

DEVELOPMENT OF VIRTUAL ENVIRONMENTS ON JAVA3D**DESARROLLO DE AMBIENTES VIRTUALES EN JAVA 3D**

**MSc. Luz Marina Santos Jaimes, Ing. Edwin Albeiro Flórez Miranda
Ing. Wilson Alonso Pérez Miranda**

Universidad de Pamplona,
Ciudadela universitaria, Pamplona, Colombia
{lsantos, eaflorez, wperez}@unipamplona.edu.co

Abstract: In works focused in the area of the virtual reality new forms of interaction with animation programs 3D and the knowledge that occurs to the interior of immersed virtual worlds in the cyberspace are analyzed. This work reflects about the characteristics of the virtual reality and the graphical library 3D like support to develop virtual environments, considering that with the programming of *applets* it is possible to interact with the virtual world through Internet.

Resumen: En trabajos enfocados en el área de la Realidad Virtual se analizan nuevas formas de interacción con programas de animación 3D y el conocimiento que se da al interior de los mundos virtuales inmersos en el ciberespacio. Este trabajo reflexiona acerca de las características de la realidad virtual y la librería gráfica Java 3D como soporte para desarrollar ambientes virtuales, considerando que con la programación de *applets* se puede interactuar con el mundo virtual a través de Internet.

Keywords: Virtual Reality, Java 3D, environments, models 3D, three-dimensional, real time

1. INTRODUCCIÓN

La presencia e influencia de la tecnología en todos los ámbitos de la vida cotidiana, ocurre de manera independiente del debate que aún se da respecto de sus bondades y defectos. La tecnología, sus productos, servicios, manifestaciones y mensajes sociales y culturales, cubiertos y encubiertos de que es portadora, forma parte o es un hecho corriente en la vida de casi todos nosotros.

Los ambientes virtuales son modelos tridimensionales que permiten aplicar aspectos importantes dentro de la realidad virtual como son: entornos interactivos, entornos inmersos, entornos tridimensionales y lo más importante, que todo ocurre en tiempo real; estos entornos se integran en uno solo que se denomina entorno virtual, distinguido por contener detalles como la forma, el color, la iluminación y otras características físicas de los objetos incluidos en el mundo sintético.

El computador utiliza estas informaciones para representar los objetos de la manera más realista posible y para modificar de forma dinámica las representaciones en función de las conductas de los usuarios, generando así la posibilidad de integrar y modificar los espacios visuales en tiempo real. La incorporación de la realidad virtual en entornos interactivos permite la aplicación más eficiente de los aspectos que la componen y hace que el usuario se sienta inmerso en ella.

2. REALIDAD VIRTUAL (INTEGRACION DEL SISTEMA)

La realidad virtual era una de las ramas informáticas que más sorprendían a los profanos y que más ilusionaban a los ingenieros y programadores que trabajaban en su desarrollo. Además de las ilimitadas aplicaciones en el sector del ocio, tanto en videojuegos, como en simulaciones lúdicas, esta tecnología, totalmente desarrollada, se aplica en simulaciones de prácticas peligrosas, como entrenamiento de profesionales de riesgo, bomberos, desactivación de explosivos, situaciones de emergencia, etc.

De la realidad virtual se pueden destacar las siguientes características.

2.1 Es un entorno tridimensional

En general, podemos decir que se trata de un entorno n-dimensional, si queremos considerar al tiempo como la cuarta dimensión y otras tantas posibilidades en el momento sólo abiertas para los matemáticos y científicos de "alto calibre". El hecho de ser tridimensional no sólo indica que el aspecto es aparentemente tridimensional, sino que de hecho la representación interna de los elementos que componen el Ambiente Virtual es tridimensional.

2.2 Es un entorno interactivo y autónomo

La idea de la realidad virtual es también proveer una forma en la que los usuarios puedan interactuar con el entorno, en cuanto estén en la capacidad de alterar el ambiente y/o ser alterados por él, al igual que los elementos del entorno pueden interactuar entre sí. Adicionalmente, estas alteraciones deben ser autónomas, es decir, que sea el usuario quien decida este tipo de interacción.

2.3 Es un entorno inmersivo

El objetivo final de la realidad virtual es sumergir al usuario en un ambiente simulado. Mucha gente tiende a malinterpretar esta característica y consideran sólo realidad virtual lo que tenga un casco o gafas u otros dispositivos externos más complejos. El uso de dispositivos periféricos en aplicaciones de realidad virtual, y su mayor o menor complejidad en realidad definen diferentes niveles de inmersión. La inmersión se refiere principalmente al hecho de reconocer que el usuario está dentro de un entorno tridimensional, y esto se consigue cuando existe un arriba, un abajo, un cerca, un lejos, y otra cantidad de aspectos espaciales y temporales.

2.4 Todo ocurre en tiempo real

Esta característica se refiere al hecho de que las entradas provistas por el usuario en una aplicación de realidad virtual, como es el movimiento, la alteración del ambiente, realizar acciones sobre un objeto, etc., se reflejan inmediatamente (o al menos lo más inmediato que puede conseguir un computador) en los dispositivos de salida al usuario.

2.5 Presencia vs. Realismo

Existen dos aspectos que toda aplicación de realidad debe considerar en mayor o menor grado. Uno de ellos es la presencia que se refiere al hecho de que el ambiente pueda representar las características necesarias para que un usuario se sienta dentro de él, como por ejemplo la gravedad, la existencia de colisiones, de clima, del paso del tiempo, etc. Aparte de las características mencionadas el mayor o menor grado de presencia se consigue principalmente, con un buen tiempo de respuesta, de modo que el usuario no perciba retardos entre las acciones que él realiza y el efecto inmediato que produce en el ambiente.

3. ASPECTOS DE DISEÑO EN DESARROLLO DE APLICACIONES DE REALIDAD VIRTUAL CON JAVA 3D

3.1 Caso de uso

El modelo de casos de uso es una parte que trata de mostrar los requisitos funcionales de un sistema, es decir, las condiciones y posibilidades que debe cumplir el sistema. En la figura 1 se muestra el modelo de casos de uso para el ambiente virtual. En este modelo básicamente hay dos actores "Usuario" y "Navegador Web", y el caso de uso "Cargar Modelo".

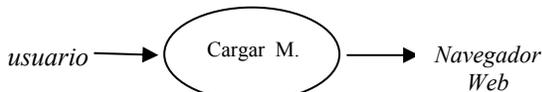


Fig. 1. Modelo de caso de uso para el ambiente virtual

3.2 Grafo de Escena (scene graph)

Una definición común de un grafo es que se trata de una estructura compuesta por nodos y arcos con una relación (n:m) entre nodos. Un nodo es un elemento, y un arco es la relación existente entre los elementos.

Un *scene graph* es una estructura de datos árbol en donde hay un nodo raíz y una relación de uno a muchos (1:n) entre sus elementos. Los nodos en un *scene graph* son instancias de las clases de Java 3D. Los arcos representan los dos tipos de relación posible entre ellas. La relación más común entre nodos es padre-hijo (representada por una flecha sólida), en donde solo puede haber un padre para cada nodo y cualquier número de hijos. El otro tipo de relación es por referencia (representada con una flecha punteada) que asocia un objeto del tipo nodo componente (*NodeComponent*) con un nodo (*node*) del *scene graph*. (Ver Figura 2 y 3)

Un universo virtual en Java 3D se crea desde un *scene graph*. El *scene graph* está formado por objetos *NodeComponent* que definen los atributos de geometría y apariencia utilizados para renderizar¹ un objeto visual (aquellos que se encuentran en el *scene graph*).

¹ *Render* es un término que proviene del mundo del diseño gráfico en 3D, que significa convertir un modelo 3D en una imagen de aspecto real, aplicando a cada superficie las adecuadas texturas, juegos de sombras y luces, etc.

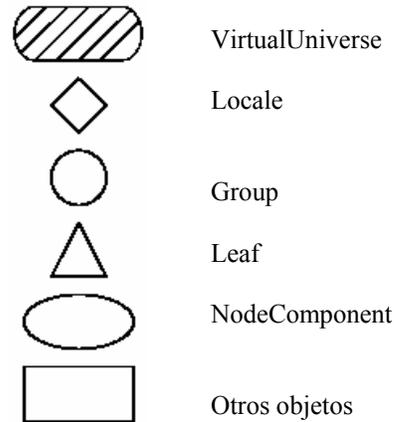


Fig. 2 Representación de símbolos del grafo de escena

Solamente hay un camino de la raíz del árbol hasta cada hoja. Cada camino especifica la información sobre localización, orientación y tamaño del objeto visual que está en la hoja. El renderizador de Java 3D aprovecha esto para representar los objetos en el orden que él considera más conveniente.

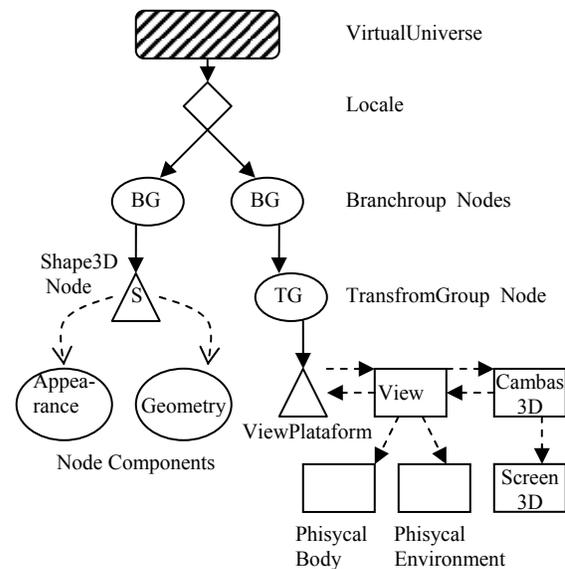


Figura 3. Grafo de Escena

El grafo de escena tiene solamente un *VirtualUniverse*, que a su vez tiene un objeto *Locale* que proporciona referencias a un punto en

el escenario virtual y sirve de raíz para varios sub-gráficos. Los objetos *BranchGroup* son la raíz de los sub-gráficos y tienen dos categorías:

- ✓ Por un lado está la rama de contenido gráfico que especifica el contenido del universo virtual (aparición, comportamiento, geometría, etc.)
- ✓ Por el otro está la rama de vista gráfica, que especifica los parámetros de visualización (posición y dirección).

4. PROGRAMACION JAVA 3D

4.1 El lenguaje Java

Java es un lenguaje de programación de alto nivel que cumple características como que es simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutral, multitarea y dinámico. Java es un lenguaje compilado e interpretado, normalmente los lenguajes tradicionales son o compilados o interpretados. La figura 4 muestra el proceso que sigue un programa en java.

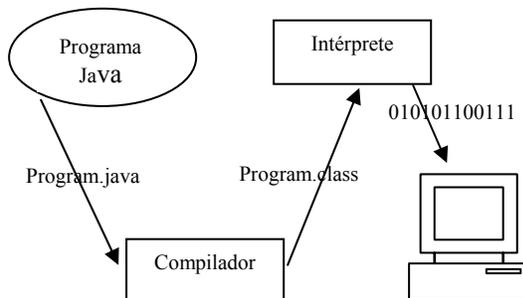


Fig.4. Funcionamiento de un programa en Java.

4.2 La plataforma Java.

Una plataforma es el entorno de hardware y software en que puede ser ejecutado un programa informático. La plataforma Java se diferencia de la mayoría de las demás plataformas en que se trata de una plataforma software únicamente, que correrá encima de otras plataformas basadas en hardware. La mayoría de las demás plataformas, sin embargo, se describen como una combinación de hardware y un sistema operativo. La plataforma Java está formada por dos componentes:

- La Máquina Virtual Java (*Java Virtual Machine* o Java VM).
- La Interface de Programación de Aplicaciones Java (*Java Application Programming Interface* o Java API).

4.3 Características generales de Java.

Las características principales que nos ofrece Java respecto a cualquier otro lenguaje de programación y más concretamente respecto a C/C++, son las siguientes:

Simple. Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características más confusas de algunos de éstos lenguajes. C/C++ no es un lenguaje conveniente sobre todo por razones de seguridad, pero C y C++ son los lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.

Orientado a objetos. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria.

Distribuido. Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.

Robusto. Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las

posibilidades de error, además maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.

Seguro. En el caso de los *applets* (programas que se ejecutan desde el navegador web) se pueden poner políticas de seguridad para que no se pueda escribir en el disco del cliente o enviar información confidencial del cliente al servidor.

Portable. Más allá de lo dicho sobre la portabilidad cabe destacar que Java proporciona una librería de ventanas portable de modo que la aplicación se verá igual en cada una de las plataformas.

Interpretado. Uno de los puntos débiles de java es la velocidad de interpretación. Aunque la velocidad de interpretación mejora con el paso de las versiones de la máquina virtual sigue siendo más lento que la compilación tradicional del lenguaje C++ por ejemplo, esperemos que se mejore la velocidad ya que Java no es todavía un lenguaje maduro, también existen compiladores tradicionales de Java como por ejemplo el gcj de GNU que aún está algo verde pero promete llevar la velocidad tradicional a Java, también existen compiladores de *bytecode* y en las últimas versiones de Sun se incluye la tecnología *Hot Spot*, en la que se realizan compilaciones dinámicas optimizando el código en tiempo de ejecución.

Multihilo. Incorpora la ejecución de hilos en el propio lenguaje a diferencia de C++, que se necesitaría, o bien utilizar una librería para gestionarlos, o bien llamadas a bajo nivel (no recomendable para un lenguaje de alto nivel y orientado a objetos). Un hilo no es más que un proceso independiente con lo cual se tiene la oportunidad de lanzar tareas que se realicen simultáneamente y luego recoger el resultado de las distintas tareas y componerlo. Con esto se gana velocidad aprovechando la multitarea de la plataforma en la que se ejecuta.

4.4 Java3D.

La librería gráfica Java 3D, también conocida como API Java 3D, es una librería del API Java encargada de proporcionar una colección de componentes dedicados a la realización de aplicaciones tridimensionales.

Se trata de una interface para la escritura de programas que nos permitirán visualizar e interactuar con gráficos tridimensionales. Java 3D es una extensión estandarizada del entorno de desarrollo de Java (*Java Development Kit, JDK*) que es un conjunto de herramientas dedicadas a compilar y ejecutar aplicaciones Java. La librería API Java 3D ofrece una colección de entidades de alto nivel para la creación y manipulación de geometrías tridimensionales así como estructuras para la renderización de tales geometrías. Java 3D contiene funciones para la creación programas de aplicación de imágenes, visualizaciones, animaciones, e interacciones con gráficos 3D.

La librería gráfica API Java 3D es una jerarquía de clases de Java que sirve como interface para la renderización de sofisticados gráficos tridimensionales así como renderizaciones sonoras.

Trabaja a alto nivel utilizando constructores que permiten la creación y manipulación de objetos de geometría 3D. Estas geometrías se encuentran incluidas en un universo virtual que es el que finalmente será renderizado. Este API está creado con una flexibilidad que permite crear universos virtuales de variados tamaños, desde astronómicos hasta subatómicos.

A pesar de toda esta funcionalidad Java 3D permite sencillez a la hora de su uso, puesto que los detalles de renderización son manejados automáticamente. Un programa Java 3D crea instancias de objetos Java 3D y los sitúa dentro de una estructura de datos llamado *scene graph*.

El *scene graph* es una descripción en forma de grafo de los objetos Java 3D capaz de especificar completamente el contenido del universo virtual y el como debe ser renderizado dicho universo.

Los programas Java 3D son capaces de ejecutarse como aplicaciones aisladas, como *applets* para clientes de Internet o de ambas formas.

Paquetes del API Java 3D: Hay cientos de campos y métodos en las clases del API Java 3D, sin embargo, un simple universo virtual que incluya animaciones puede ser construido con un solo unas pocas clases.

El API define alrededor de 100 clases presentadas en el paquete base *javax.media.j3d*, otros paquetes son usados para escribir programas 3D como es el caso del paquete *com.sun.j3d.utils* referido como clases de utilidades java 3D. El paquete base incluye solo las clases de más bajo nivel para programación en java 3D, mayores funcionalidades son proporcionadas por el paquete de utilidades. Las clases de utilidad recaen en cuatro categorías: Cargadores de contenido (objetos visuales en un grafo de escena, p.e., cubos, etc.), ayudas de construcción del grafo de escena, clases de geometría y utilidades.

También cada programa java 3D usa clases del paquete *java.awt* y *javax.vecmath*. Las clases de AWT crean una ventana para mostrar la renderización y el paquete *javax.vecmath* define clases matemáticas para puntos, vectores, matrices, entre otras.

5. CONCLUSIONES

Con la aplicación de la realidad virtual se logra demostrar los alcances de las herramientas proporcionadas por Java 3D para la creación y animación de mundos virtuales. A pesar de las dificultades relacionadas con la restringida información sobre manejo de este tipo de herramientas y sus aplicaciones.

Java 3D es una herramienta completa y proporciona ventajas significativas para el desarrollo de aplicaciones en el área de la realidad virtual.

Los ambientes virtuales permiten la integración directa entre usuario y la maquina a partir del manejo eficiente de los aspectos que componen la realidad virtual en todas las áreas de la ingeniería.

La programación de ambientes virtuales en Java 3D requiere conocimientos previos en programación orientada a objetos o por lo menos haber programado en C++.

REFERENCIAS

- What's Real about Virtual Reality? F.P. Brooks Jr, 1999. [Artículo \(formato PDF\)](#)
- What is Virtual Reality? J. Isdale, 1998 [Artículo \(formato HTML\)](#)
- Virtual Enviroments. F. Balaguer, A. Manguili [Artículo \(formato ZIP\)](#)
- Club VR, *Cronologia Realidad Virtual*. RV-UNAM. World Wide Web URL: <http://exodus.dcaa.unam.mx/virtual/history1.html>
- D.S. Perry, Lynellen, Smith, Christopher M., y Young, Steven, *Una investigación sobre interfaces actuales de realidad virtual*. World Wide Web URL: <http://www.acm.org/crossroads/espanol/xrds3-3/vrhci.html>
- Sitio Oficial Java 3D <http://java.sun.com/products/java-media/3D/>
- Introduction to Programming with Java3D (SIGGRAPH'99) by H. Sowizral & D. Nadeau [Curso introductorio en HTML](#)