Curso de Java

Unidad V

“Java para Graficación”

Rogelio Ferreira Escutia

Contenido

PARTE 1

1) Introducción a Java

Java - Historia

- Es un lenguaje que se originó en 1991 como parte de un proyecto de investigación ("Green Team", formado por Patrick Naughton, Mike Sheridan, y James Gosling) para desarrollar un nuevo lenguaje de programación llamado "Oak".
- Este lenguaje se enfocó para comunicar dispositivos caseros como televisiones y videocaseteras, los cuales pudieran compartir el mismo software pero utilizando diferentes microprocesadores.
- Este lenguaje no prospero en el mercado de dispositivos caseros, por lo que se cambió el mercado en el cual se emplearía, optando utilizarlo en el naciente Web, para el desarrollo de componentes multimedia en páginas web.
- Estas aplicaciones fueron denominadas "applets" y fue en ese entonces que se cambió el nombre por Java.

Java - Historia

- **“Green Team” de izquierda a derecha: *Al Frazier, Joe Palrang, Mike Sheridan, Ed Frank, Don Jackson, Faye Baxter, Patrick Naughton, Chris Warth, James Gosling, Bob Weisblatt, David Lavallee, and Jon Payne.***

- Java Technology: The Early Years, http://java.sun.com/features/1998/05/birthday.html, marzo 2004

Java - Historia

- **Dispositivo e interface del prototipo denominado *7 (Star Seven) con el cual se hizo una demostración de la plataforma Java (aún no denominada con este nombre).**

- Java Technology: The Early Years, http://java.sun.com/features/1998/05/birthday.html, marzo 2004

Plataforma Java

- La plataforma Java es una nueva forma de trabajar con computadoras, basado en el poder de las redes y en la idea de que el mismo software pueda correr en diferentes tipos de computadoras.
- La idea es crear un software que trabaje desde pequeños dispositivos hasta supercomputadoras, el único requisito es el dispositivo soporte la plataforma Java.
- En la actualidad, la plataforma Java es soportada por teléfonos, tarjetas inteligentes, electrodomésticos, etc.

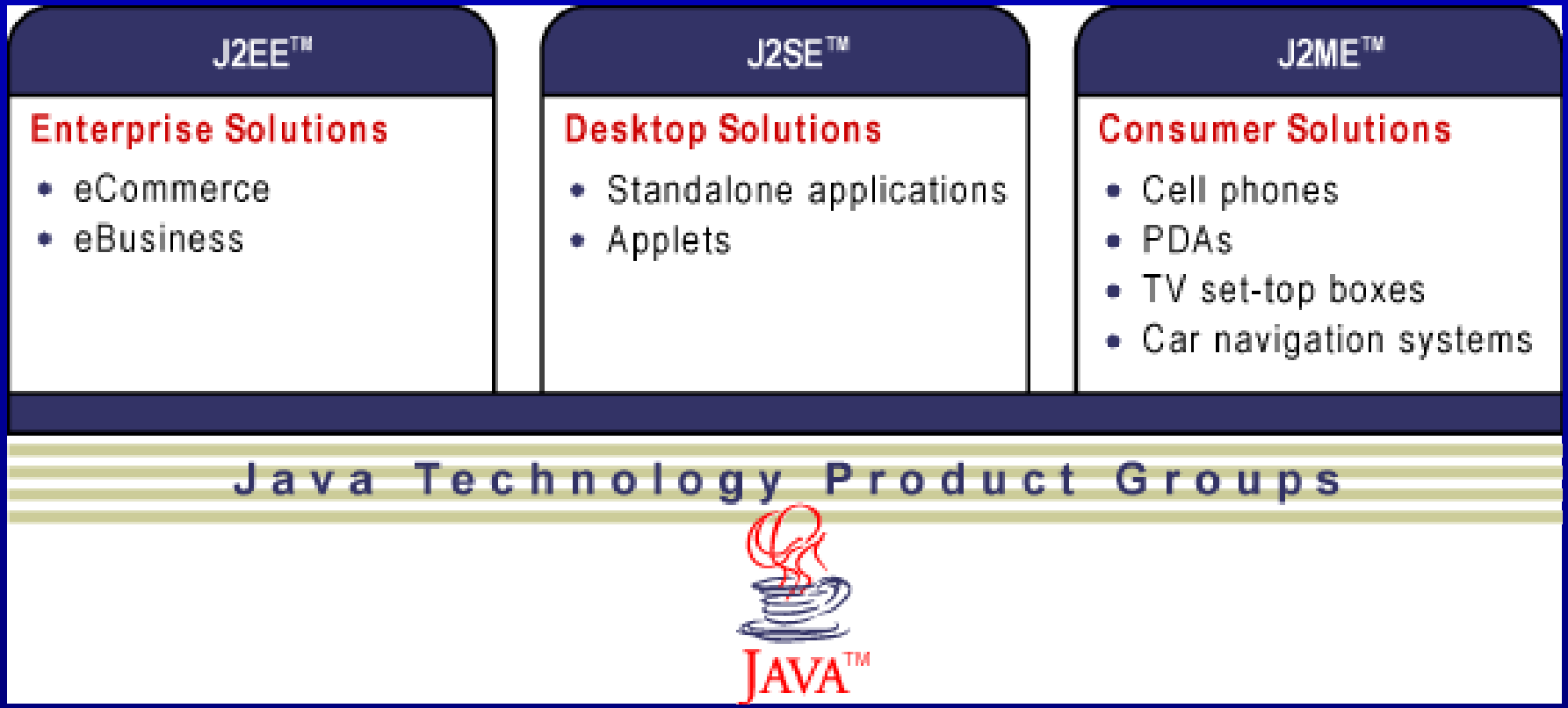
- What is the Java Platform, http://java.sun.com/java2/whatis/1996/, marzo 2004

Plataforma Java

- La plataforma SUN esta formado por los siguientes componentes:

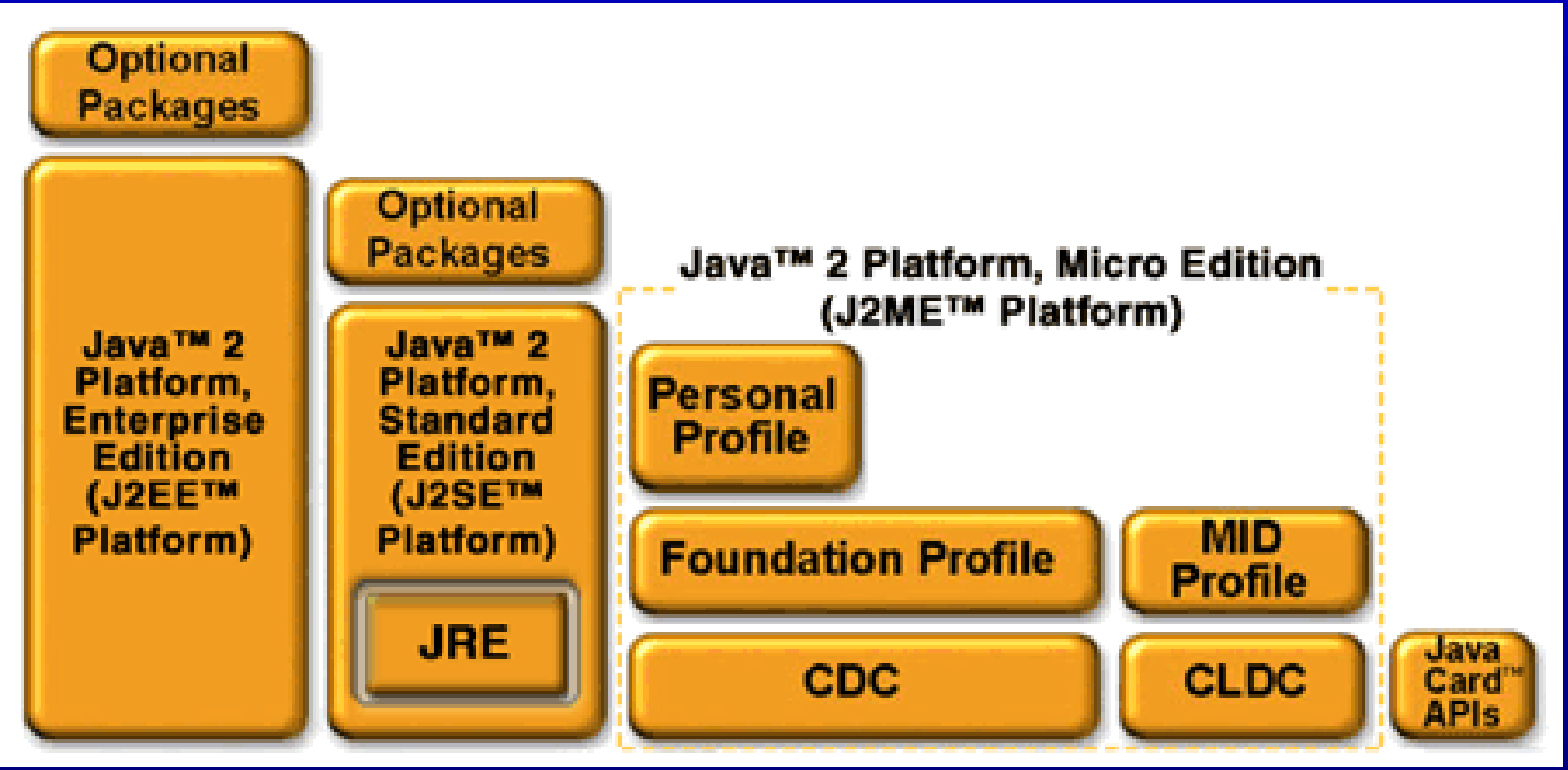
  J2SE: Aplicaciones de escritorio.
  J2EE: Aplicaciones empresariales.
  J2ME: Aplicaciones Móviles (Embedded Systems).
  Java Card: Tarjetas inteligentes.

Plataforma Java



- SUN - http://www.sun.com – marzo 2004

Plataforma Java



- Java 2 Platform, http://java.sun.com/java2/whatis/index.html, marzo 2004

Java Card

- Una tarjeta inteligente es una tarjeta del tamaño de una tarjeta de crédito con un circuito integrado en su interior.
- Un circuito contiene un microprocesador y memoria, los cuales le dan la habilidad a la tarjeta de procesar y almacenar información.
- En el caso de la plataforma Java Card, las aplicaciones en forma de byte-code son cargadas en la zona de memoria, para después ser ejecutadas por la máquina virtual.
- El código ejecutable es independiente de la plataforma, por lo que cualquier tarjeta que tenga incorporada una máquina virtual podrá ejecutar el código.

• Java 2 Platform, http://java.sun.com/java2/whatis/index.html, marzo 2004

Java - Características

- Java es un lenguaje de Programación Orientado a Objetos (Object-Oriented Programming), por lo que se busca crear objetos, o piezas de código autónomo, que pueda interactuar con otros objetos para resolver un problema.

- El lenguaje de programación Java fue desarrollado para tener las siguientes características:

  1) Orientado a Objetos.
  2) Distribuido.
  3) Simple.
  4) Multihilo.
  5) Seguro.
  6) Independiente de la plataforma.

2) Instalación de Java y Java 3D

Instalación – Java (Windows)

- **Bajar Java de:**

  http://java.sun.com

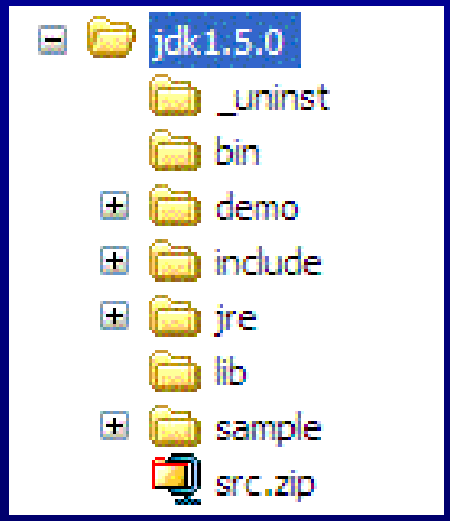
- **Se baja el archivo (descarga gratuita):**

  jdk-1_5_0-windows-i586.exe

- **Se ejecuta el archivo anterior y se instala en el directorio:**

  C:\jdk1.5.0

- **El directorio queda:**



Prueba – Java (Windows)

- Para probar la instalación se cambia al directorio bin y se corre la siguiente instrucción, la cual nos sirve para detectar la versión de Java que esta instalada:

  C:\>cd jdk1.50 C:\jdk1.54.0>cd bin
  C:\jdk1.5.0>java –version
  java version "1.5.0“
  Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0-b64)
  Java HotSpot(TM) Client VM (build 1.5.0-b64, mixed mode, sharing)

- Si se observa lo anterior, significa que ya esta instalado el Java. Se Requiere agregar agregar una variable de entorno, para que las aplicaciones Java pueden ser compiladas desde cualquier directorio del disco duro (en Windows XP). Para ello se selecciona "Panel de Control"/"Rendimiento y Mantenimiento"/"Sistema"/"Variables de Entorno"/"Nueva". En ese menu se escribe lo siguiente:

  Nombre de la variable: PATH
  Valor de la variable: C:\jdk1.5.0\bin

Programa – Java (Windows)

- En un editor de textos teclear el siguiente código bajo el nombre hola.java

```
/*      hola_texto.java */
public class hola_texto
{
   public static void main (String[] args)
     {System.out.println("Hola");}
}
```

Se compila el código objeto con:

```
> javac hola_texto.java
```

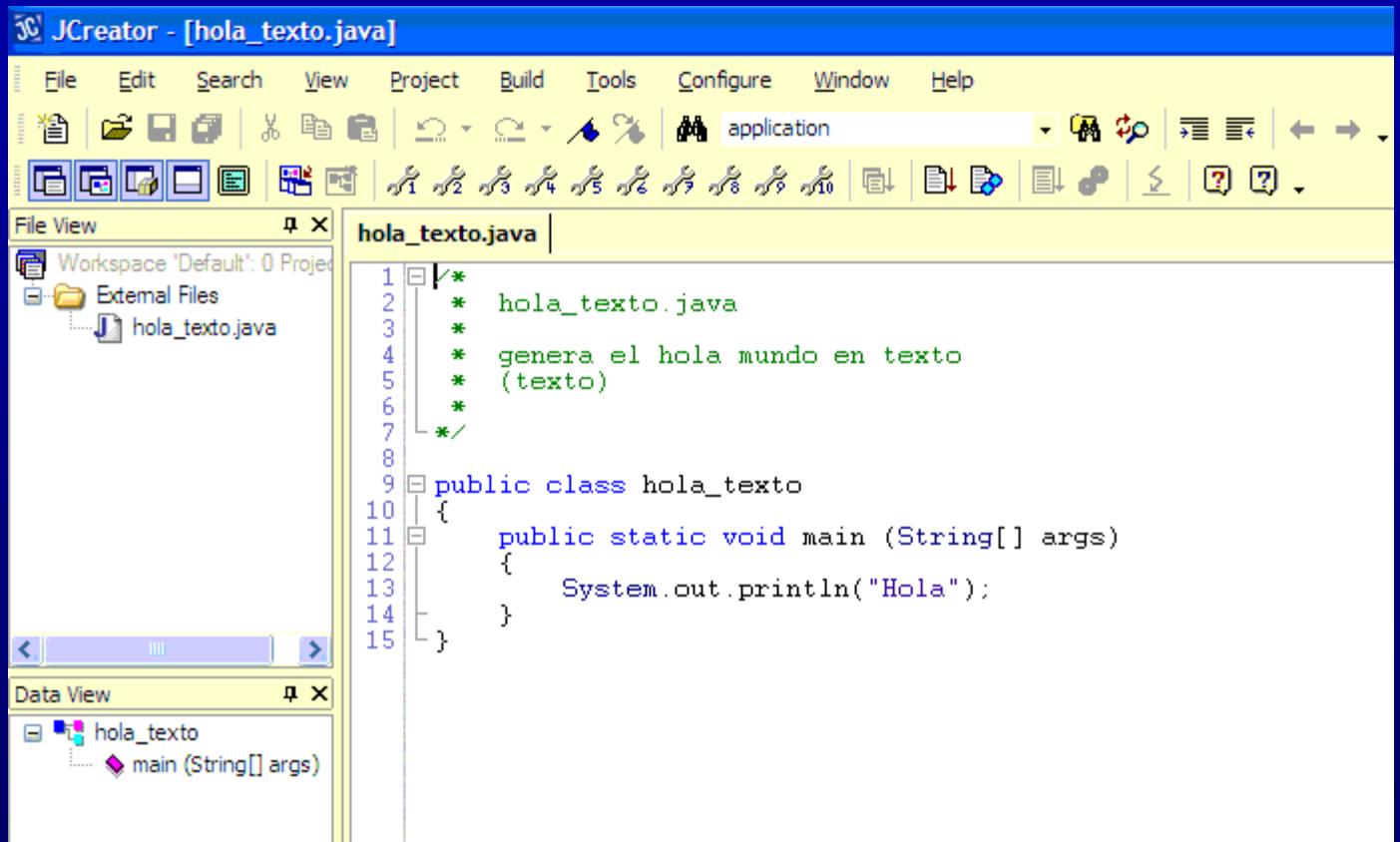
Para correr el programa y el observar el resultado en modo texto se ejecuta lo siguiente:

```
>  java hola_texto
```

En la línea anterior NO TECLEAR java hola_texto.class

Entorno – Java (Windows)

- Se recomienda la instalación de alguno entorno de programación.
- Se utiliza el software libre JCreator, el cual se descarga de:
  http://www.jcreator.com
- Con este software se puede editar y correr programas en java de una manera fácil.



Instalación – Java 3D (Windows)

- Se baja de:
  http://www.sun.com
- El arhivo para windows que se baja de internet se llama:
  java3d-1_3_1-windows-i586-directx-sdk.exe
- Se ejecuta y se genera el siguiente directorio:
  C:\j2sdk1.4.0_01\demo\java3d
- para probar un programa con Java3D nos cambiamos al siguiente directorio:
  C:\j2sdk1.4.0_01\demo\java3d\Sound
- y corremos el archivo:
  SimpleSounds.java
- se abrirá el navegador y mostrará un cubo girando, y emitirá sonidos en inglés, con esto queda comprobado que esta corriendo Java 3D

3) Conceptos sobre Java 3D

Java 3D

- **Filosofía de Java:**

  Write once, run anywhere

- ***Filosofía de Java 3D***

  Think objects . . . **not vertices**
  Think content . . . **not rendering process**

Java 3D

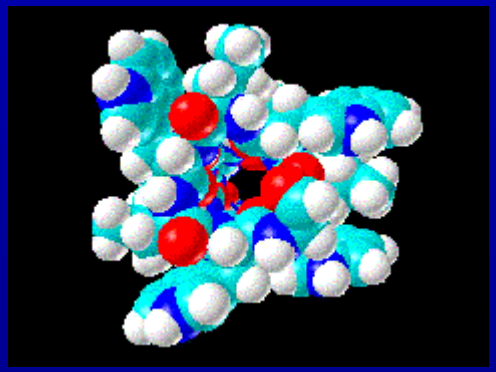
- Java 3D es una API de alto nivel para construcción de modelos 3D interactivos basados en lenguaje Java.

- Características:

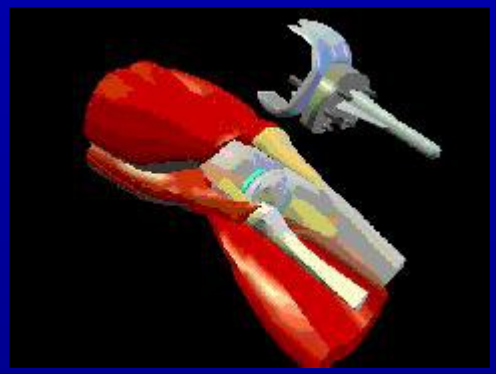
  Múltiples plataformas, uso de múltiples dispositivos de graficación y de entrada de datos.
  Dibujo de datos vía OpenGL/Direct3D
  Utiliza aceleración 3D en Hardware cuando esta disponible.
  Optimización de renderización.
  Creación de aplicaciones en la web vía applets.

Java 3D

- **Aplicaciones**



Graficación



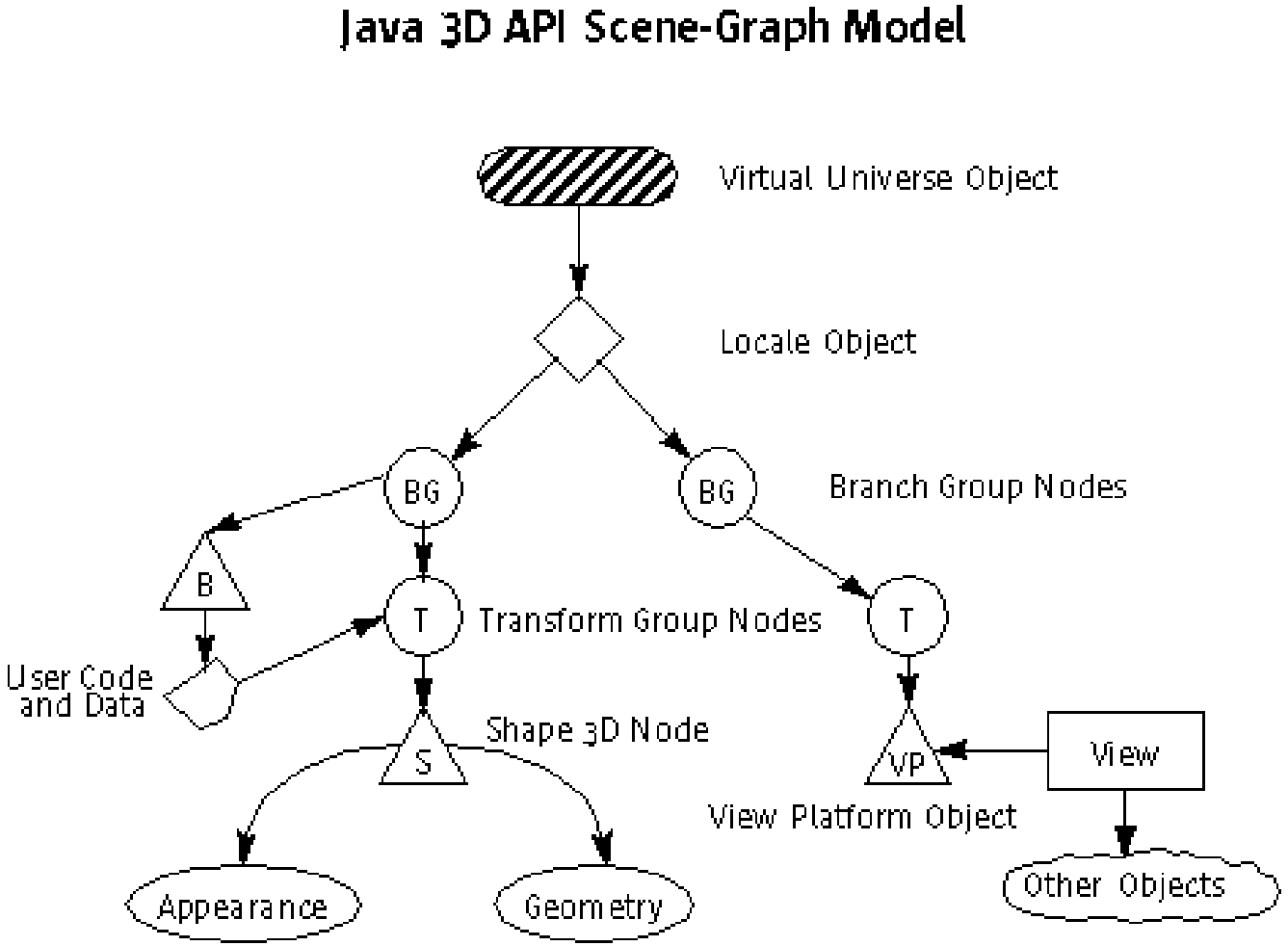
Medicina

CAD

Java 3D

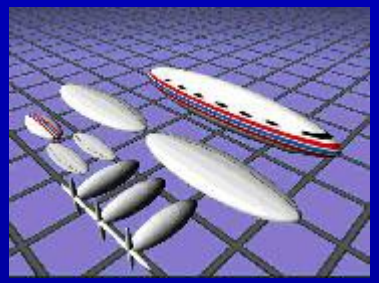
- Una escena (scene graph) es un árbol (family tree) que contiene datos de escenas, los cuales pueden ser:
- Hijos (Children), compuestos por figuras, luces, sonidos, etc.
- Familiares (Parents) que son grupos de Hijos y otros familiares.
- Por medio de los anteriores se genera un grupo jerárquico (hierarchical) de figuras.
- La aplicación crea una escena usando clases de Java 3D y métodos.
- Java 3D renderiza la escena dentro de la pantalla.

API Java 3D

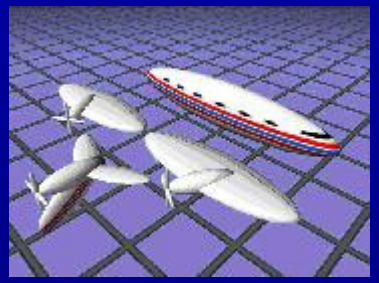

Java 3D API Scene-Graph Model
Virtual Universe Object
Locale Object
BG
BG
Branch Group Nodes
B
T
Transform Group Nodes
T
User Code and Data
Shape 3D Node
S
VP
View
View Platform Object
Other Objects
Appearance
Geometry

Java 3D - Escenas

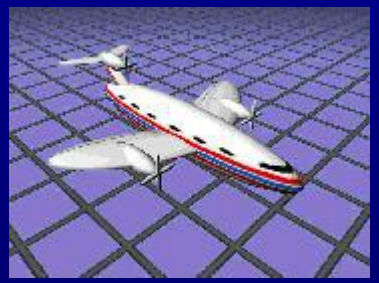
- **Construcción de una escena**



Definir componentes



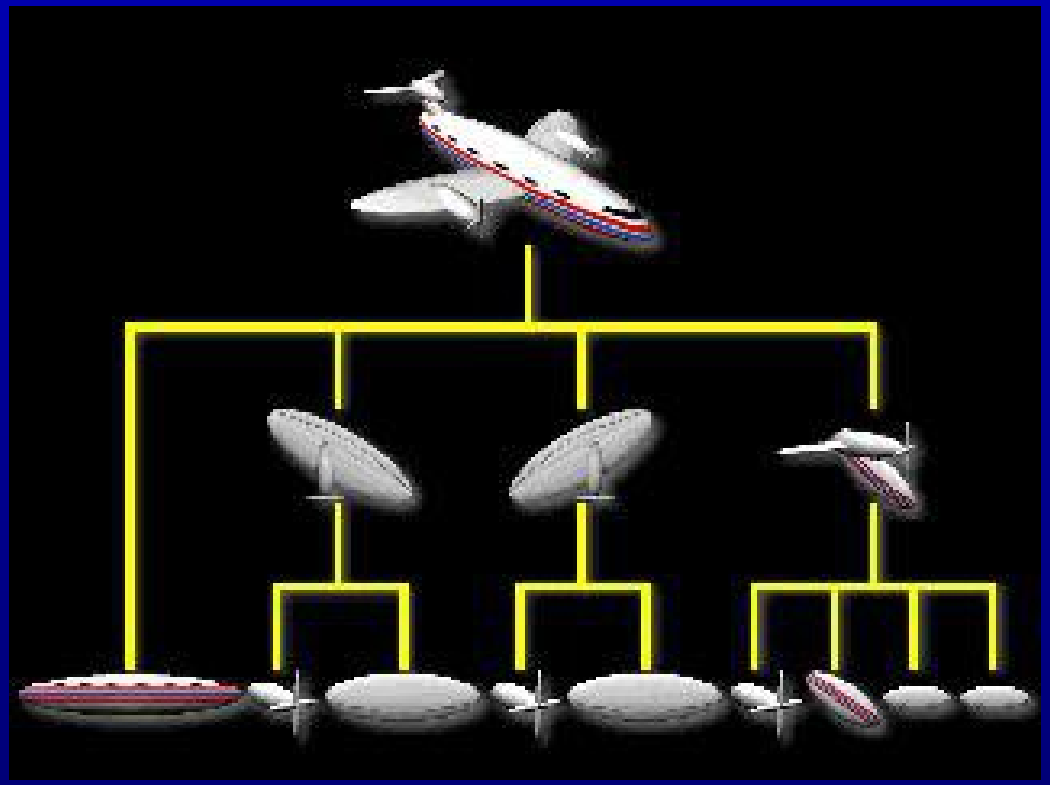
Ensamble de componentes



Objeto final

Java 3D - Escenas

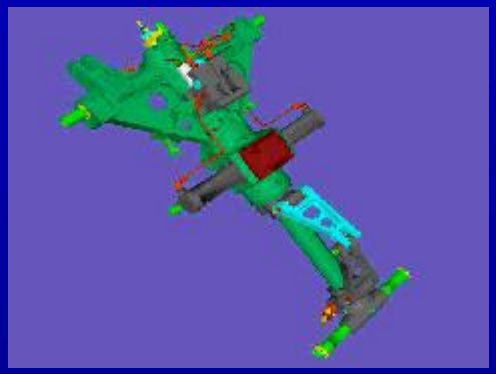
- **El diagrama de escena muestra mas claramente los componentes del objeto creado.**



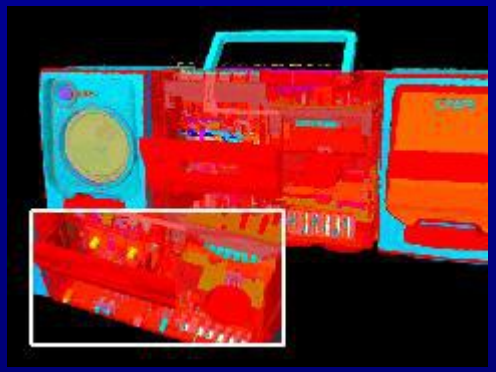
- Introduction to Programming with Java 3D, Sowizral y Nadeau, http://www.sdsc.edu/~nadeau, mayo 2005

Java 3D - Escenas

- **Construcción de objetos**



Tren de Aterrizaje
192 formas



Grabadora
11,000 componentes

Java 3D - Escenas

- Una escena esta construida por los siguientes componentes:

- Formas (geometría y apariencia)
- Grupos y Transformaciones
- Luces
- Niebla y Fondos
- Sonidos y efectos de sonido
- Comportamientos
- Puntos de vista

Java 3D - Escenas

- **Terminología de Escenas:**
- **Nodo (node): es un componente de una escena (scene graph).**
- **Nodos Hojas (leaf nodes): nodos sin hijos. Pueden contener formas, luces, sonidos, comportamientos, etc.**
- **Nodos Grupo (group nodes): nodos con hijos. Pueden contener transformaciones.**
- **Nodo Componente (node component): un conjunto de atributos para un nodo (geometría de una forma, color de una forma, sonidos a ejecutar, etc).**
- **Un conjunto de nodos hoja, nodos grupo, nodos componentes forman una jerarquía de clases de Java 3D (Java 3D class hierarchy)**

Construcción de Escenas

- **Construir nodos por instanciación de clases Java 3D:**

```
Shape3D mi_forma = new Shape3D( mi_geometria, mi_apariencia );
```

- **Modificar nodos llamando a los métodos de una instancia:**

```
mi_forma.setAppearance( nueva_apariencia );
```

- **Construcción de nodos grupo:**

```
Group mi_grupo = new Group();
mi_grupo.addChild( mi_forma );
```

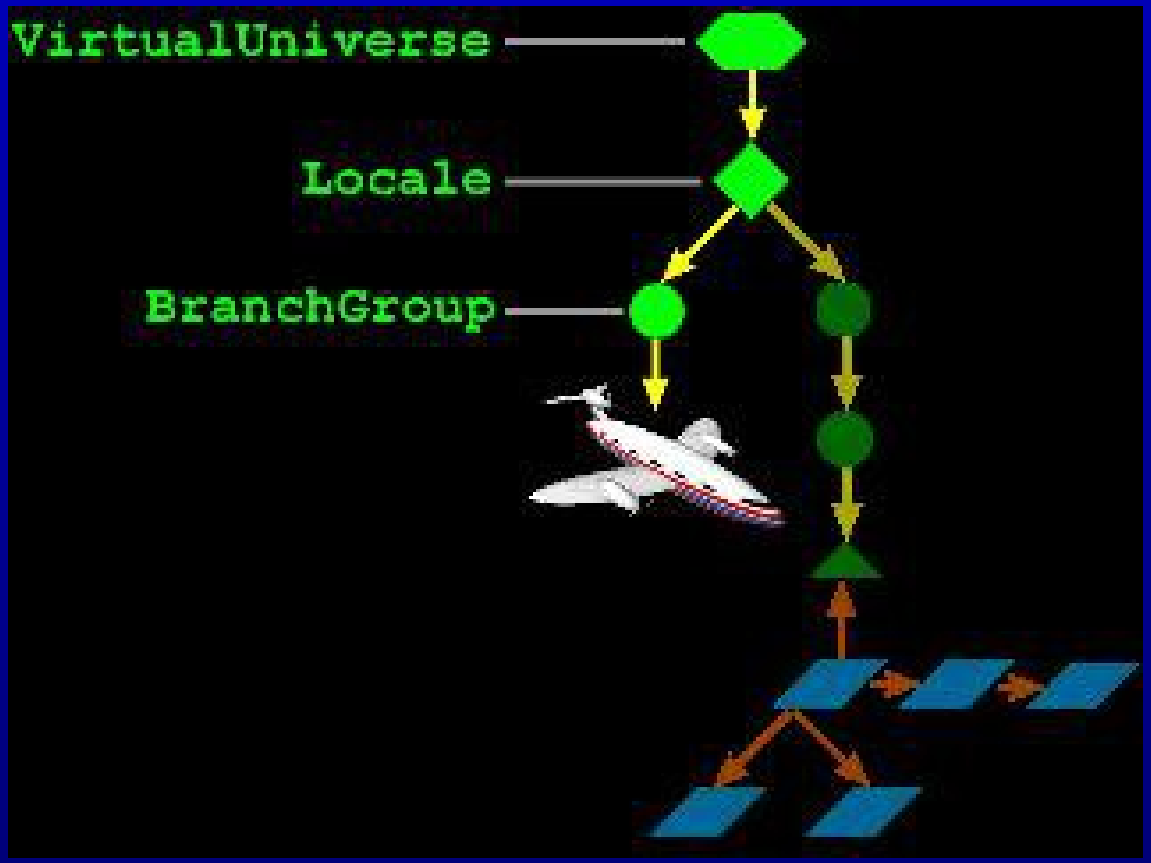
Universo

- Se requiere ensamblar partes de la escena final por medio de escenas (scene graph).
- Ensamblarlas en un contenedor común, denominado universo virtual (virtual universe).
- Un universo virtual es una colección de escenas. Por lo general se utiliza un solo universo por aplicación.
- Local (locale) es una posición en el universo en el cual se inserta la escena. Por lo general se utiliza un Local por Universo.
- Rama (Branch Graph) es parte de una escena. Normalmente se utilizan varias ramas por Local.

Ramas

- Una escena común está dividida en 2 tipos de ramas:
- Rama de Contenido (Content Branch) que contiene formas, luces, etc. Normalmente se utilizan varias ramas por Local.
- Ramas de Vista (View Branch) se utiliza para controlar la información de la vista a observar del objeto 3D. Por lo general se utiliza una rama de vista por Universo.

Diagrama de Universo



- Introduction to Programming with Java 3D, Sowizral y Nadeau, http://www.sdsc.edu/~nadeau, mayo 2005

Construcción

- **Construcción de un Universo:**

```
VirtualUniverse mi_universo = new VirtualUniverse( );
```

- **Construcción de un Local:**

```
Locale mi_local = new Locale( mi_universo );
```

- **Construcción de una rama de grupo:**

```
BranchGroup mi_rama = new BranchGroup( );
```

Construcción

- **Construir nodos y grupos de nodo:**

```
Shape3D mi_forma = new Shape3D( mi_geom mi_apariencia );
Group mi_grupo = new Group( );
my_grupo.addChild( mi_forma );
```

- **Agregarlos a la rama:**

```
mi_rama.addChild( mi_grupo );
```

- **Agregar la rama al local:**

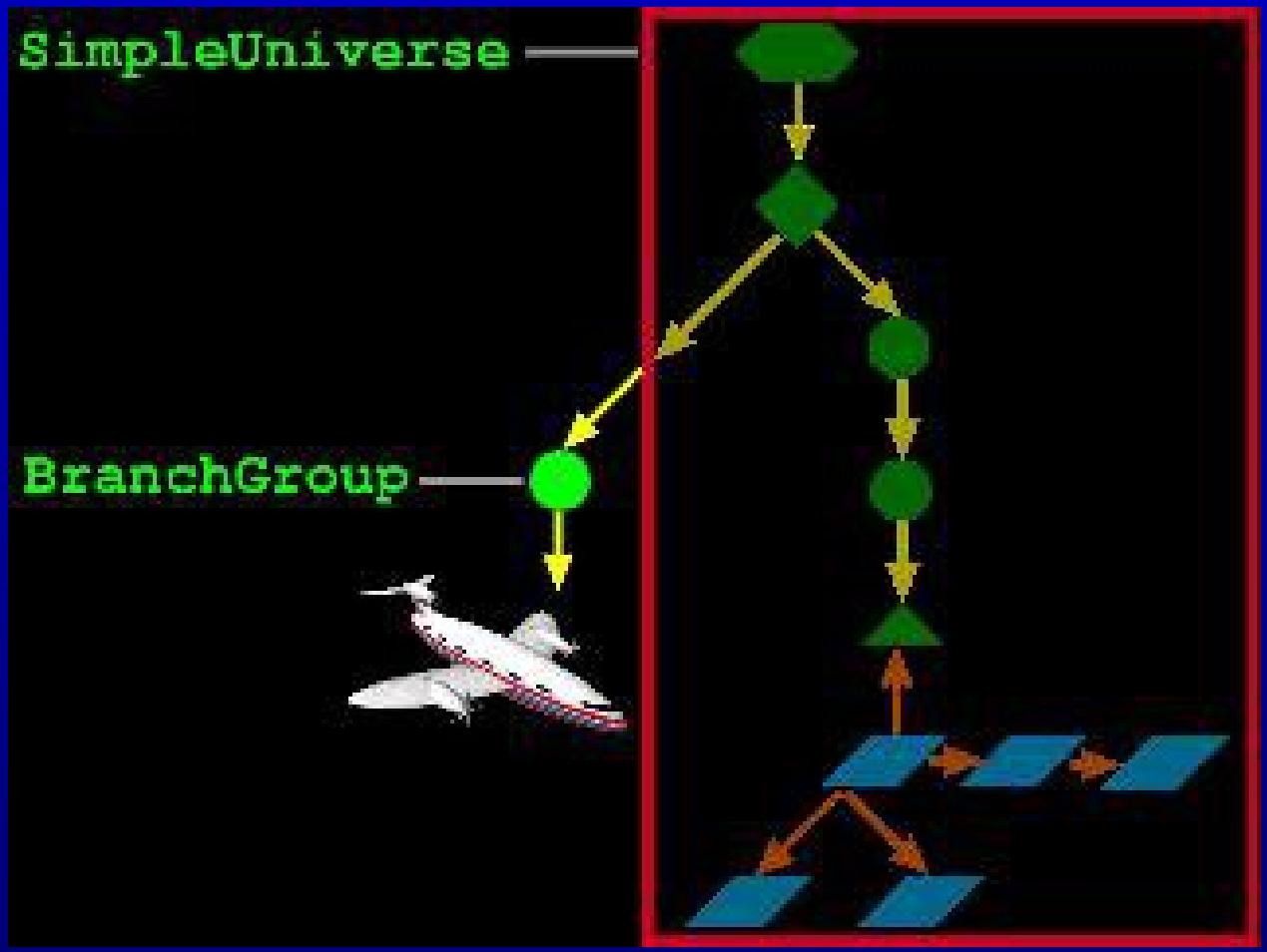
```
mi_local.addBranchGraph( mi_rama );
```

Universo Simple

- La utileria SimpleUniverse es un arreglo común de un Universo, un Local y clases de vistas (encapsula una superestructura):

```
SimpleUniverse mi_simple = new
    SimpleUniverse( mi_Canva );
mi_simplem.addBranchGraph( mi_rama );
```

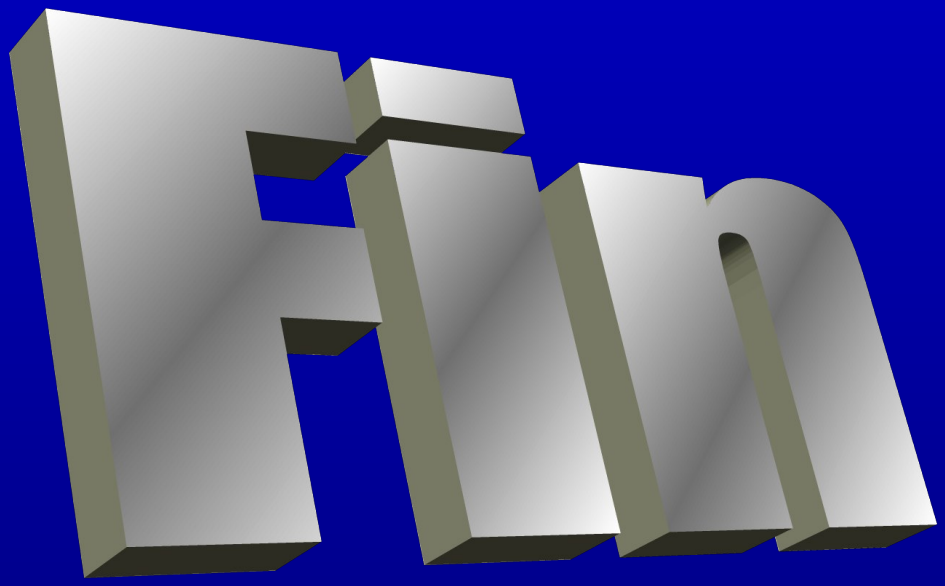
Universo Simple



- Introduction to Programming with Java 3D, Sowizral y Nadeau, http://www.sdsc.edu/~nadeau, mayo 2005

PARTE 2

PARTE 3



Unidad V – Java para Graficación