

IA725 – Computação Gráfica I

Professores:

Léo Pini Magalhães (leopini@dca.fee.unicamp.br)

Wu Shin – Ting (ting@dca.fee.unicamp.br)

Sala: PE-24

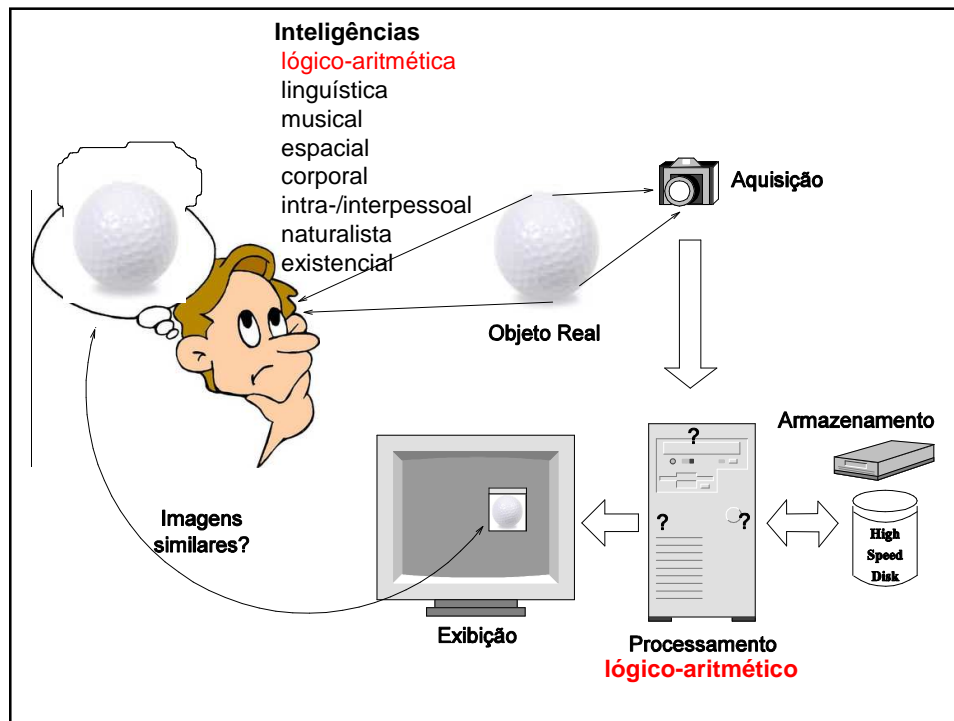
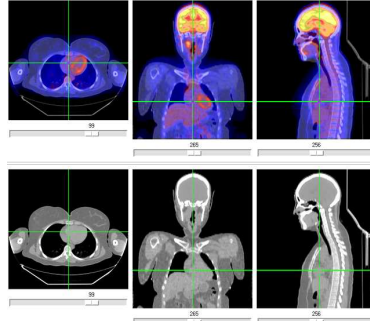
Horário: Segundas e quartas, das 10:00h às 12:00h

<http://www.dca.fee.unicamp.br/cursos/IA725/1s2010>

Computação Gráfica é a área da **Ciência da Computação** que estuda a **construção de imagens** por computadores, a partir de dados processáveis por estes.

Aplicações

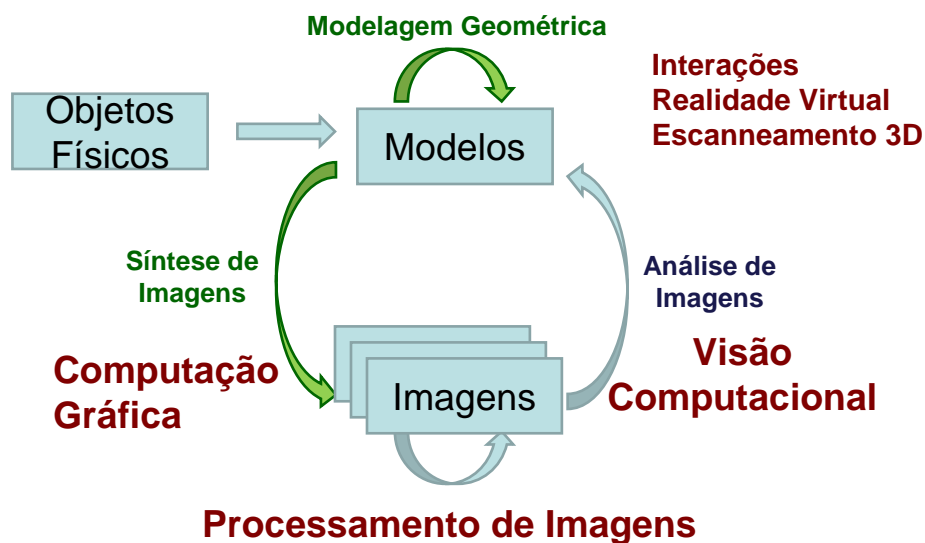
- Arte digital
- Publicidade
- Entretenimento (Jogos, Cinema)
- Arquitetura
- Engenharia (CAD/CAM)
- Simulações
- Medicina
- Geoprocessamento
- Visualização de Informação etc



Sem inteligências “espacial” e “naturalista”,

- como as máquinas podem “ver” figuras, cor, tons e sombreamentos?
- como as máquinas podem produzir, a partir de conceitos, imagens próximas à nossa percepção ou “recriar” experiências visuais?

Áreas Correlatas



Paradigma

Cubo

Concepção

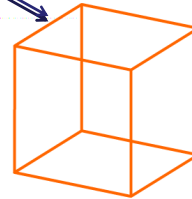
Representação
numérica

Linguagem
Matemática

Linguagem
de
Programação

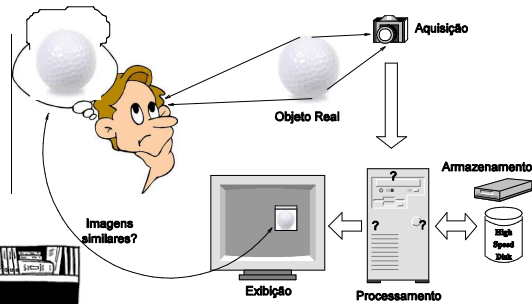
```
// top face
p0 = {x:-50, y:-50, z:-50};
p1 = {x:50, y:-50, z:-50};
p2 = {x:50, y:-50, z:50};
p3 = {x:-50, y:-50, z:50};
// bottom face
p4 = {x:-50, y:50, z:-50};
p5 = {x:50, y:50, z:-50};
p6 = {x:50, y:50, z:50};
p7 = {x:-50, y:50, z:50};
pointArray = [p0, p1, p2, p3, p4, p5, p6, p7];

// p0 to p8 , the 3d points of a cube;
```



Modelos de Formação de Imagens 2D

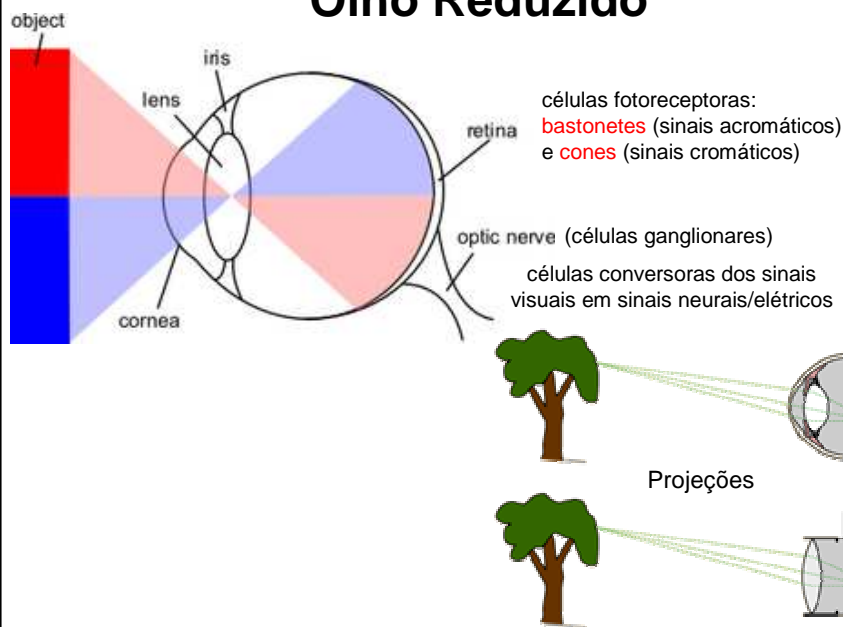
Duas alternativas



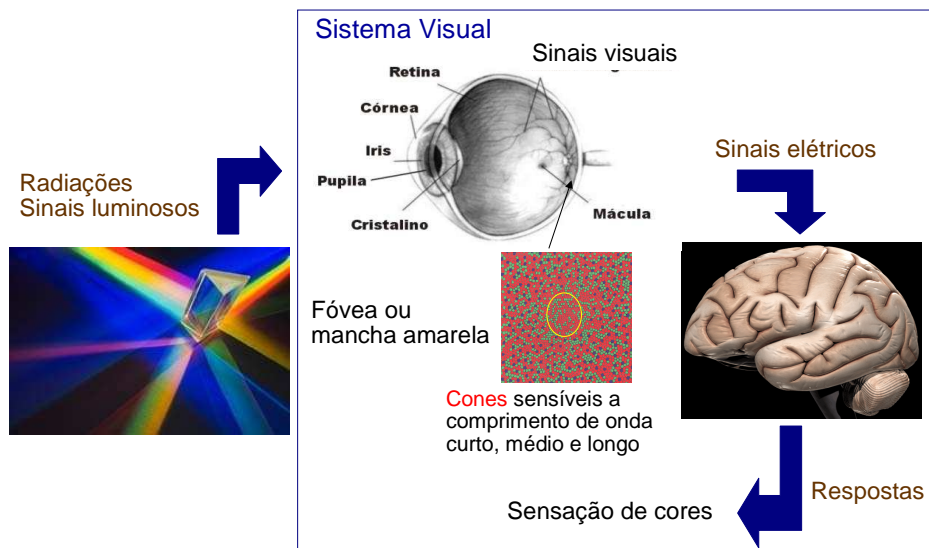
Conceber as formas em 3D

Conceber as formas em 2D

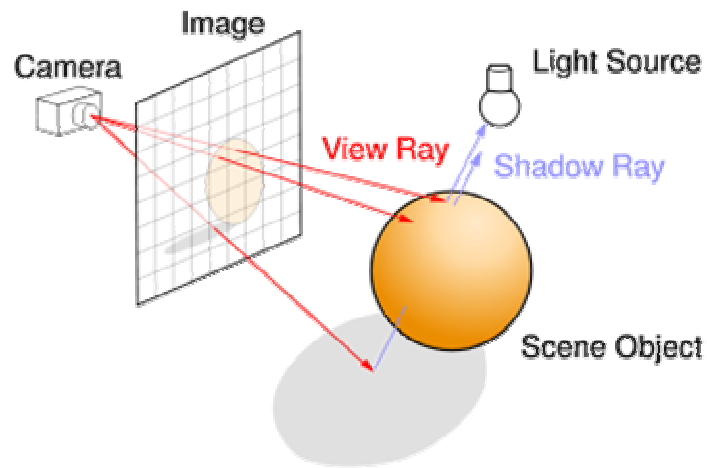
Olho Reduzido



Percepção de Cores



Direção dos Raios Luminosos



Paradigma

Cubo

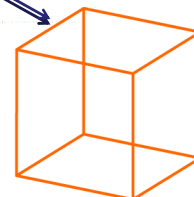
Concepção

Representação numérica:

- Geometria
- Trajetória dos raios
- Cor

Linguagem Matemática

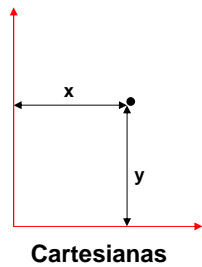
Linguagem de Programação



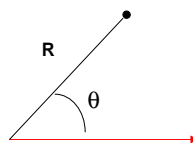
Pontos

Distintas formas de **identificação** por números

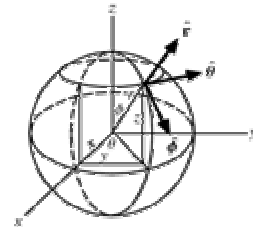
- Coordenadas cartesianas
- Coordenadas polares
- Coordenadas cilíndricas
- Coordenadas esféricas



Cartesianas



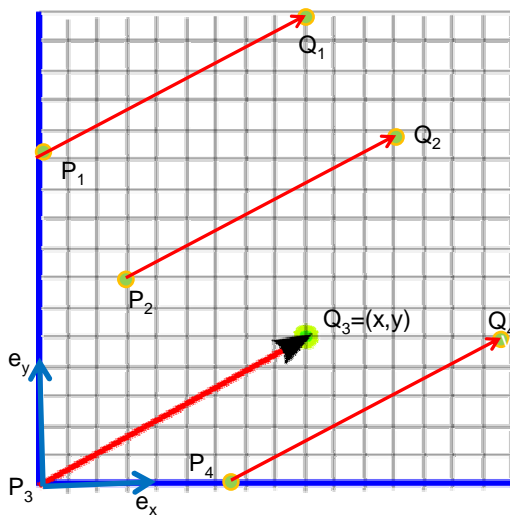
Polares



Esféricas

n valores escalares no espaço R^n

Vetores

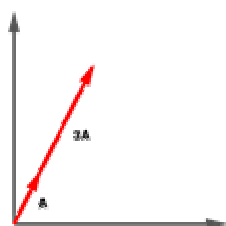
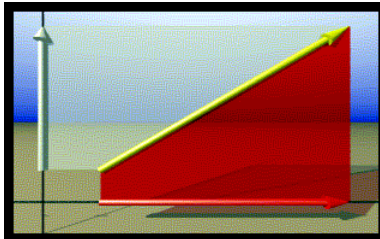


“Diferença” entre 2 pontos
 $\mathbf{v} = \mathbf{Q}_i - \mathbf{P}_i$

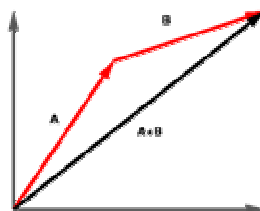
- Magnitude ($|\mathbf{v}|$)
- Orientação

Elementos geométricos que
podem ser manipulados
algebricamente.

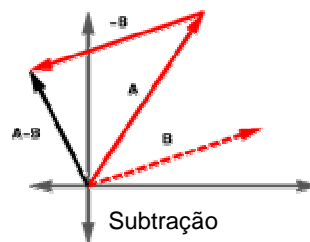
Operações sobre Vetores



Multiplicação por um escalar

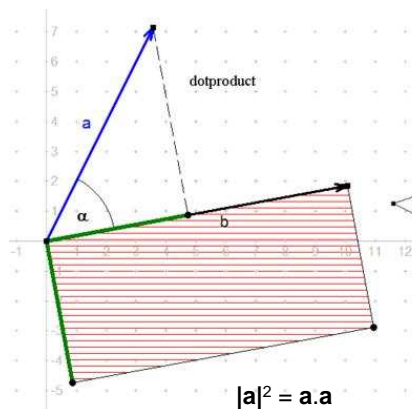


Adição



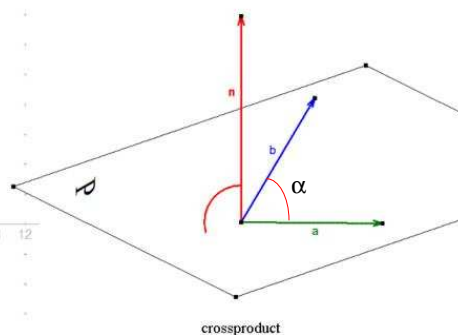
Subtração

Operações sobre Vetores Produto Escalar e Vetorial



$$|a|^2 = a \cdot a$$

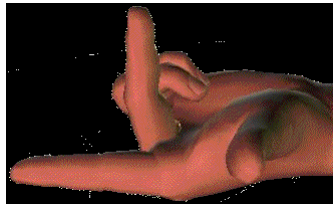
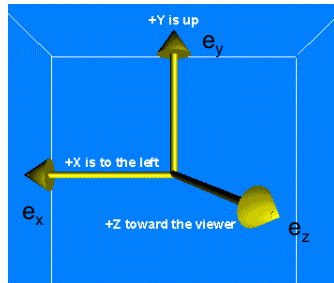
$$\cos \alpha = \frac{a \cdot b}{|a||b|}$$



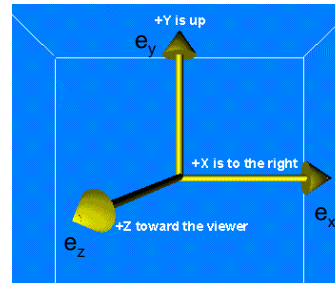
$$n = a \times b$$

$$\sin \alpha = \frac{|a \times b|}{|a||b|}$$

Sistemas de Referência 3D

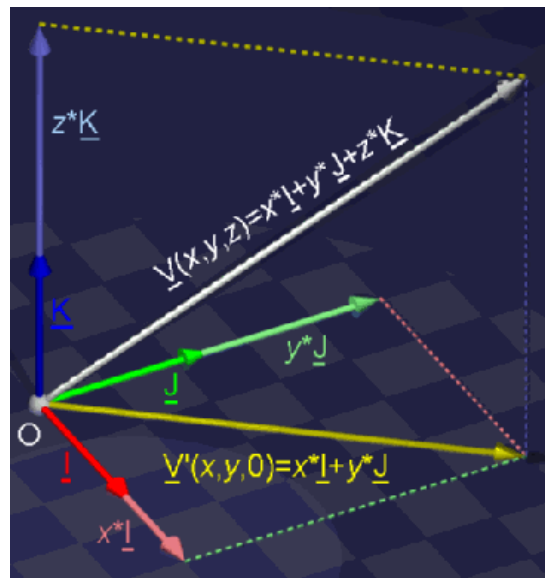


Mão-esquerda



Mão-direita

Combinações Lineares



Combinações Lineares

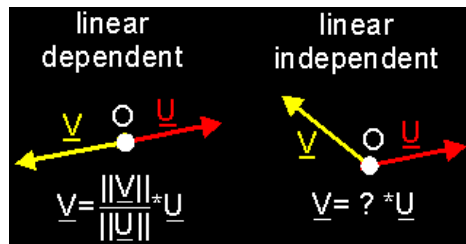
$$\vec{v} = a_1 \vec{v}_1 + a_2 \vec{v}_2 + \dots + a_n \vec{v}_n$$

a_i : escalares $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$: vetores

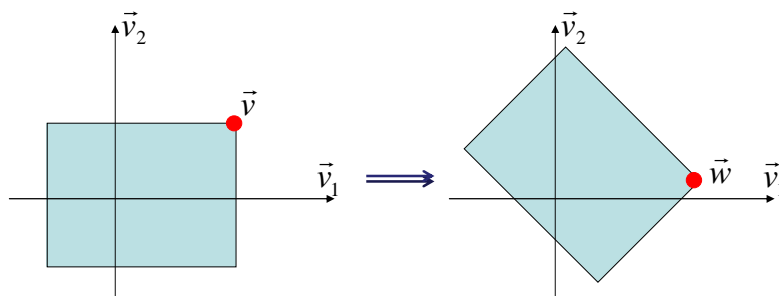
$\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$ são **linearmente dependentes** se existem a_i , nem todos nulos, tais que

$$a_1 \vec{v}_1 + a_2 \vec{v}_2 + \dots + a_n \vec{v}_n = 0$$

Em caso contrário, eles são **linearmente independentes**.



Transformações



$$\vec{v} = a_1 \vec{v}_1 + a_2 \vec{v}_2 + \dots + a_n \vec{v}_n$$

$\downarrow T$

$$\vec{w} = b_1 \vec{v}_1 + b_2 \vec{v}_2 + \dots + b_n \vec{v}_n$$

Álgebra Linear

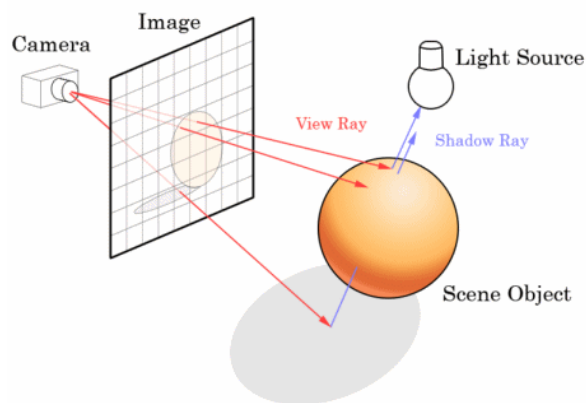
$$\vec{v} = a_1 \vec{v}_1 + a_2 \vec{v}_2 + \dots + a_n \vec{v}_n$$

↓
T

$$\vec{w} = b_1 \vec{v}_1 + b_2 \vec{v}_2 + \dots + b_n \vec{v}_n$$

$$\mathbf{T} \vec{v} = \vec{w} \Leftrightarrow \begin{bmatrix} t_{00} & t_{01} & \cdots & t_{0n} \\ t_{10} & t_{11} & \cdots & t_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{1n} & t_{2n} & \cdots & t_{nn} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}$$

Métodos Numéricos



Solucionar sistemas de equações (p.ex., interseção de raios luminosos com superfícies)

Paradigma

Cubo

Concepção

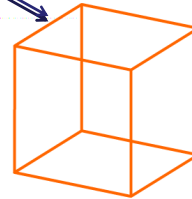
Representação
numérica

Linguagem
Matemática

Linguagem
de
Programação

```
// top face
p0 = {x:-50, y:-50, z:-50};
p1 = {x:50, y:-50, z:-50};
p2 = {x:50, y:-50, z:50};
p3 = {x:-50, y:-50, z:50};
// botton face
p4 = {x:-50, y:50, z:-50};
p5 = {x:50, y:50, z:-50};
p6 = {x:50, y:50, z:50};
p7 = {x:-50, y:50, z:50};
pointArray = [p0, p1, p2, p3, p4, p5, p6, p7];

// p0 to p8 , the 3d points of a cube;
```



Processamento Gráfico

- Envolve muitas operações em ponto flutuante. Considere padrão IEEE.
- A depuração é mais complexa.
- Tem como objetivo produzir imagens perceptualmente aceitáveis.
- Pode ter a eficiência melhorada, se levarmos em conta a arquitetura do sistema.

Sistema Gráfico

Armazenamento



Processamento

Síntese
Processamento
Análise
Interações



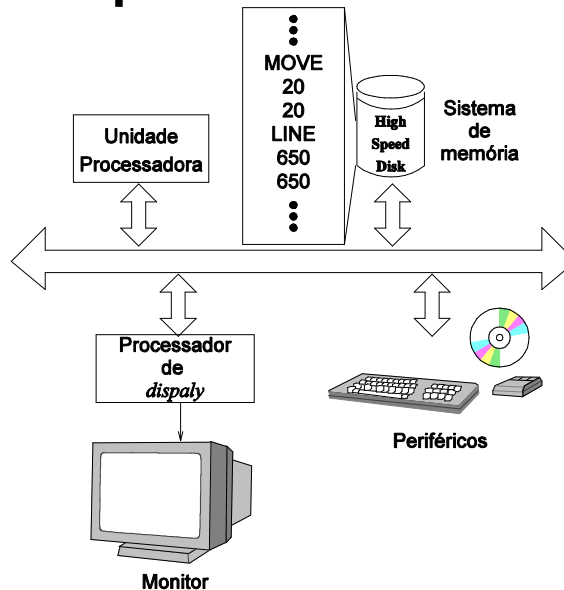
Aquisição



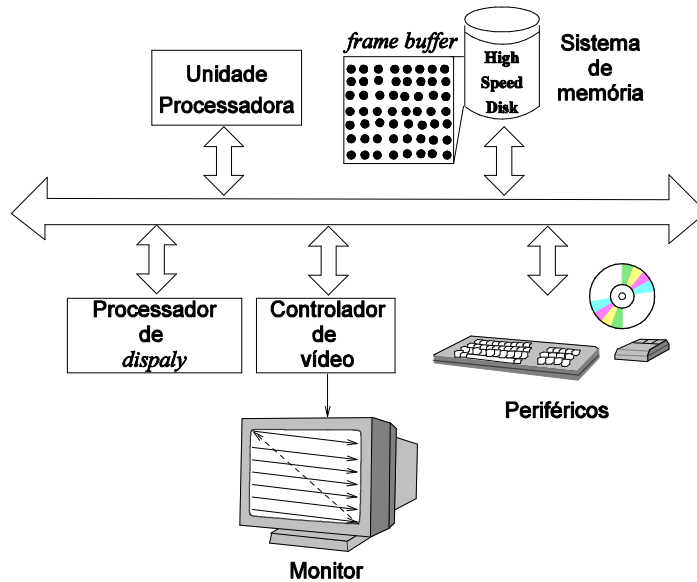
Exibição



Arquitetura Vetorial



Arquitetura *Raster*



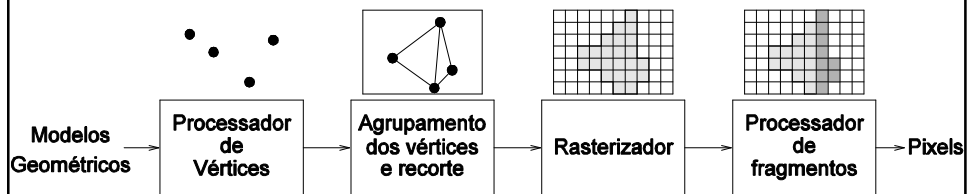
API

Interface do Programa de Aplicação

Interface de programação das funcionalidades do sistema

- Funções gráficas integradas com as funções de gerenciamento de janelas: **Java3D**
- Comandos gráficos independentes dos comandos relacionados com o sistema de janelas: **OpenGL**, **Direct 3D**

OpenGL



OpenGL – Um Código-fonte

```
/* Definição da geometria e das fontes luminosas */

#include <GL/glut.h>

GLfloat light_diffuse[] = {1.0, 0.0, 0.0, 1.0}; /* Red diffuse light. */
GLfloat light_position[] = {1.0, 1.0, 1.0, 0.0}; /* Infinite light
location. */
GLfloat n[6][3] = { /* Normals for the 6 faces of a cube. */
    {-1.0, 0.0, 0.0}, {0.0, 1.0, 0.0}, {1.0, 0.0, 0.0},
    {0.0, -1.0, 0.0}, {0.0, 0.0, 1.0}, {0.0, 0.0, -1.0} };
GLint faces[6][4] = { /* Vertex indices for the 6 faces of a cube. */
    {0, 1, 2, 3}, {3, 2, 6, 7}, {7, 6, 5, 4},
    {4, 5, 1, 0}, {5, 6, 2, 1}, {7, 4, 0, 3} };
GLfloat v[8][3]; /* Will be filled in with X,Y,Z vertexes. */
```

OpenGL – Um Código-fonte

```
void displayBox(void) {
    /* Enable a single OpenGL light. */
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
    glEnable(GL_LIGHT0);
    glEnable(GL_LIGHTING);
    /* Setup the view of the cube. */
    glMatrixMode(GL_PROJECTION);
    gluPerspective(
        /* field of view in degree */ 40.0,
        /* aspect ratio */ 1.0,
        /* Z near */ 1.0, /* Z far */ 10.0);
    glMatrixMode(GL_MODELVIEW);
    drawBox();
}

void drawBox(void) {
    int i;

    gluLookAt(0.0, 0.0, 5.0, /* eye is at (0,0,5) */
              0.0, 0.0, 0.0, /* center is at (0,0,0) */
              0.0, 1.0, 0.0); /* up is in positive Y direction */
    for (i = 0; i < 6; i++) {
        glBegin(GL_QUADS);
        glNormal3fv(&n[i][0]);
        glVertex3fv(&v[faces[i][0]][0]);
        glVertex3fv(&v[faces[i][1]][0]);
        glVertex3fv(&v[faces[i][2]][0]);
        glVertex3fv(&v[faces[i][3]][0]);
        glEnd();
    }
}
```

Quais profissões?

Artista/Designer

Utilizam ferramentas computacionais para criar soluções de visualização e interatividade.

Engenheiros

Aplicam técnicas conhecidas para a implementação de ferramentas computacionais capazes de gerar e processar imagens.

Pesquisadores

Desenvolvem tecnologias de hardware e software para síntese de imagens foto-realistas

O que se estuda?

Perfil de Artista/Designer

- Ferramentas (AutoCAD, Maya, Photoshop, CorelDraw, Premiere, etc)
- Artes

Perfil de Engenheiro

- Programação
- Projetos Digitais
- Matemática Básica (Trigonometria, Transformadas, Cálculo vetorial, Álgebra e Métodos Numéricos)
- Física Básica (Óptica, Luz, Cinemática e Dinâmica)

Perfil de Pesquisador

- Programação
- Matemática Avançada
- Física Avançada
- Técnicas Avançadas de Projetos Digitais