




Java 3D™ API


Profa. Isabel Harb Manssour
(<http://www.inf.pucrs.br/~manssour/Java3D>)

XVI SIBGRAPI – 12-15 de Outubro de 2003




Roteiro

1. **Introdução**
2. **Criando Universos Virtuais**
3. **Realismo, Interação e Animação**
4. **Comentários Finais**



1. Introdução



Introdução

- **Conceitos de Computação Gráfica** ▶
- Linguagem Java
- A API Java 3D
- Histórico
- Aplicações
- Instalação e Utilização



Introdução

Conceitos de Computação Gráfica

■ Transformações geométricas

- Operações matemáticas que permitem alterar uniformemente o **aspecto** de objeto(s), mas não a sua **topologia**
- Tipos
 - Escala
 - Rotação
 - Translação



Introdução

Conceitos de Computação Gráfica

■ Transformações geométricas: Translação

- T_x, T_y, T_z : constantes de deslocamento

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$



Introdução

Conceitos de Computação Gráfica

■ Transformações geométricas: Escala

- E_x, E_y, E_z : fatores de escala

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} E_x & 0 & 0 & 0 \\ 0 & E_y & 0 & 0 \\ 0 & 0 & E_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Introdução

Conceitos de Computação Gráfica

■ Transformações geométricas: Rotação

- Deve-se definir o plano sobre o qual a rotação será realizada (em torno de qual eixo se procederá a rotação)

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \text{sen} \alpha & 0 \\ 0 & -\text{sen} \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_y = \begin{bmatrix} \cos \alpha & 0 & -\text{sen} \alpha & 0 \\ 0 & 1 & 0 & 0 \\ \text{sen} \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_z = \begin{bmatrix} \cos \alpha & \text{sen} \alpha & 0 & 0 \\ -\text{sen} \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Introdução

Conceitos de Computação Gráfica

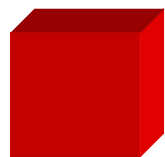
- Transformações geométricas
 - Em geral são acumuladas
 - É feita uma combinação das matrizes em uma só (multiplicando-as)



Introdução

Conceitos de Computação Gráfica

- Projeção no plano da imagem



3D → 2D



Introdução

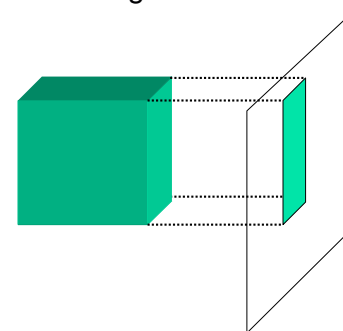
Conceitos de Computação Gráfica

- Projeção no plano da imagem

3D
 (x,y,z)

2D
 (x,y)
 (x,z)
 (y,z)

Projeção Paralela



Introdução

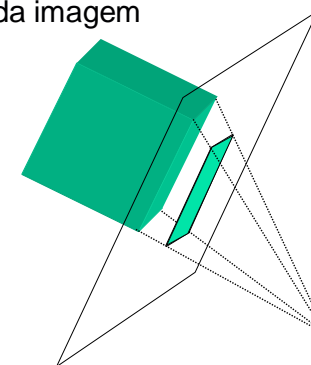
Conceitos de Computação Gráfica


- Projeção no plano da imagem

3D
 (x,y,z)


2D
 (x',y')
 (x',z')
 (y',z')


Projeção Perspectiva





 **Introdução**
Conceitos de Computação Gráfica

- Câmera Sintética
 - “Observador Virtual”
 - Define de que local se deseja que a cena seja exibida (de cima, de baixo, do lado esquerdo, do lado direito, etc)
 - Necessidade da existência deste observador: um mesmo conjunto de objetos no universo 3D, visto de diferentes lugares tem, para quem os observa, diferentes coordenadas para cada posição




 **Introdução**

- Conceitos de Computação Gráfica
- **Linguagem Java** 
- A API Java 3D
- Histórico
- Aplicações
- Instalação e Utilização

 **Introdução**
Linguagem Java

- Java
 - Linguagem de programação
 - Orientada a objetos
 - Independente de arquitetura (multiplataforma)
 - Portável
 - Robusta
 - Segura
 - Interpretada
 - Etc...

 **Introdução**
Linguagem Java

- Java
 - Compilada e interpretada
 - O compilador transforma o programa em *bytecodes* (instruções de máquina compreendidas pela *Java Virtual Machine*)
 - O interpretador transforma os *bytecodes* em linguagem de máquina

Introdução

Linguagem Java

- Exemplo:

Classe

Professor
- nome: String - matricula: int - cargaHoraria: int
+ setNome (n: String) + getNome (): String + setMatricula (m: int) + getMatricula(): int + setChsemanal (c: int) + getChsemanal(): int + getChmensal(): float

→ Atributos

→ Métodos

Introdução

Linguagem Java

- Herança
 - Capacidade de reusar código pela especialização de soluções genéricas já existentes

```

public class classeA {}
public class classeB extends classeA {}
public class classeC extends classeB {}

```

classeA: superclasse direta de B, superclasse indireta de C

classeB: subclasse direta de A, superclasse direta de C

classeC: subclasse direta de B, subclasse indireta de A

Introdução

Linguagem Java

- Exemplo:

Professor
- nome: String - matricula: int - cargaHoraria: int
+ setNome (n: String) + getNome (): String + setMatricula (m: int) + getMatricula(): int + setChsemanal (c: int) + getChsemanal(): int + getChmensal(): float

p1

nome: Maria
 matrícula: 110
 cargaHorária: 12

p2

nome: José
 matrícula: 123
 cargaHorária: 40

Objetos

```

Professor p1, p2;
p1 = new Professor();
p1.setNome("Maria");
...
p2 = new Professor();
p2.setNome("Jose");
...

```

Introdução

Linguagem Java

- Pacote
 - Recurso interessante que permite agrupar classes relacionadas (que tenham afinidades funcionais)
 - Estruturam sistemas de grande porte, facilitando a localização das classes
 - Java dispõe de uma grande quantidade de pacotes:
 - java.lang, java.util, ...
 - Devem ser "importados"
 - import javax.swing.JOptionPane;

Introdução
Linguagem Java

- Interface gráfica com o usuário
 - `java.awt`: *Abstract Windowing Toolkit* (classes básicas);
 - `javax.swing`: *Swing Components* - fornece melhores alternativas aos componentes definidos na classe `java.awt`

Introdução
Linguagem Java

- Interface gráfica com o usuário
 - *Layouts*

The image shows three windows illustrating different Java layouts:

- FlowLayout**: A window with a title bar, a text field, and two buttons labeled 'dois' and 'três' arranged horizontally.
- BorderLayout**: A window with a title bar, a text field at the top, and three buttons labeled '4', '5', and '2' arranged in a grid-like structure.
- GridLayout**: A window with a title bar and three buttons labeled 'um', 'dois', and 'três' arranged in a 2x2 grid (with the top-right cell empty).

Introdução
Linguagem Java

- Interface gráfica com o usuário
 - *Containers e Componentes*

The diagram shows a window titled 'Componentes de Alto-nível' with a blue title bar. A 3D effect shows the window's structure with labels:

- Frame**: The outer border of the window.
- Menu Bar**: The blue title bar area.
- Content Pane**: The main area of the window where content is displayed.

Introdução
Linguagem Java

- API Java
 - *Java Base API*
 - I/O
 - GUI
 - ...
 - *Java Standard Extension API*
 - Acesso a banco de dados
 - *Java Media Framework*
 - ...

Introdução

Linguagem Java

- *Java Media Framework* API
 - Suporta tecnologias gráficas e multimídia
 - Áudio
 - Vídeo
 - 2D
 - Animação
 - 3D
- Um dos componentes da *API Java Standard Extension* é Java 3D

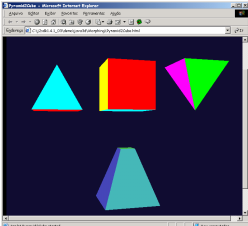
Introdução

- Conceitos de Computação Gráfica
- Linguagem Java
- **A API Java 3D**
- Histórico
- Aplicações
- Instalação e Utilização

Introdução

A API Java 3D

- O que é a API Java 3D?
 - Hierarquia de classes Java
 - Interface para o desenvolvimento de sistemas gráficos tridimensionais




Introdução

A API Java 3D

- O que é a API Java 3D?
 - Construtores de alto nível
 - Criação e manipulação de objetos geométricos, especificados em um universo "virtual"
 - Grande flexibilidade para criar universos virtuais
 - Representação de cenas: "grafo de cena"
 - Detalhes de *rendering* gerenciados automaticamente

[Sun 2003]



Introdução

A API Java 3D


- Programa Java 3D
 - Cria objetos Java 3D e posiciona-os em um grafo de cena
- Grafo de cena
 - Combinação de objetos 3D em uma estrutura de árvore
 - Especifica o conteúdo do universo virtual e como este é visualizado



Introdução

A API Java 3D

- Programas Java 3D podem ser escritos como
 - Aplicação
 - *Applet*
 - Suporte para Java 3D nos navegadores está disponível através de um *plugin* que pode ser obtido no site da Sun (java.sun.com)
 - Ou ambas
 - Classe *MainFrame*



Introdução

- Conceitos de Computação Gráfica
- Linguagem Java
- A API Java 3D
- **Histórico**
- Aplicações
- Instalação e Utilização



Introdução

Histórico

- Idéia
 - Construir uma API que fosse independente de plataforma
 - “Prima” de VRML
 - Alguns componentes familiares
 - Muitos componentes novos
- 1998

Introdução
Histórico

- Especificação da API Java 3D
 - Intel, Silicon Graphics, Apple e Sun Microsystems
 - Sun colocou a sua implementação para *download*
- Java 3D
 - Baseada em OpenGL e Direct3D (depende da plataforma na qual o programa é executado)



O diagrama mostra a arquitetura de dependências da API Java 3D. No topo, um retângulo amarelo rotulado 'Applet / Aplicação' depende de um retângulo amarelo rotulado 'Java 3D'. O 'Java 3D' depende de dois retângulos amarelos rotulados 'OpenGL' e 'Direct3D'. Ambos 'OpenGL' e 'Direct3D' dependem de um retângulo amarelo rotulado 'Hardware' na base.

Introdução


- Conceitos de Computação Gráfica
- Linguagem Java
- A API Java 3D
- Histórico
- **Aplicações**
- Instalação e Utilização

Introdução
Histórico

- Versões da API Java 3D
 - Java 3D 1.1 API
 - Implementação inicial
 - Java 3D™ 1.1.3 API
 - Java 3D™ 1.2.1 API
 - Nova implementação da arquitetura
 - Java 3D™ 1.3 API
 - Java 3D™ 1.3.1 API
 - Versão atual

Introdução
Aplicações


- Exemplos de aplicações onde Java 3D é usado
 - Desenvolvimento de jogos
 - Comércio eletrônico
 - Visualização de dados
 - Elaboração de interfaces



Introdução

Aplicações


- Desenvolvimento de jogos
 - Exemplos Java 3D
 - *Four by four*
 - Disciplina de Computação Gráfica [Tori 2002]
 - Objetivo: aprender conceitos de Computação Gráfica
 - Java 3D no lugar de OpenGL
 - Exemplos: War 3D e Enduro 3D



Introdução

Aplicações

- Visualização de dados
 - Já foram criados *toolkits* Java 3D para a visualização de dados
 - VisAD
 - SPLViz
 - ...



Introdução

Aplicações


- Comércio eletrônico
 - Visualização 3D dos produtos
 - Loja virtual
 - Representação 3D
 - Interação



Introdução

Aplicações

- Visualização de dados
 - Exemplo: VisAD (<http://www.ssec.wisc.edu/~billh/visad.html>)
 - *Visualization for Algorithm Development*
 - Visualização e análise interativa e colaborativa de dados numéricos




Introdução

Aplicações


- Exemplo: elaboração de interfaces
 - Configurador 3D
 - Usuário seleciona e aplica um material e verifica o resultado (Courtesy of TechniCon Corporation)

http://javaboutique.internet.com/resources/books/java3d/java3d2_5.html



Introdução


- Conceitos de Computação Gráfica
- Linguagem Java
- A API Java 3D
- Histórico
- Aplicações
- **Instalação e Utilização**



Introdução

Instalação e Utilização

- *Software* necessário para trabalhar com Java 3D
 - Gratuito
 - <http://java.sun.com>
- Primeiro passo:
 - Instalar o *Java™ 2 Platform, Standard Edition (J2SE™)*
<http://java.sun.com/j2se/downloads.html>
(versão 1.2 ou posterior)



Introdução

Instalação e Utilização

- Segundo passo:
 - Instalar a Java 3D API (versão atual 1.3.1)
 - Versão OpenGL ou DirectX (*Windows*)
<http://java.sun.com/products/java-media/3D/download.html>
(software e documentação)
- Navegador para executar exemplos
 - É necessário um *Plug-in*



Introdução

Instalação e Utilização

- Executar os exemplos
 - `c:\Arquivos de programas\jdk1.4.1_05\demo\java3d`
 - *Morphing*, Texto, Som, ...
- Escrever, compilar e executar programas
 - Comandos:
 - `javac FileName.java`
 - `java FileName`
 - Ou usar um ambiente de programação
 - Exemplo: BlueJ (<http://www.bluej.org>)



Introdução

Instalação e Utilização

- Desenvolvimento de programas Java 3D
 - Importar pacotes
 - `com.sun.j3d.utils.*;`
 - `javax.media.j3d.*;`
 - Grafo de Cena
 - Canvas3D



Referências

- [Bicho 2002] A. L. Bicho, L. G. da Silveira Jr, A. J. A. da Cruz e A. B. Raposo. **Programação Gráfica 3D com OpenGL, Open Inventor e Java 3D**. [REIC - Revista Eletrônica de Iniciação Científica. v. II, n. 1](http://www.sbc.org.br/reic/edicoes/2002e1/tutoriais/ProgramacaoGrafica3DcomOpenGLOpenInventorJava3D.pdf), março, 2002.
- [Barrilleaux 2001] J. Barrilleaux. **3D User Interfaces with Java 3D**. Manning Publications Co. 2001. 499 p.
- [Brown 1999] K. Brown. **Ready-to-run Java 3D**. New York, NY: John Wiley & Sons, 1999. 400 p.
- [Lathrop 1997] O. Lathrop. **The Way Computer Graphics Works**. Wiley Computer Publishing, 1997.



Referências

- [Sowizral 1998] H. Sowizral, K. Rushforth, M. Deering. **The Java™ 3D API Specification**. 2nd Edition. Addison-Wesley. 1998. 482 p.
- [Sun 2003] Sun Microsystems Java 3D Engineering Team. **Java 3D API Tutorial**. Disponível em <http://developer.java.sun.com/developer/onlineTraining/java3d/>. Acesso: setembro/2003.
- [Tori 2002] R. Tori, R. Nakamura. **Desenvolvimento de Jogos para Aprendizagem de Java 3D: Um Estudo de Caso**. Workshop de Jogos, 2002.
- [Walsh 2002] A. E. Walsh. **Java 3D: API Jump-start**. Upper Saddle River, NJ: Prentice Hall, 2002. 245 p.