

I Jornadas de Software Libre de Castilla-La Mancha
Escuela Politécnica Superior de Albacete
9-12 diciembre 2004

Java 3D y Looking Glass

Desarrollos open source de SUN

Víctor López Jaquero

José Pascual Molina

Departamento de Informática, UCLM

{ victor | jpmolina } @info-ab.uclm.es

Contenidos

- **Primera parte: Java 3D**
 - A cargo de Víctor López
- **Segunda parte: Looking Glass**
 - A cargo de José Pascual Molina

Primera parte

Java 3D



¿Qué es Java 3D?

- Java 3D es un API orientado a objetos para el lenguaje Java de SUN para la programación de aplicaciones gráficas tridimensionales que permite
 - Construir objetos 3D
 - Visualizarlos
 - Controlar su comportamiento

<http://java.sun.com/products/java-media/3D/>

Ventajas de Java 3D

- Es una API de código abierto
- Interfaz de alto nivel
- La visualización se basa en las APIs:
 - OpenGL
 - DirectX (sólo Windows)
- Programación 3D independiente de la plataforma (Linux, Solaris, Windows, Mac Os X, Irix 6.5)
- Existen cargadores para importar objetos en el API
 - VRML97, 3D Studio, Lightwave, Quake 2, ...
- La escena se construye creando un grafo
- La aplicación 3D puede ser un Applet.

Ventajas de Java 3D

➤ Ejemplo de aplicación



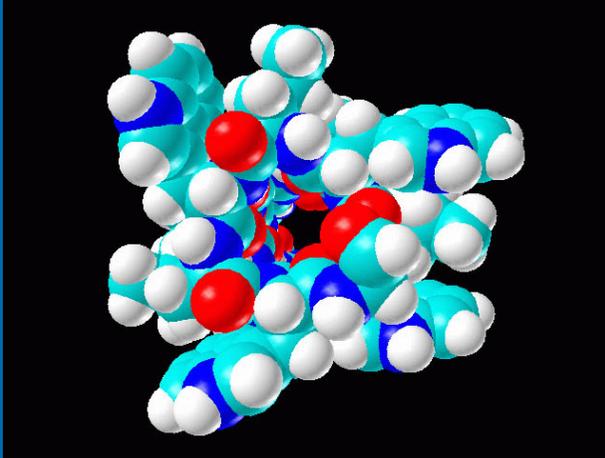
Desventajas de Java 3D

- Hasta ahora Sun sólo proporcionaba apoyo para la versión de Solaris y la de Windows.
- El API oculta detalles de cómo se visualiza la escena.
- Los componentes de Java 3D son pesados (*heavyweight*).
- Java 3D es una extensión que no forma parte de la distribución estándar de Java.
- No es tan rápido como una aplicación en código nativo en OpenGL o DirectX.

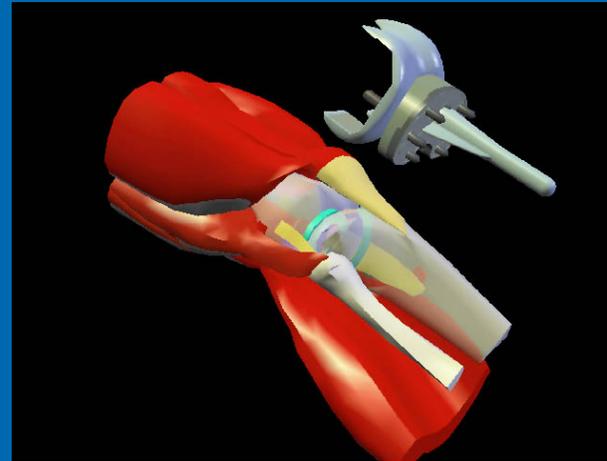
Aplicaciones de Java 3D

- Visualización científica
- Visualización de información
- Visualización médica
- Sistemas de Información Geográfica (GIS)
- Diseño Asistido por Ordenador (CAD)
- Animación
- Educación

Aplicaciones de Java 3D



Visualización científica



Educación en medicina



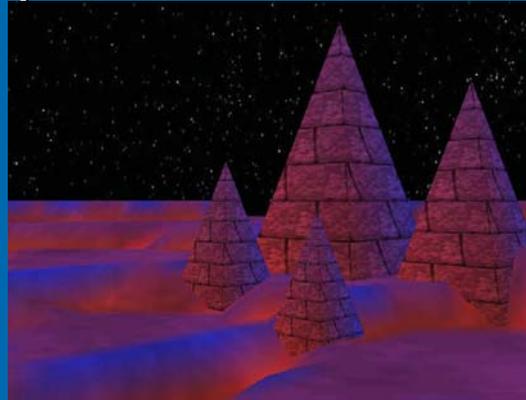
CAD



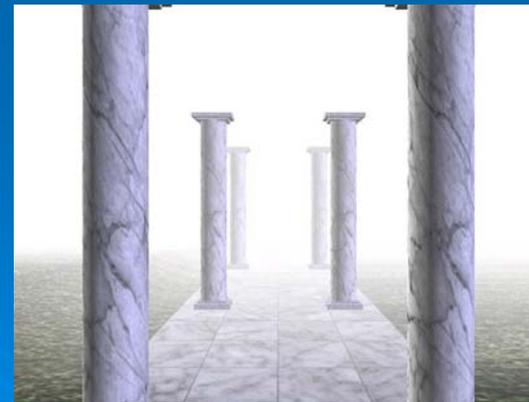
Animación

Características de Java 3D

- De los objetos se puede controlar su color y textura.

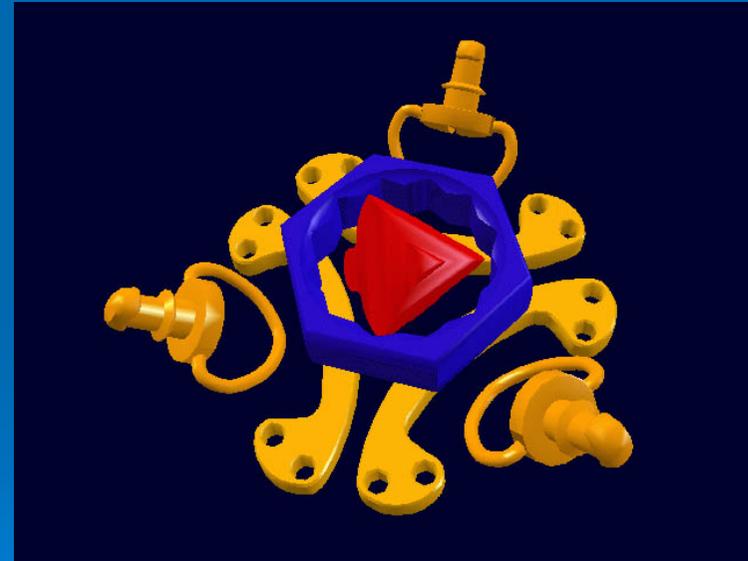
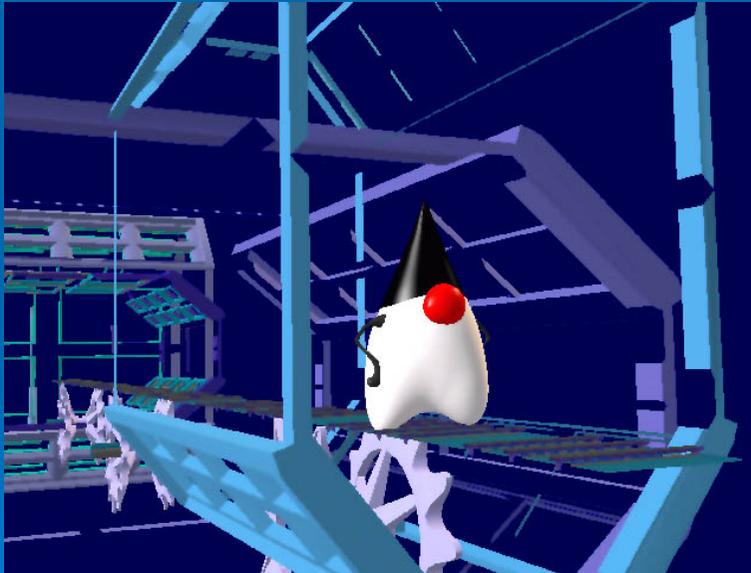


- Permite añadir luces y efectos de niebla.



Características de Java 3D

- De los objetos se puede controlar su tamaño, posición y orientación y cómo dichos atributos evolucionan en el tiempo:

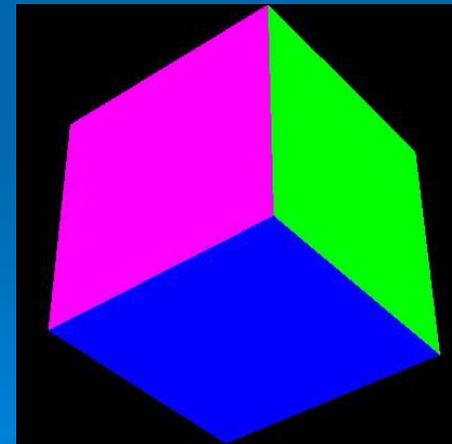
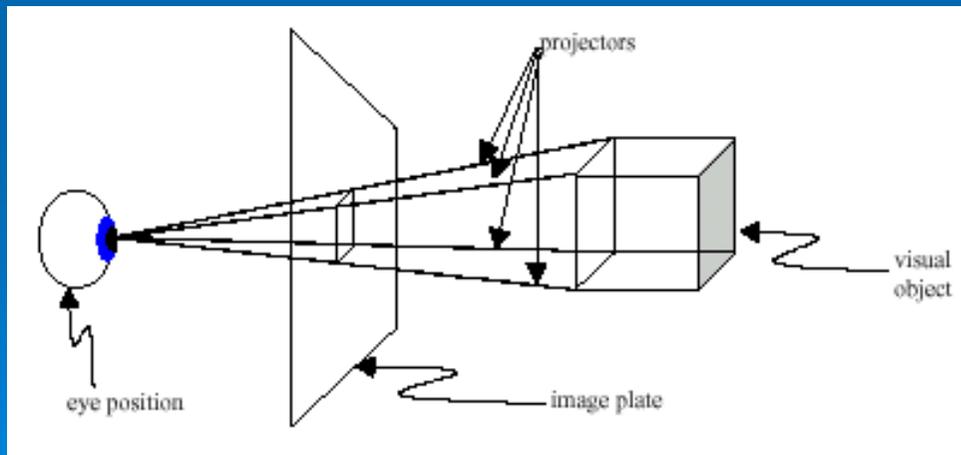


Modos de Programación de Java 3D

- Modo conservador (*retained*)
 - El programador no puede acceder al API que está usando Java 3D para la visualización (OpenGL, DirectX).
- Modo inmediato (*immediate*)
 - Permite al programador interactuar con el API subyacente.
- Modo mixto (*mixed*)
 - Permite mezclar el modo conservador y el modo inmediato.

El Grafo de Escena

- Las escenas 3D visualizadas en Java 3D se describen mediante el grafo de la escena.
- El grafo de la escena se compone de:
 - Subgrafo de contenidos (los objetos que se visualizan en sí)
 - Subgrafo de vista (configuración de cómo se visualiza el subgrado de contenidos)



El Grafo de la Escena

- Cuando se añade una rama al grafo en un nodo que sea descendiente de un nodo *Locale*, ésta pasa a estar viva (*Live*), mientras que si se quita del árbol, la rama deja de estar viva
 - Sólo se visualizan las ramas vivas.
- Todos los nodos del grafo de escena contienen unos bits (capabilities) que determinan qué propiedades del nodo pueden ser modificadas.
- El grafo de escena puede ser “compilado” para ser optimizado, teniendo en cuenta los bits de capacidades.
- Las capacidades son por defecto de sólo lectura cuando un nodo está vivo o compilado.

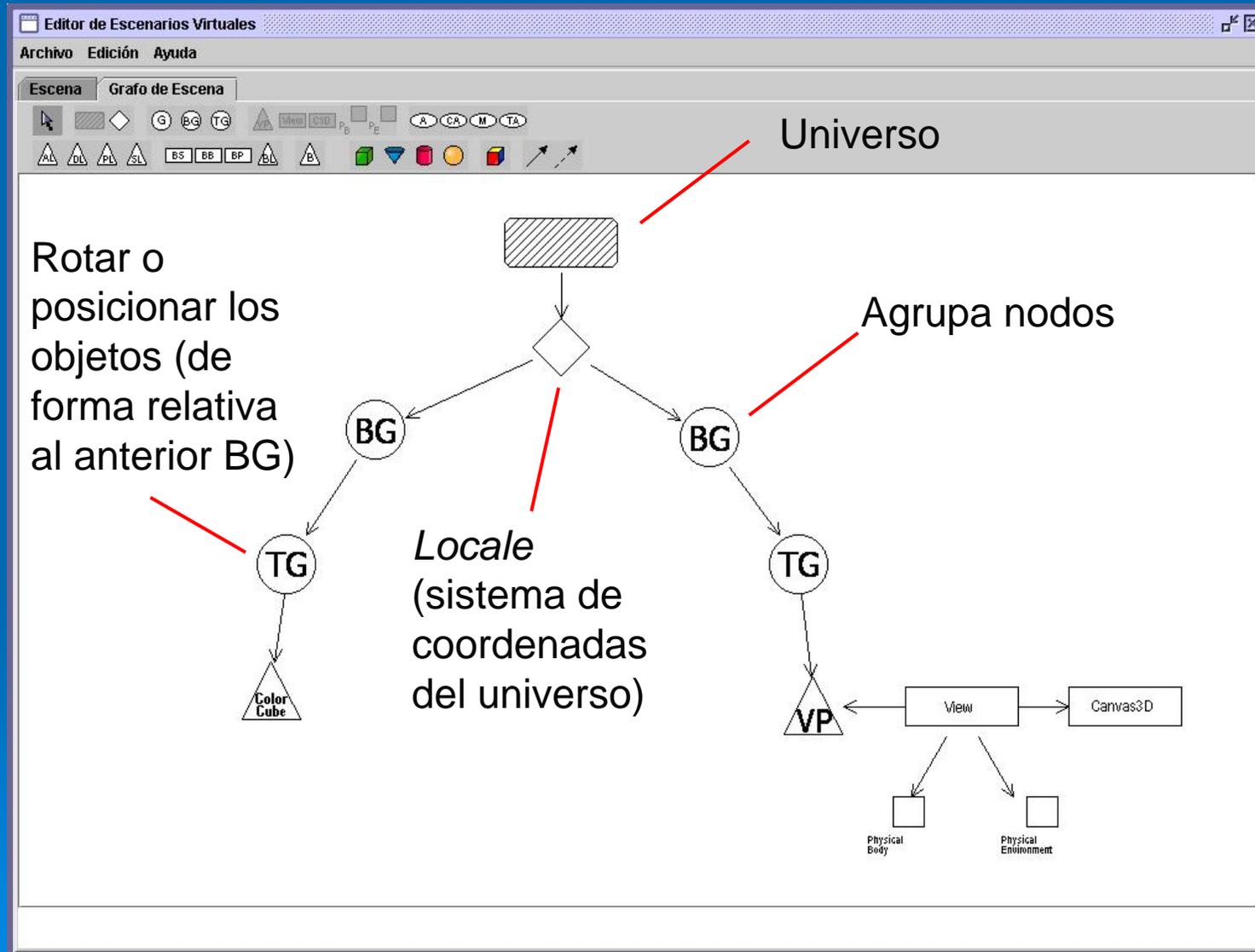
El Grafo de Escena

➤ Elementos del grafo de escena

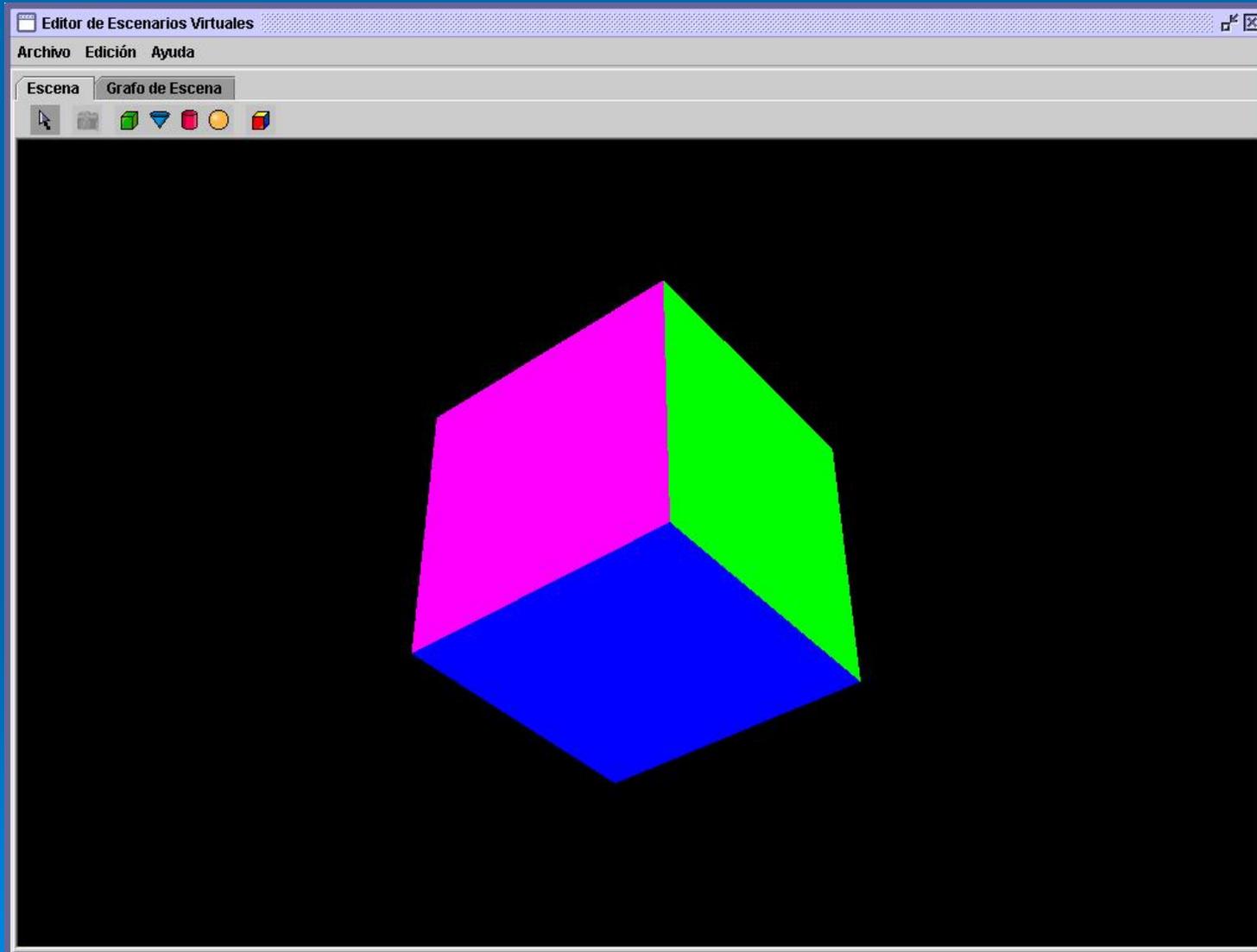
- Grupos
 - BranchGroup
 - TransformationGroup
 - Switch
 - SharedGroup
- Individuales
 - Enlaces a SharedGroups (links)
 - Shape3D (cubo, cubo de colores, cono, esfera, ...)
 - Luces (AmbientLight, SpotLight, ...)
 - Comportamientos
 - Sonidos
 - Efectos de niebla (*fog*)
 - Imágenes de fondo (*background*)



El Grafo de Escena



El Grafo de Escena



El Grafo de Escena

➤ Ejemplo de código de una escena simple

```
import java.applet.Applet;
import java.awt.BorderLayout;
import java.awt.Frame;
import com.sun.j3d.utils.applet.MainFrame;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.picking.behaviors.*;
import com.sun.j3d.utils.universe.*;
import javax.media.j3d.*;
import javax.vecmath.*;

public class Simple extends Applet {

    private Locale locale;
    private Canvas3D canvas3D;

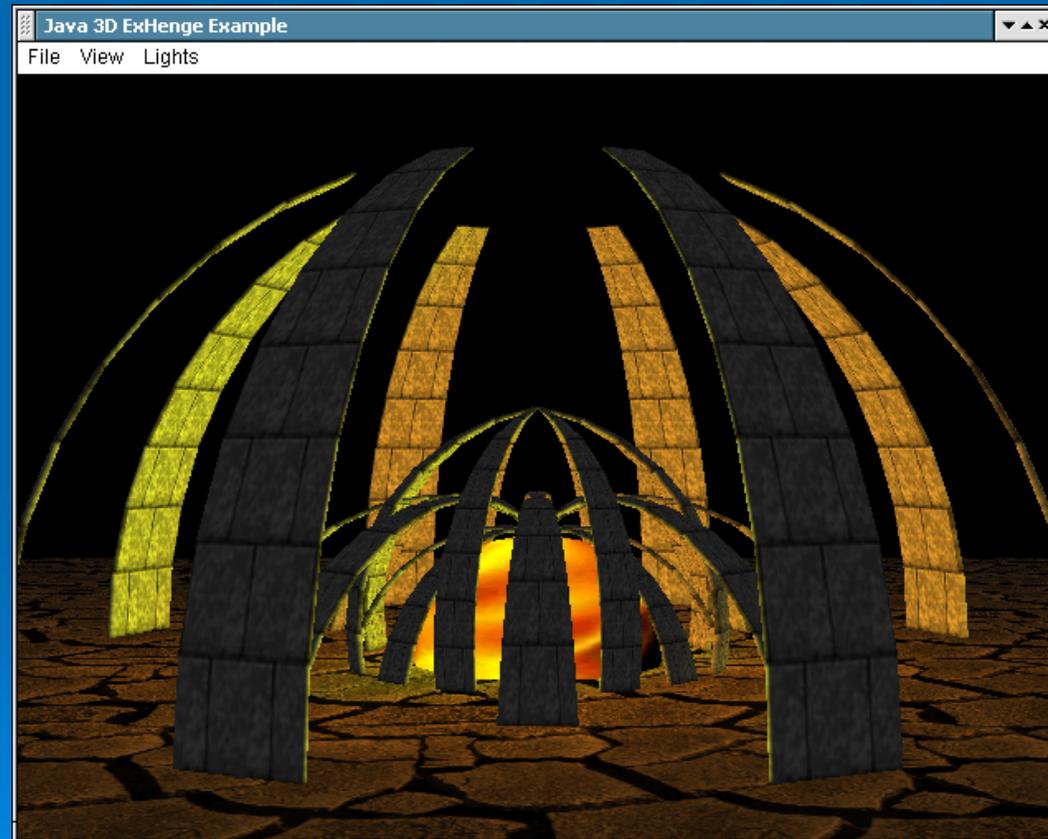
    public Simple() {
        setLayout(new BorderLayout());
        createViewBranchGraph();
        createContentBranchGraph();
        add("Center", canvas3D);
    }
}
```

Hágase la Luz

- Tipos de nodos de iluminación
 - AmbientLight: ilumina a todos los objetos de la escena desde todas las direcciones.
 - DirectionalLight: posicionadas en el “infinito”, e iluminan en una dirección concreta.
 - PointLight: irradia luz desde un punto concreto del espacio en todas direcciones.
 - SpotLight: irradia luz desde un punto concreto del espacio en una dirección concreta y con un haz de luz de un tamaño determinado (como un foco en teatro).
- Los nodos de iluminación actúan dentro del **área de influencia** que se les asigne (*InfluencingBounds*).

Hágase la Luz

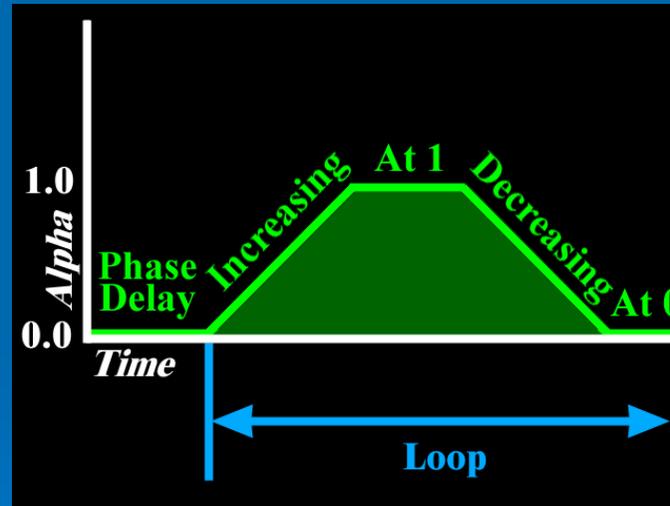
➤ Ejemplo ExHenge



... Y Llegó el Movimiento

➤ Comportamientos

- Permiten añadir animación a las escenas de una forma sencilla.
- Los valores alfa describen la dinámica del comportamiento.



- Los comportamientos se asocian a los nodos que modifican.

... Y Llegó el Movimiento

➤ Comportamientos

- Permiten añadir animación a las escenas de una forma sencilla.
- Interpolator (interpoladores)
 - ColorInterpolator
 - PathInterpolator (PositionPathInterpolator, ...)
 - PositionInterpolator
 - RotationInterpolator
 - ScaleInterpolator
 - TransparencyInterpolator

... Y Llegó el Movimiento

➤ Comportamientos (II)

- MouseBehavior
 - MouseRotate
 - MouseTranslate
 - MouseZoom
- PickMouseBehavior
 - PickRotateBehavior
 - PickTranslateBehavior
 - PickZoomBehavior

... Y Llegó el Movimiento

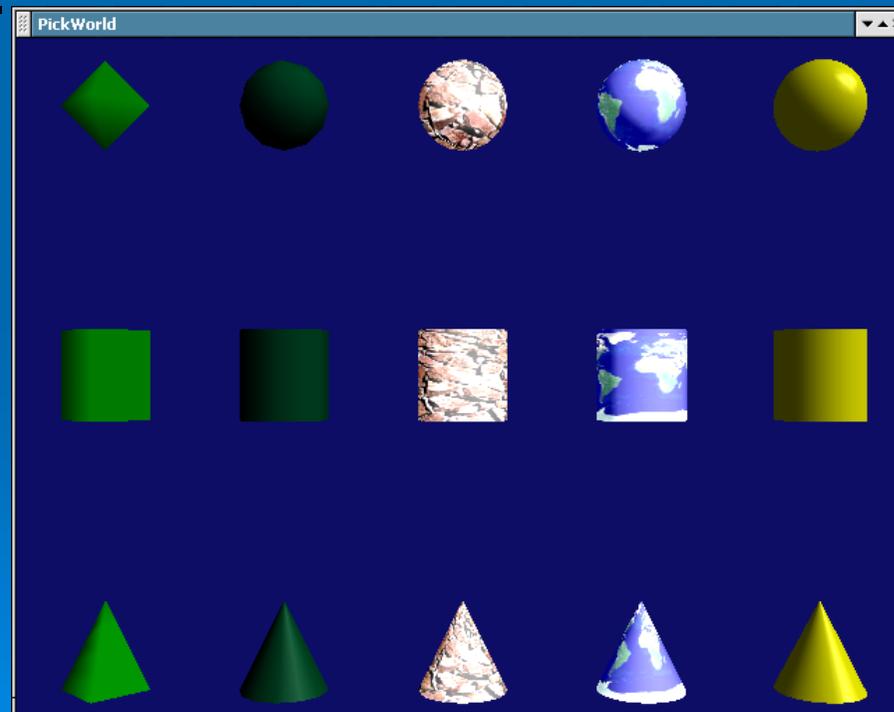
➤ Comportamientos (III)

- Un comportamiento se activa cuando
 1. Los objetos sobre los que se aplican entran en el área de acción definida para el comportamiento (*SchedulingBound*)
 - Ej. El objeto sobre el que se aplica está a menos de 10 metros del observador.
 2. Se cumple su condición de activación (*WakeUpCondition*)
 - Ej. Cuando pasen 10 segundos.

... Y Llegó el Movimiento

➤ Comportamientos

- Ejemplo de selección e interacción con objetos.

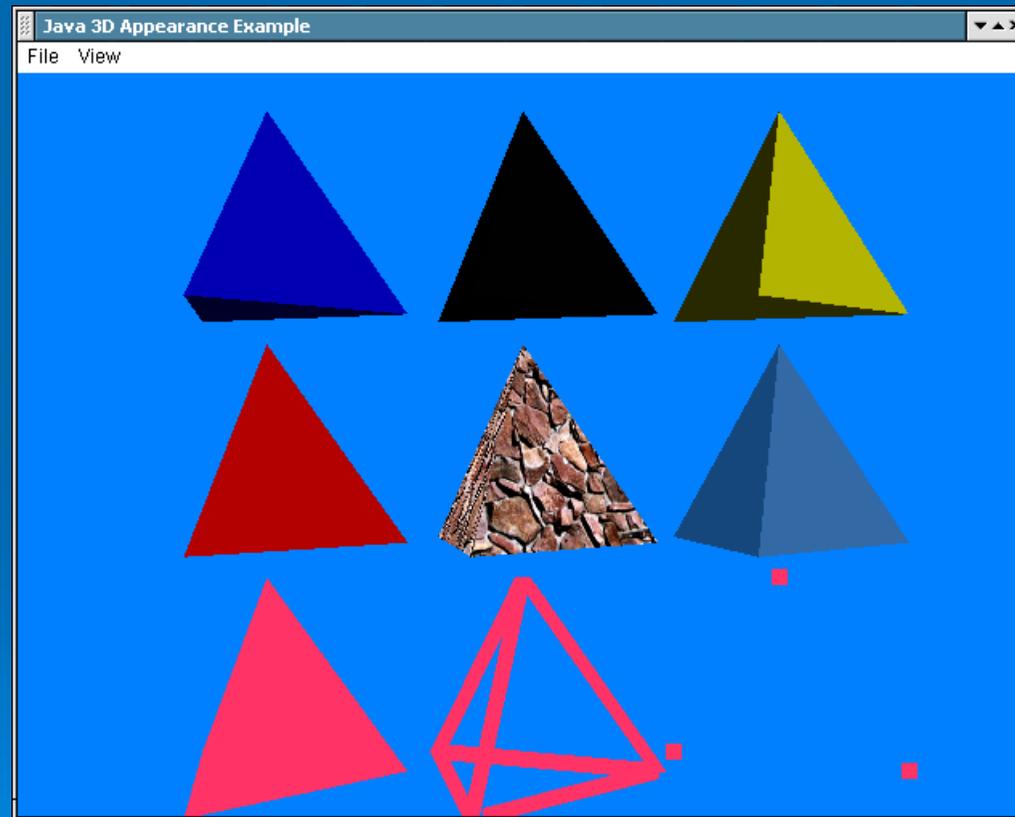


Las Apariencias es lo Que Importa

- Java 3D permite personalizar la apariencia de cada objeto en la escena, incluyendo su:
 - Color
 - Transparencia
 - Modelo de sombreado (Gouraud, phong, ...)
 - Grosor de las líneas
 - ...
- La apariencia de un objeto se manipula a través de la clase **Appearance**.

Las Apariencias es lo Que Importa

- Ejemplo con distintos tipos de apariencias



Resumen

- Java 3D es una extensión de Java que proporciona una interfaz de **alto nivel** para la creación de aplicaciones 3D.
- En un API **multiplataforma**.
- Permite incluir en la escena multitud de **formatos**.
- Se puede **integrar** con cualquier aplicación escrita en el lenguaje Java.
- Es de código abierto. ;)

Referencias

- Sitio de la comunidad de Java 3D
 - <http://www.j3d.org>
- Sitio oficial de Java 3D en Sun
 - <http://java.sun.com/products/java-media/3D/>
- Sitio del proyecto de código abierto de Java 3D
 - <https://java3d.dev.java.net/>
- Tutorial muy completo de Java 3D
 - <http://www.sdsc.edu/~nadeau/Courses/Siggraph99/>
- Tutorial de Java 3D en castellano
 - <http://www.programacion.com/java/tutorial/3d/>

Segunda parte

Proyecto Looking Glass



¿Qué es Looking Glass?

- Looking Glass (LG3D) es un proyecto de innovación desarrollado por Sun Microsystems basado en su tecnología Java, también Java 3D
- Aplica la tecnología 3D al clásico sistema de ventanas, las ventanas son representadas en un entorno 3D y manipuladas como objetos 3D
- Su objetivo no es sólo ofrecer un entorno más vistoso, sino una experiencia más rica tanto para el trabajo como para el entretenimiento

http://www.sun.com/software/looking_glass/index.html

¿Qué es Looking Glass?

➤ Romper barreras

- Las dos dimensiones del entorno de escritorio actual
- La manera en la que evolucionan esos entornos



Pioneros

- 1979, Xerox Star
 - “La mejor forma de predecir el futuro es inventarlo” (Alan Kay)

A screenshot of the Xerox Star 8010 graphical user interface. The desktop is filled with various windows and icons. On the left, there is a window titled "XEROX 6085 Workstation" with a document titled "User-Interface Design". In the center, there is a window titled "Activity under the old and the new" showing a bar chart. On the right, there is a window titled "DOS 3.3 Lotus data" showing a table of data. At the bottom, there is a taskbar with various icons. The desktop background is a grid of icons representing files and folders.

XEROX 6085 Workstation

User-Interface Design

To make it easy to compose text and graphics, to do electronic filing, printing, and mailing all at the same workstation, requires a revolutionary user interface design.

Bit-map display - Each of the pixels on the 19" screen is mapped to a bit in memory; thus, arbitrarily complex images can be displayed. The 6085 displays all fonts and graphics as they will be printed. In addition, familiar office objects such as documents, folders, file drawers and in-baskets are portrayed as recognizable images.

The mouse - A unique pointing device that allows the user to quickly select any text, graphic or office object on the display.

See and Point

All functions are visible to the user on the keyboard or on the screen. The user scrolling and retrieval by selecting them with the mouse and touching the MOVE, COPY, DELETE, PROPERTIES command keys. Text and graphics are edited with the same keys.

Shorter Production Times

Experience at Xerox with prototype work stations has shown shorter production times and thus lower costs, as a function of the percentage of use of the workstations. The following equation can be used to express this:

Year	Non 6085	6085
1978	95.2	15.8
1980	41.1	59.9
1982	45	55
1984	30	70
1986	10	90
1988	5	95

Text and Graphics

To replace typesetting, the 6085 offers a choice of type fonts and sizes, from 6 point to 36 point:

- 18-point text.
- 24-point text.
- 36-point text.

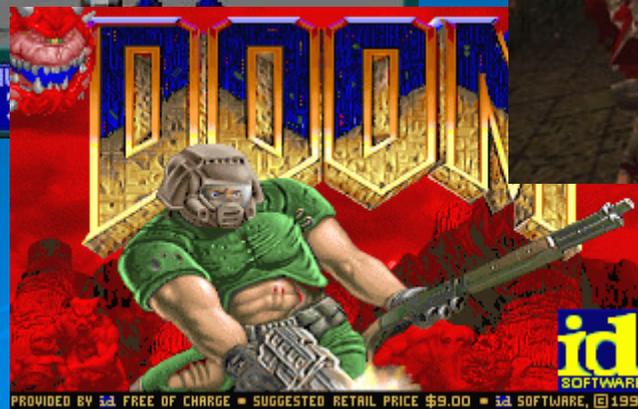
Pioneros

- 1985, NASA: Proyecto VIEW



Pioneros

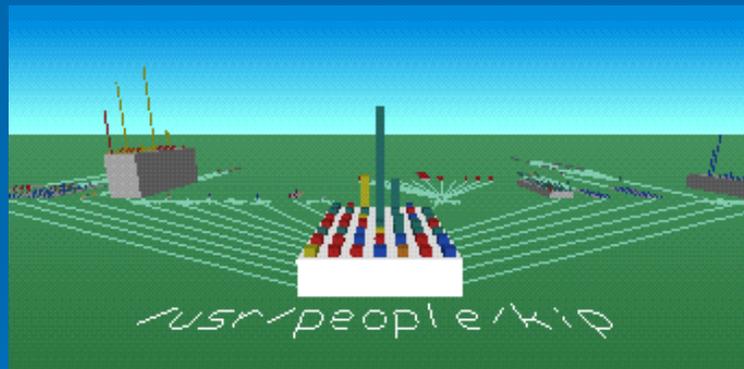
- 1992, Wolfenstein3D: Videojuegos 3D
 - Cada vez más exigentes, popularizan el hardware 3D
 - ¿Un desktop tan entretenido como jugar al Quake?



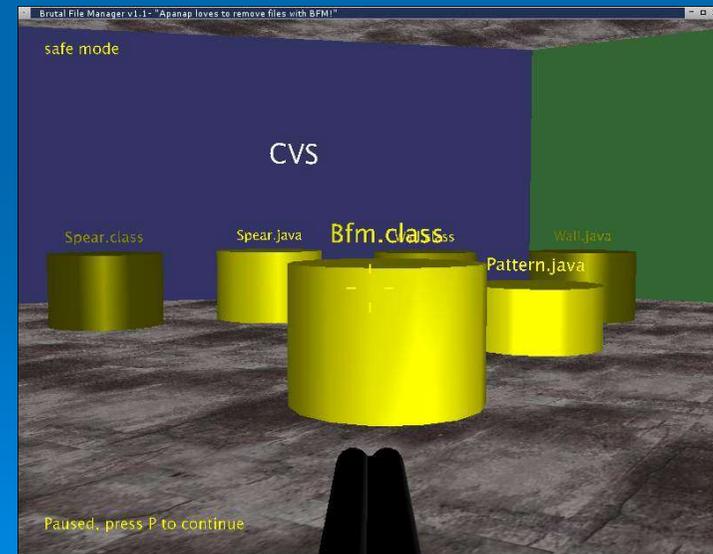
Pioneros

➤ Nuevas metáforas

- Sgi FSN (“fusion”), 1992
 - ¡Como en Parque Jurásico!
 - http://www.sgi.com/fun/freeware/3d_navigator.html

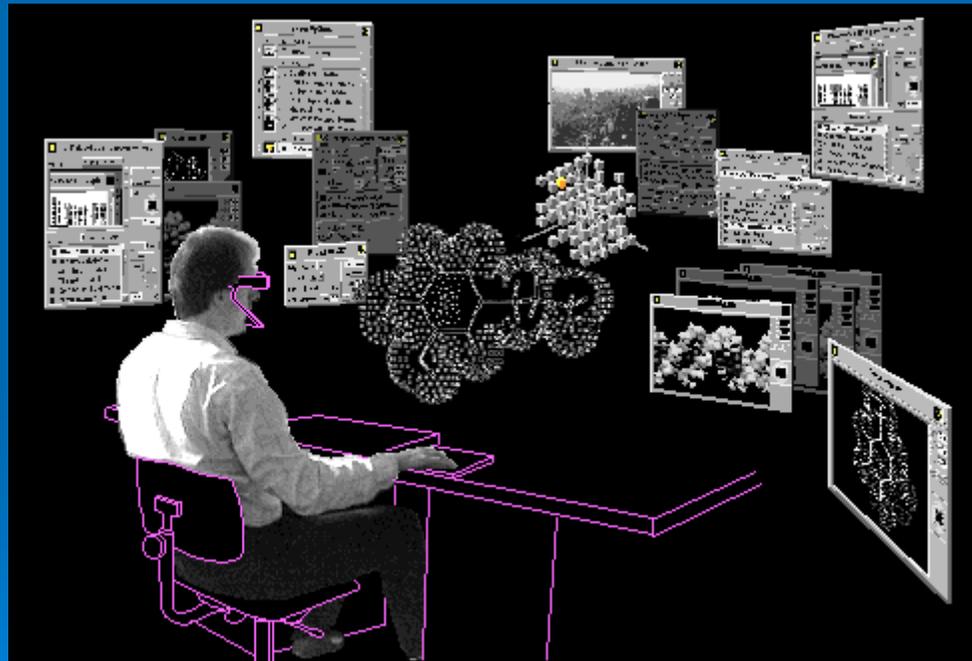


- The Brutal File Manager, 2003
 - <http://www.forchheimer.se/bfm/>



Pioneros

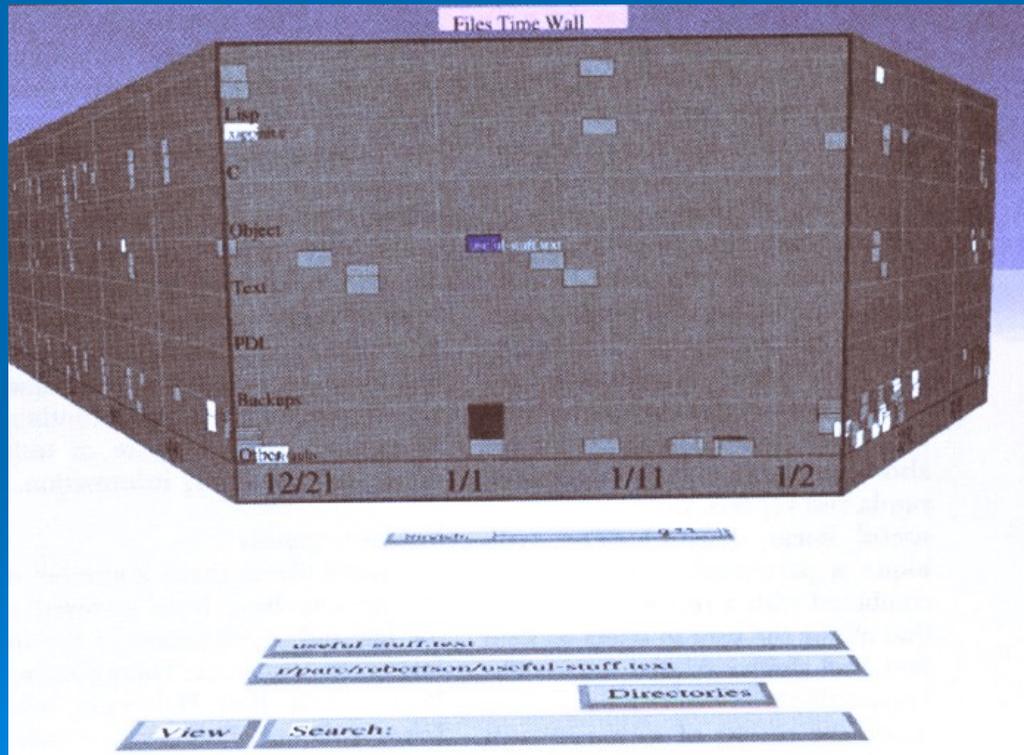
- 1993, James Leftwich: InfoSpace
 - Crear una relación 3D con la información



<http://www.well.com/www/jleft/orbit/infospace/>

Pioneros

- 1993, Xerox Information Visualizer
 - Superar los límites físicos de la pantalla



¿Revolución? en Internet

- La expansión de Internet abrió la puerta a nuevas posibilidades, entre ellas la de acceder a mundos virtuales multiusuario a través de la Web
- Dos tecnologías se convirtieron en la referencia para la creación de mundos virtuales y aplicaciones 3D para la Web:
 - VRML (1.0, 1995)
 - Java 3D (1.0, 1997)



¿Evolución? en el desktop

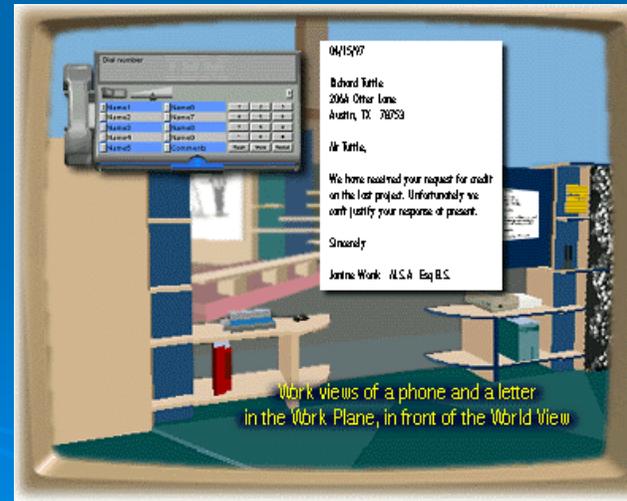
➤ Retos

- Los entornos de ventanas actuales están basados en bitmaps y primitivas 2D.
- Los gráficos 3D se llevan a cabo aparte, separados de esos gráficos 2D.
- Pero representar las ventanas en 3D requiere acceder a los entresijos de los sistemas operativos:
 - La salida gráfica de las aplicaciones debe poder utilizarse como textura en el entorno 3D
 - La entrada por teclado y ratón debe ser capturada e interpretada en el contexto del entorno 3D

¿Evolución? en el desktop

➤ Experimentos

- Microsoft Task Gallery [Robertson, 2000]
 - <http://research.microsoft.com/adapt/TaskGallery/>
- IBM RealPlaces [Roberts, 2000]
 - http://www-3.ibm.com/ibm/easy/eou_ext.nsf/publish/580



¿Evolución? en el desktop

➤ Alternativas

- Clockwise3D, 2000
 - <http://www.clockwise3d.com/>
- 3DNA Desktop, 2002
 - <http://www.3dna.net/>



¿Evolución? en el desktop

➤ Preguntas

- ¿Cómo es una ventana en 3D?
- ¿Qué hay en la cara posterior de una ventana?
- ¿Cómo son los elementos de una ventana en 3D?
- ¿Cómo serán las nuevas aplicaciones 3D?
- ¿Cómo se utiliza el ratón y el teclado en un entorno tridimensional?
- ...

Características de LG3D

➤ Ejecutar aplicaciones 2D



Características de LG3D

- Rotar ventanas, anotar texto detrás



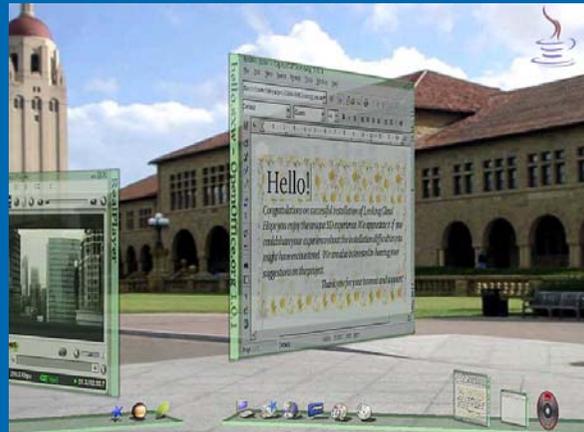
Características de LG3D

- Organizar las ventanas en 3D



Características de LG3D

➤ Panorama



Características de LG3D

➤ Nuevas aplicaciones 3D



Demo

The advertisement features a background image of Earth from space with a futuristic interface of vertical panels. The text is arranged in three main sections: a title, a call to action, and a URL. The Sun logo is positioned on the right side of the bottom section.

**Project Looking Glass:
A Revolutionary Evolution of Today's Desktop**

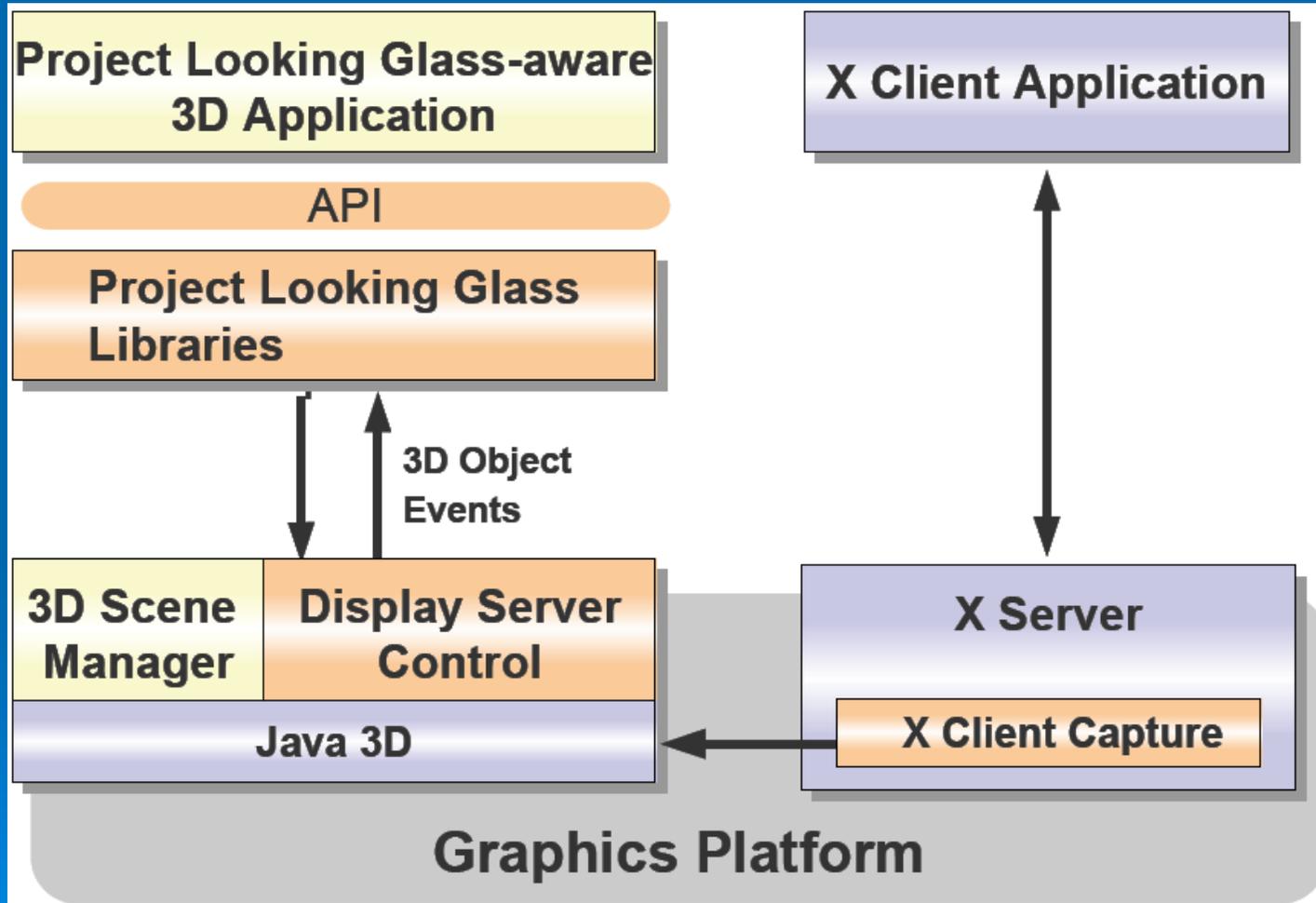
To receive updates or find out how to
get involved, join the mail list at:

www.sun.com/software/looking_glass

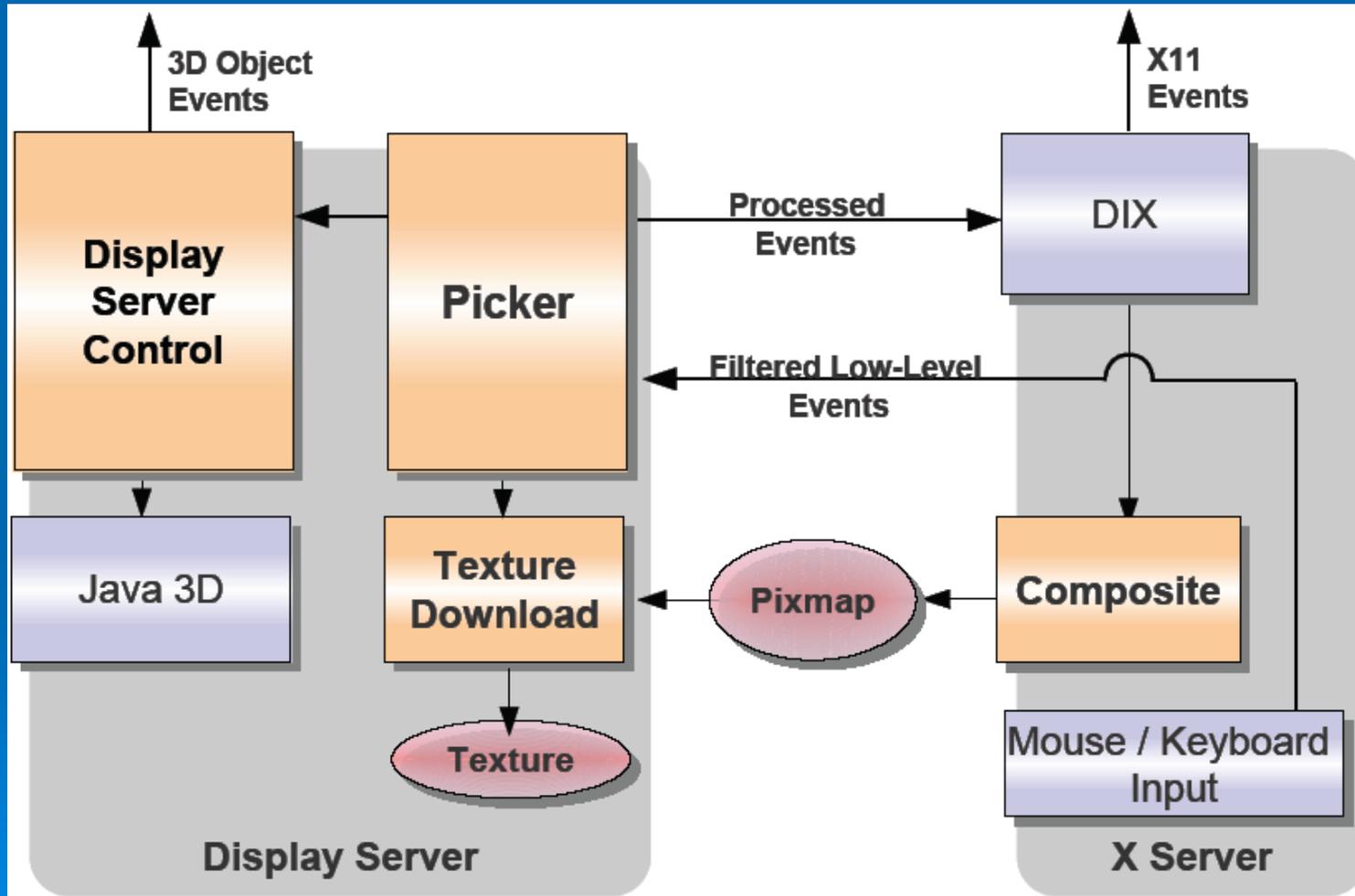


http://www.sun.com/software/looking_glass/demo.html

Arquitectura



Plataforma gráfica



Licencia Open Source

- Este prototipo de Looking Glass es sólo el principio, faltan por explorar muchas más ideas y posibilidades
- Por ejemplo, uno de los campos que se desea investigar es el “look & feel” de Swing 3D
- Sun ha apostado por el modelo “open-source”, ofreciendo el código del proyecto a la comunidad de software libre

<https://lg3d.dev.java.net/>

Requisitos

➤ Librerías de desarrollo

- JDK 5.0
- Java 3D 1.3.2
- JAI 1.2.2

➤ Hardware

- CPU: ~2GHz Pentium4 o similar
- Tarjeta gráfica 3D (GeForce2 32MB o similar)

<https://lg3d-core.dev.java.net/lq3d-getting-started.html>

¿Java Desktop 3D?

- Al igual que The Task Gallery o RealPlaces, Looking Glass es un proyecto experimental cuyos resultados podrían incorporarse en futuras versiones de los entornos de ventanas
- En el caso de Sun, ese entorno es Java Desktop System
- En el caso de Microsoft, el nombre clave de su futuro entorno es Longhorn

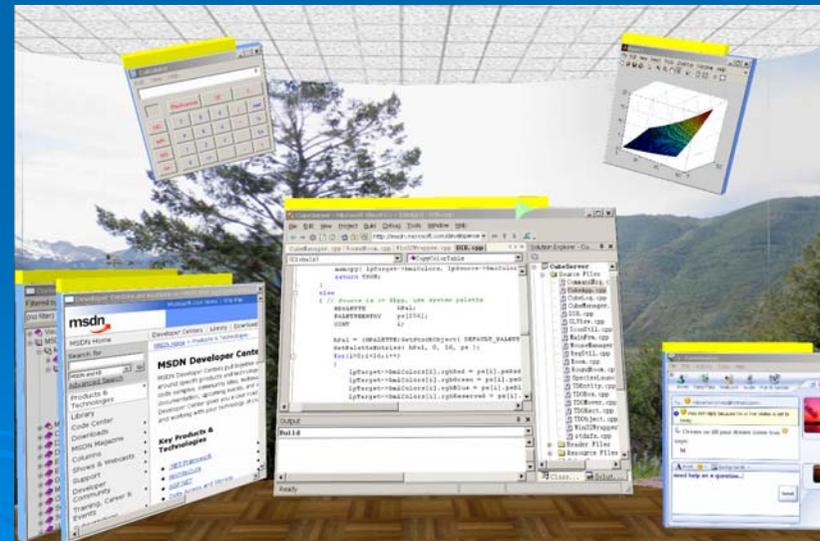
Otros desktops 3D

➤ SphereXP

- <http://www.hamar.sk/sphere/>

➤ Infinite-3D, Cube

- <http://www.infinite-3d.com/index.html>



Más allá del desktop 3D

- Algunos proyectos persiguen ir más allá de los entornos monousuario y crear espacios 3D multiusuario basados en software de código abierto
- Proyectos en curso:
 - Open Source Metaverse Project
 - <http://metaverse.sourceforge.net/>
 - Croquet
 - <http://croquetproject.org/>
 - MUPPETS
 - <http://muppets.rit.edu/>



Gracias

