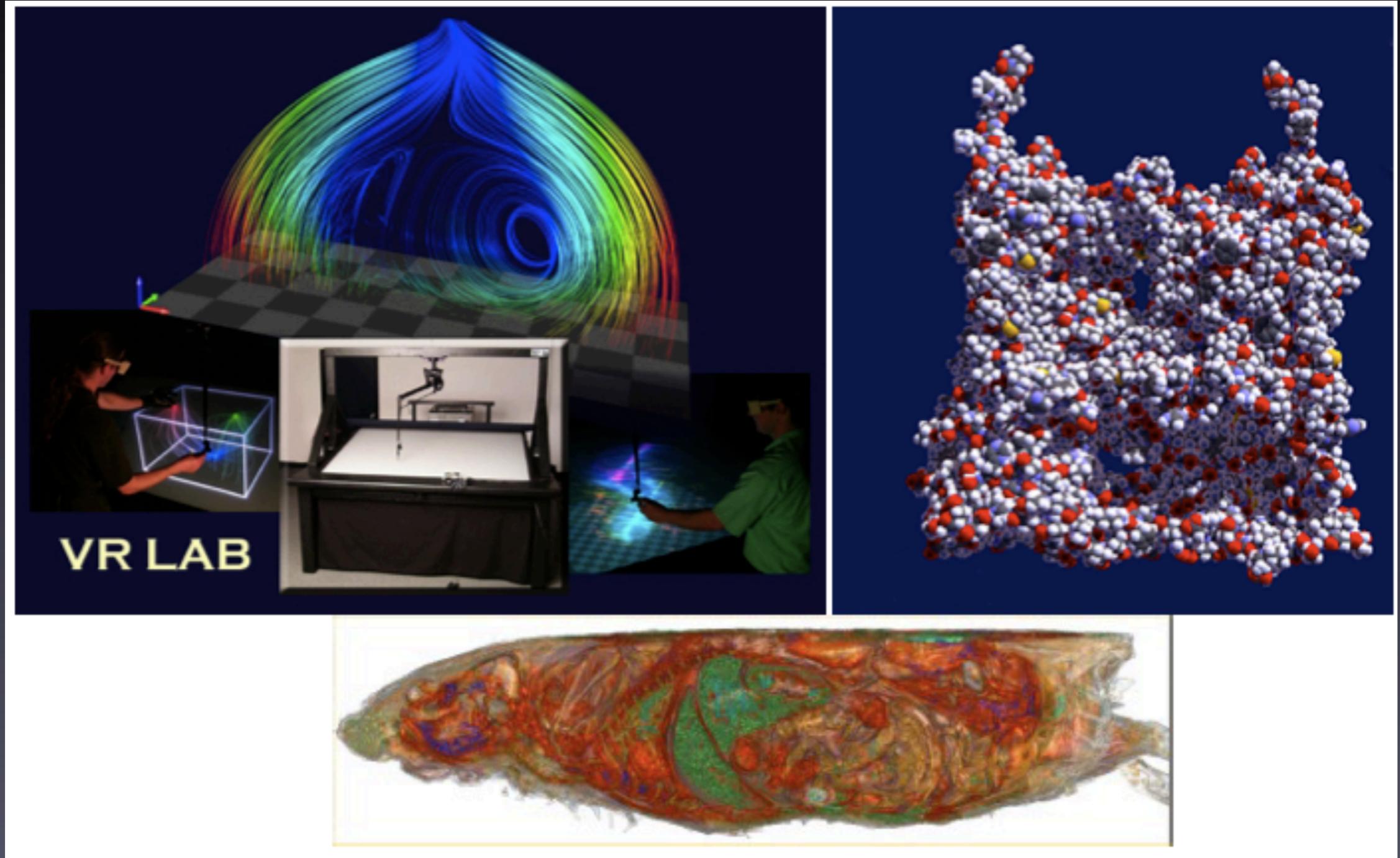
CAD/CAM



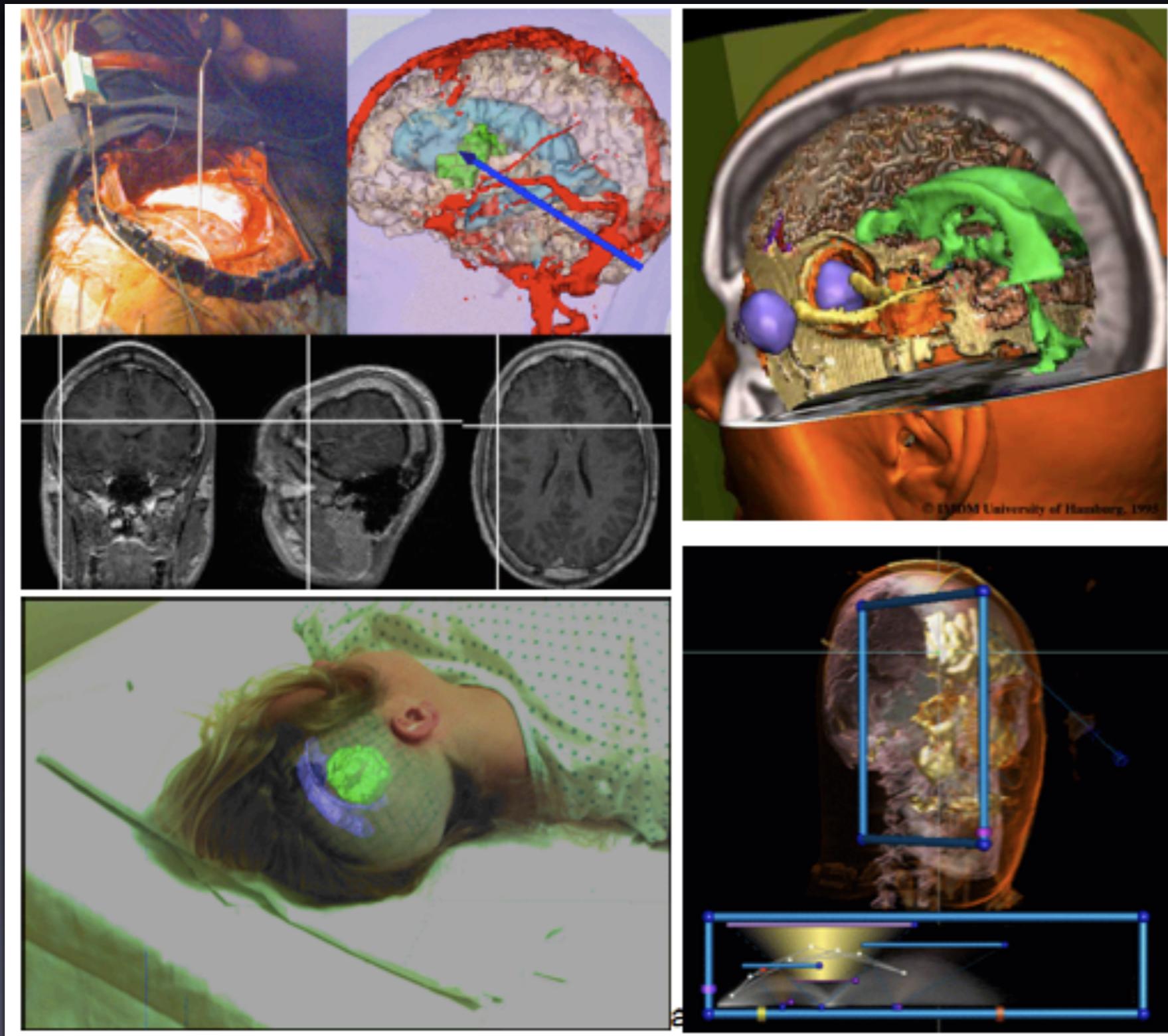
Arquitetura



Visualização científica



Medicina

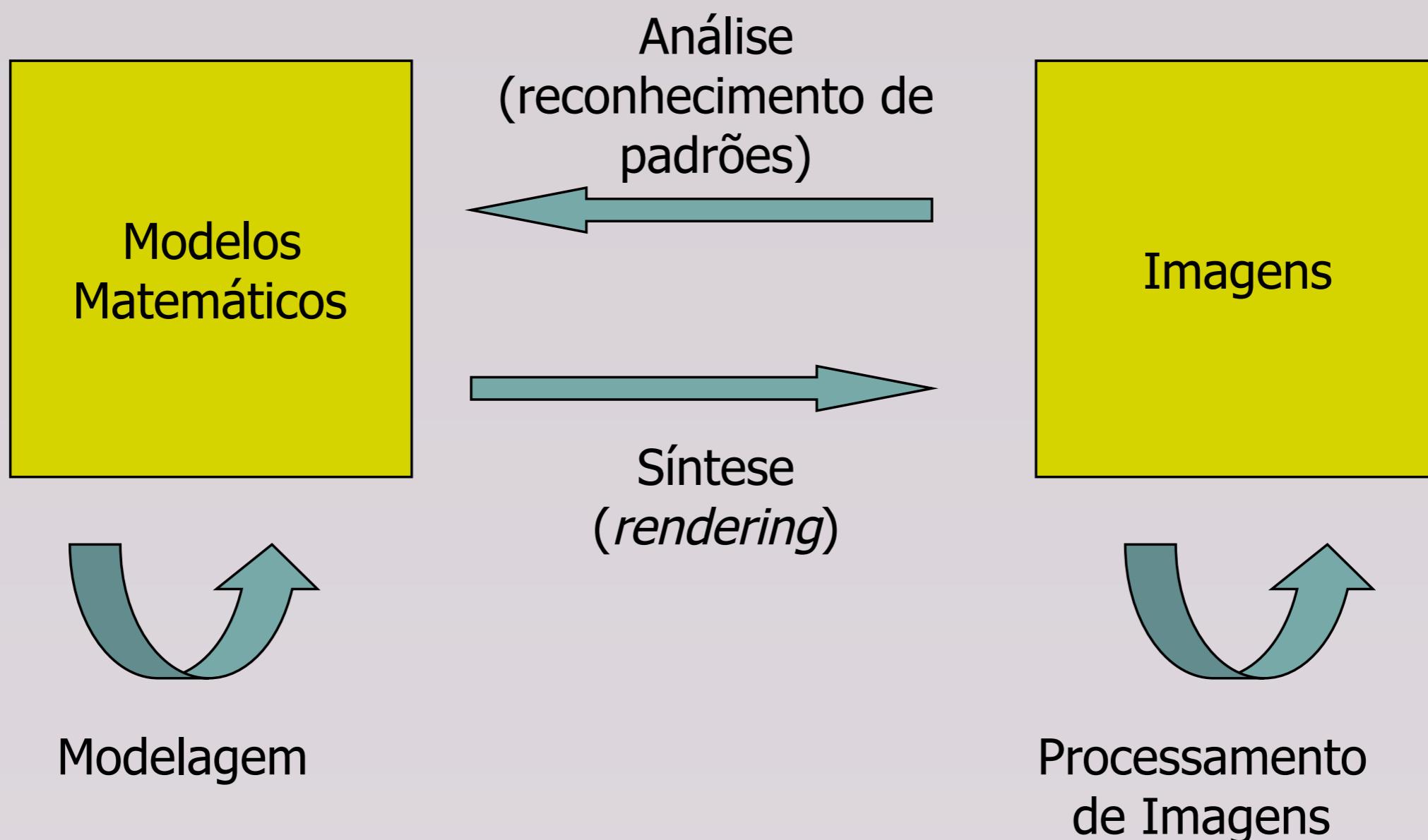


O que é Computação Gráfica ?

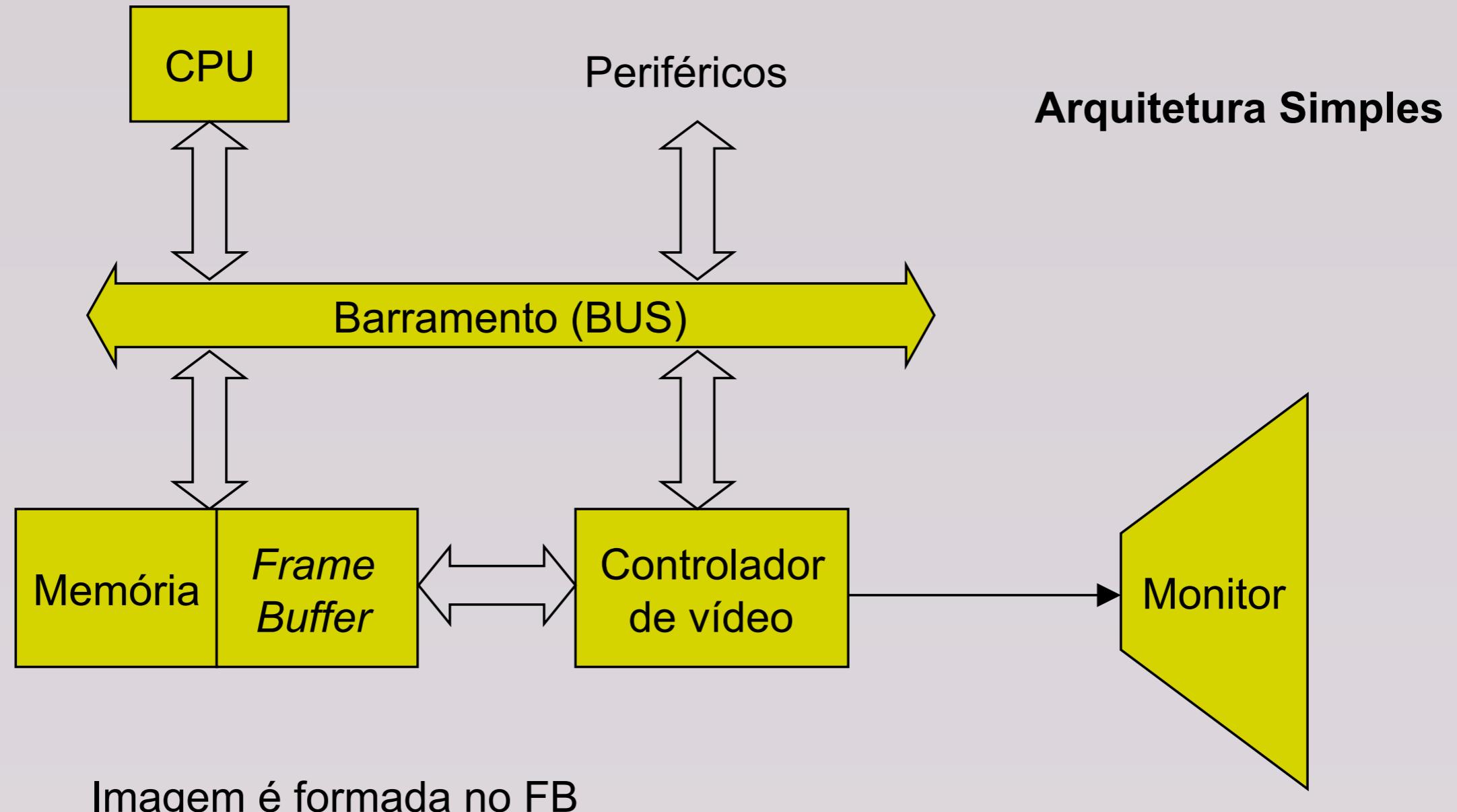
- A computação gráfica trata de todos os aspectos relacionados à criação ou síntese de imagens com um computador:
 - Hardware
 - Software
 - Aplicações



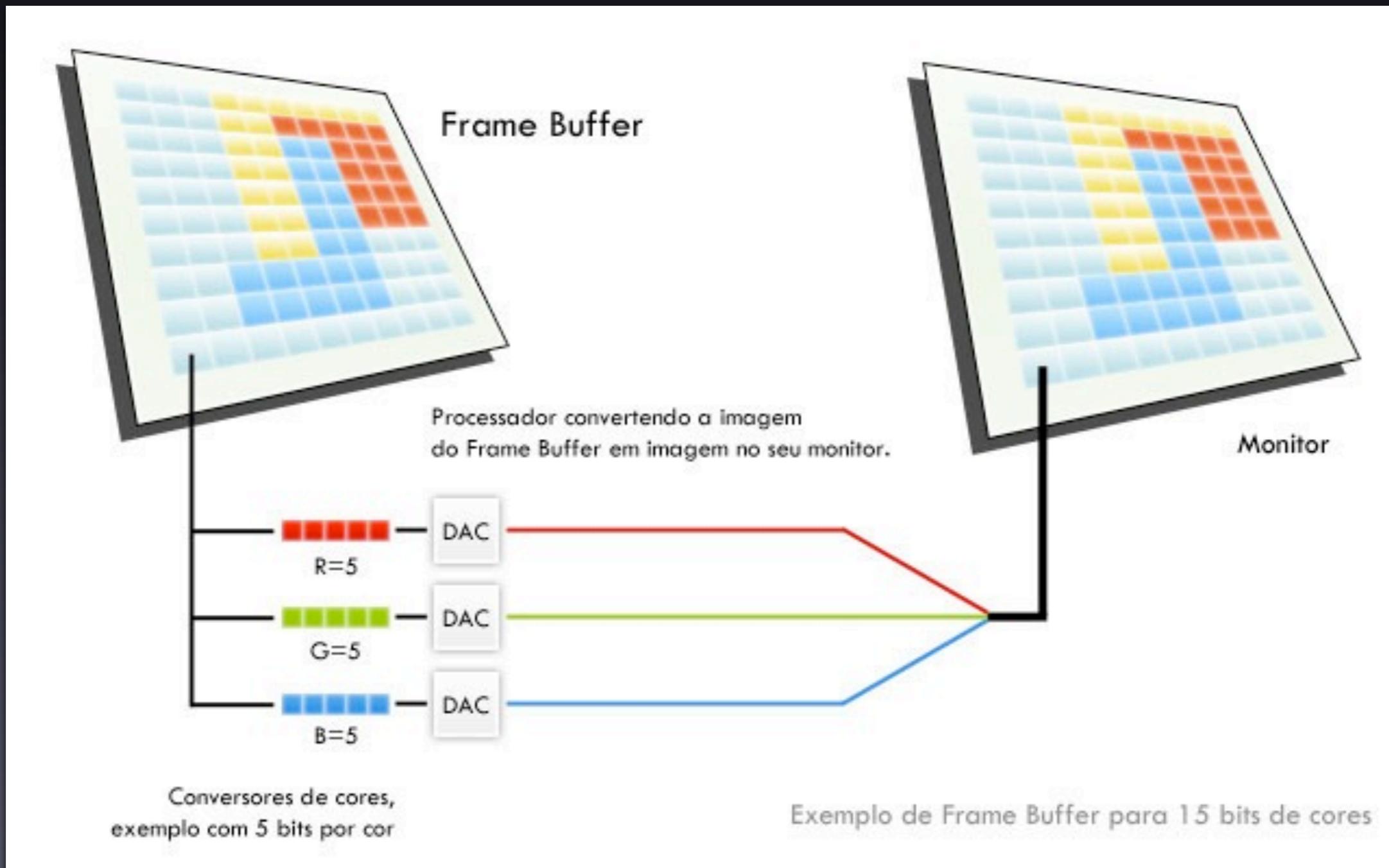
Computação Gráfica



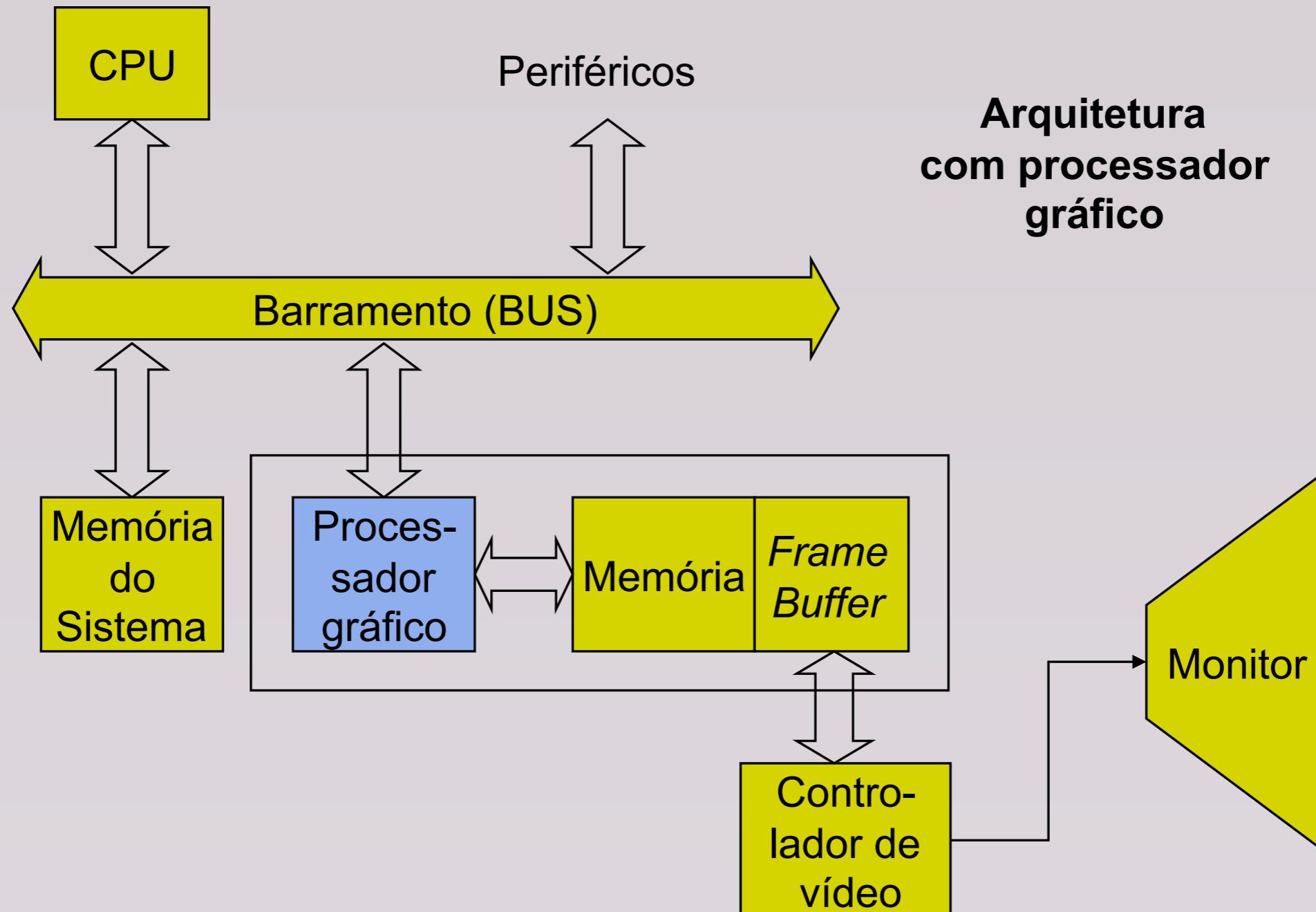
Arquiteturas



Frame buffer



Arquiteturas (ii)



Processador gráfico

- “**Graphics Processor Unit**” (**GPU**)
 - Hardware especializado que utiliza-se do paralelismo para atingir alto desempenho (CG)
- Alivia a CPU de várias tarefas, que incluem:
 - Transformações
 - Recorte
 - Projeções
 - Mapeamento de texturas
 - Rasterização
- Utiliza memória dedicada com maior banda.

O que você irá aprender

- Fundamentos dos algoritmos de computação gráfica e sua implementação
- Como API gráficas e hardware trabalham em conjunto
- Aplicação de conceitos de Matemática e Física em suas aplicações
- Aprender OpenGL e programação orientada a eventos
- Adquirir experiência de programação em C/C++

Conteúdo programático

- **Modelagem**
- **Transformações e Projeções**
 - Lineares, não lineares, paralela e perspectiva
- **Objetos geométricos**
- **Curvas e superfícies**
 - Implícitas, explícitas, paramétricas
- **Teoria das cores**
 - Sistemas de cores: RGB(A), HSV, YMCK
- **Sombreamento**
- **Recortes e preenchimento**
- **Traçado de raios (“Ray tracing”, “Ray casting”)**

Você não irá aprender...

- **Pacotes gráficos**
 - AutoCAD, 3D Studio Max, Maya, Blender
 - Photoshop e outros softwares.
- **Habilidades artísticas**
- **API gráficas em profundidade**
 - Microsoft Direct 3D
 - Shaders (GLSL, Cg, HLSL, etc)

O quê ? Matemática ?

- **Sim! Álgebra linear**
 - Pontos, vetores, operações entre eles
- **Conceitos mais avançados**
 - Coordenadas homogêneas
 - Equações diferenciais ordinárias (EDOs) e soluções numéricas
 - Quatérnios
 - Teoria de amostragem
- **Sempre de forma aplicada e visual.**

Exercícios-programas

- EP #1 (Abril)
- EP #2 (Maio)
- EP #3 (Junho)
- Entrega somente através do site PACA
- <http://paca.ime.usp.br>

$$M_{EP} = \frac{EP_1 + 2EP_2 + 3EP_3}{6}$$

Provas

- Prova #1
- Prova #2
- Prova Substitutiva
- Atenção: não haverá prova Rec!

$$M_P = \frac{P_1 + 2P_2}{3}$$

Média final

- Se MEP e MP forem pelo menos 5

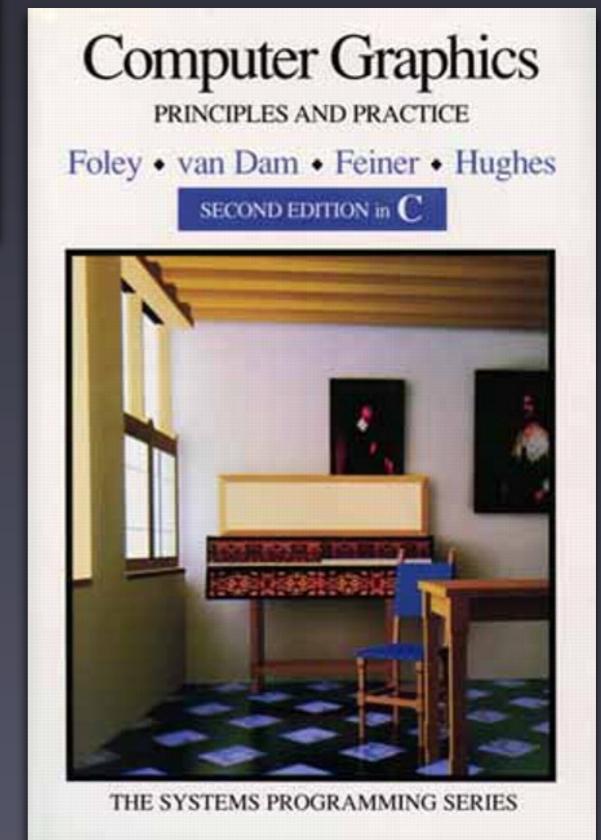
$$M_F = \frac{M_{EP} + M_P}{2}$$

- caso contrário, $M_F = \min \{MEP, MP\}$;
- Se $M_F \geq 5$ ($M_F \geq 7$; pós), você estará **aprovado**;
- caso contrário, você estará **reprovado**.

Referências

• Teoria

- Jonas Gomes e Luiz Velho.
Fundamentos da
Computação Gráfica,
IMPA. 2003.
- J. D. Foley, A. Van Dam, S.
K. Feiner and J. F. Hugues.
Computer Graphics:
Principles and Practice in
C. (2nd edition), Addison-
Wesley, 1995.



Referências

• Prática

- F. S. Hill, S. M. Kelley, Computer Graphics Using OpenGL, Prentice-Hall, 2006.
- M. Cohen e I. H. Manssour. OpenGL – Uma Abordagem Prática e Objetiva, Editora Novatec, 2006.
- D. Shreiner, M. Woo, J. Neider, T. Davis. OpenGL Programming Guide: The Official Guide to Learning OpenGL (5th Ed.), Addison-Wesley Professional, 2005.
- Guia versão 1.1: http://www.opengl.org/documentation/red_book



É divertida!

- Interdisciplinar
 - Computação, Matemática, Física, Artes, Cognição, Interação Homem-Máquina, etc.
- Ajuda a compreender a complexidade do mundo ao seu redor
- Você pode “ver” o resultado da sua aplicação (ou não :)
- Você pode “bular” ou enganar a realidade tanto que visualmente o resultado seja bom :->

Como se constrói essa cena ?



Como se constrói essa cena ?

Modelagem

- Geometria
- Materiais
- Luzes



Como se constrói essa cena ?

Modelagem

- Geometria
- Materiais
- Luzes

Animação

- Movimento



Como se constrói essa cena ?

Modelagem

- Geometria
- Materiais
- Luzes

Animação

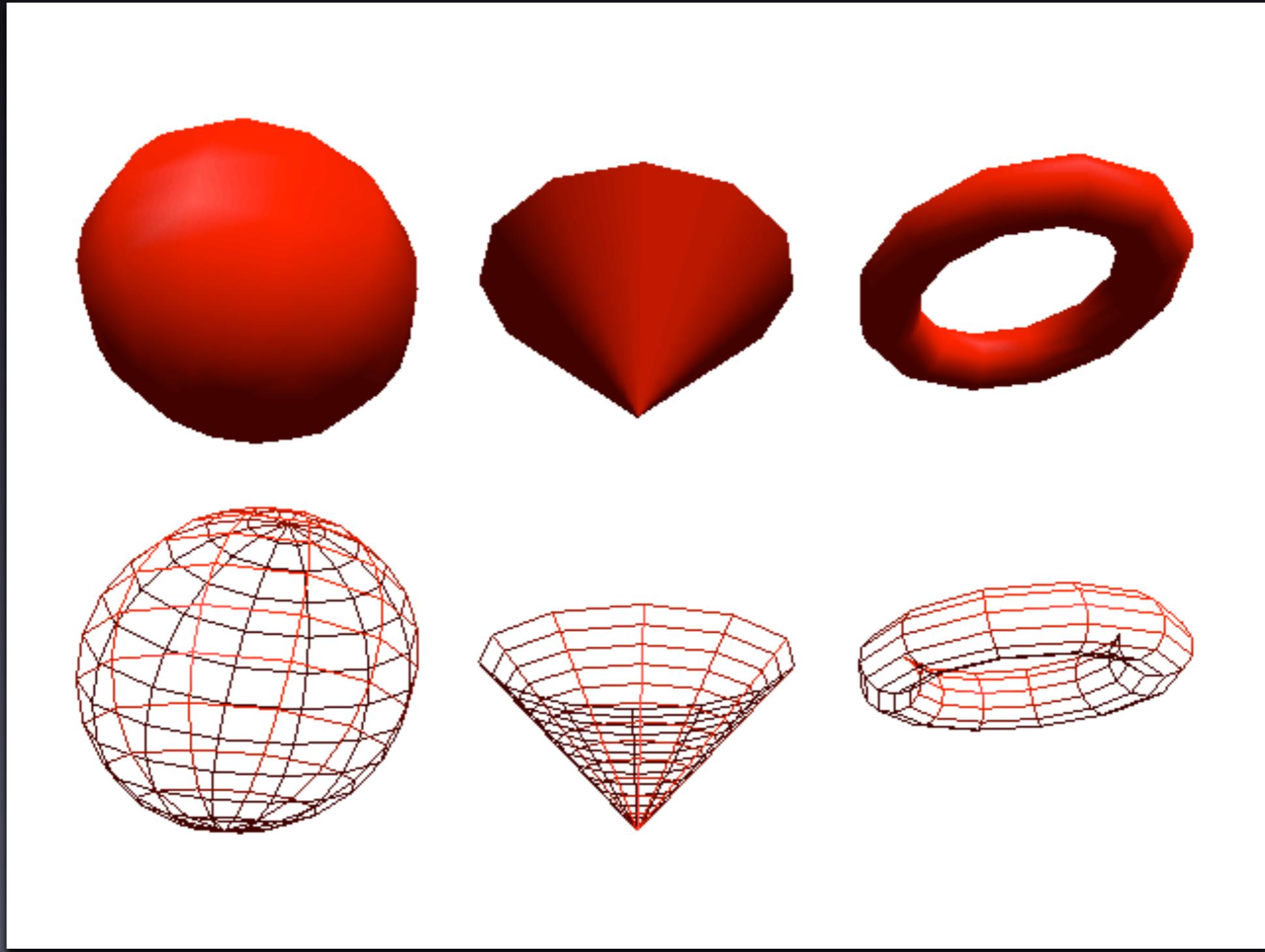
- Movimento

Renderização

- Desenho da cena
- Iluminação, sombras, texturas



shapes.c



Tarefa de Aquecimento

1. Descubra como compilar e executar o programa
“**shapes.c**” no seu sistema operacional

1. Quais as bibliotecas a serem “linkadas” ?

2. Quais os “headers” necessários ?

1. Linux, Windows, Mac OS X ?

2. Responda para que serve a função `glFrustum`

3. Quando o seu programa alcançará a linha “`return EXIT_SUCCESS`” ?

4. Respostas no fórum!