

# INTEGRACIÓN DE HERRAMIENTAS EN LAS PRÁCTICAS DE INFORMÁTICA GRÁFICA

**D. José Luis Vicén Cruz<sup>1</sup>, D. Eduardo Falces Larraga<sup>1</sup>,  
D<sup>a</sup>. Marta Arguedas Lafuente<sup>1</sup>, D. Rafael Embid Romero<sup>1</sup>**

*<sup>1</sup>E.U.P.L.A. (Escuela Universitaria Politécnica La Almunia de D<sup>a</sup> Godina)  
e-mail: joselo@eupla.unizar.es*

**Resumen:** La Informática Gráfica ha alcanzado en la actualidad un alto grado de desarrollo y complejidad, apareciendo continuamente nuevas y poderosas herramientas que permiten la creación y diseño de gráficos y escenas en 3D cada vez con mayor calidad y menor esfuerzo para el programador/diseñador. Esto, unido a la irrupción de Internet y a la potencia de los nuevos ordenadores y tarjetas de vídeo, obliga a una mayor especialización y a un mayor conocimiento de las diferentes opciones que tanto el hardware como el software ponen en manos del programador. En esta ponencia comentamos cómo en las prácticas de la asignatura de Informática Gráfica se hace un recorrido por diferentes herramientas software, como OpenGL y VRML, y diversos entornos, como Windows95/NT y navegadores como NetScape ó Explorer con extensiones VRML, o programas visores de mundos virtuales, como Viscape.

## 1.- INTRODUCCIÓN

La velocidad de desarrollo de nuevo hardware actualmente es muy alta. Procesadores de más de 400 MHz, tarjetas de vídeo con procesadores ultrarrápidos y aceleradores 3D, de 4 a 16 Mb de memoria de vídeo, nuevos buses, etc. Realmente la potencia de los ordenadores actuales era difícil de imaginar hace tan sólo unos años. Sin duda, uno de los campos que más se

ha beneficiado de este desarrollo es el de los gráficos por computadora. Imágenes fotorealistas que anteriormente únicamente se podían obtener con estaciones de trabajo gráficas (Silicon Graphics, por ejemplo) hoy día se pueden obtener con PC's domésticos. Por ello ya no es necesario realizar millonarias inversiones para iniciarse en el campo de los gráficos y animaciones por ordenador, y nuestros alumnos pueden constatar como ya no solamente se tienen que conformar con representar objetos de 3D en malla de alambre, sino que son capaces de utilizar las más modernas herramientas con las que crear mundos tridimensionales, con objetos que poseen características propias y comportamientos específicos, y que son capaces de exportar esos mundos a todo el mundo vía Internet. Las prácticas dirigidas en esta asignatura aseguran que los alumnos utilicen en cada momento la herramienta más adecuada (lenguaje C++, ensamblador, lenguajes visuales con OpenGL, y VRML con pequeños scripts en JAVA, etc.)

## **2.- ORGANIZACIÓN DE LA ASIGNATURA**

La asignatura de Informática Gráfica se ofrece como asignatura optativa en el tercer curso de Ingeniería Técnica en Informática de Sistemas, y se imparte durante tres horas semanales, de las cuales aproximadamente la mitad corresponde a prácticas y ejercicios, y la otra mitad corresponde al contenido teórico propio de la asignatura. Por todo ello, la asignatura posee una carga lectiva de 9 créditos, repartidos a partes iguales entre la teoría y la práctica.

El programa teórico desarrollado en esta asignatura incluye los siguientes apartados:

- **Hardware y software gráfico:**  
Tecnologías de impresión, tecnologías de pantallas, sistemas de representación por barrido, controladores de vídeo y dispositivos de entrada para la interacción con el operador.
- **Geometría y operaciones de visión en 2D:**  
Coordenadas homogéneas, transformaciones bidimensionales, composición de transformaciones, sistemas de coordenadas, cambio de sistema, transformación del área de visión y recorte de líneas.

- Geometría y operaciones de visión en 3D:

Representación de la geometría tridi-mensional, transformaciones tridimensionales, proyecciones ortográficas y perspectivas, puntos de fuga, volúmenes de visión y recorte de líneas en tres dimensiones.

- Curvas y superficies:

Descripción geométrica de las curvas, implicación de ecuaciones paramétricas, técnicas de interpolación y aproximación (Bezier, B-Spline), curvas racionales y superficies de forma libre.

- Realismo visual y modelado sólido:

Eliminación de líneas/superficies ocul-tas, técnicas de visibilidad, sombreado, modelos de color, rendering, representación sólida, bases de la teoría de modelado sólido, representación facetada, espacial, por contornos, operaciones booleanas, cálculos de intersecciones, etc.

En cuanto a la parte práctica de la asignatura, se ha optado por una serie de programas bajo MsDos y Windows'95. Como primer lenguaje de programación utilizamos C++ y C++ Builder, aunque en las prácticas bajo Windows'95 se admite también como herramienta el Borland Delphi.

Las prácticas bajo MsDos buscan que el alumno aprenda a utilizar el hardware gráfico a bajo nivel, utilizando C++ como lenguaje base con pequeños fragmentos en ensamblador. En este punto se trabaja directamente con interrupciones y con la memoria de vídeo, tanto a baja como a alta resolución. Se hace un especial incapié en los modos VESA (800x600 y 1024x768, principalmente), para los que los alumnos gestionan los diferentes bancos de memoria y construyen una librería básica de instrucciones tanto en 2D como en 3D.

Aquí los alumnos ya son capaces de diseñar un interface gráfico apropiado para una aplicación de representación tridimensional, con manejo de ratón incluido, que es la práctica obligatoria exigida en este punto.

Para terminar la programación bajo MsDos se realiza una introducción a algunos formatos gráficos estándar (BMP, PCX y GIF) con ejemplos de lectura y representación de los mismos, y se hace una breve introducción a las demos gráficas, con la implementación de alguno de los efectos más comunes.

Para las prácticas bajo Windows95 se cambia radicalmente el enfoque. Ahora lo que se busca es poder utilizar herramientas existentes en el mercado que nos permitan programar escenas tridimensionales aprovechando las características del sistema operativo Windows. Estas prácticas se dividen en dos bloques claramente diferenciados: por una parte, se realiza una práctica en OpenGL, utilizando un lenguaje de programación visual, como Visual C++, Builder o Delphi, con las correspondientes librerías de OpenGL, y por otra parte se realiza una práctica bajo VRML, utilizando algún navegador estándar y algunas pequeñas rutinas en JAVA.

### **3.- PRÁCTICAS EN OpenGL**

OpenGL es una librería diseñada para crear imágenes y animaciones a partir de código, fácilmente portable a otros sistemas y plataformas hardware, y optimizada para una muy rápida ejecución.

Incorpora todos los conceptos vistos en la parte teórica de la asignatura, además de otros avanzados que destacamos a continuación:

- Mapeado de texturas:  
Capacidad de aplicar una imagen bitmap a una superficie.
- Z-Buffer:  
Capacidad para calcular distancias desde la situación del observador. Permite eliminar automáticamente superficies ocultas.
- Doble Buffer:  
Soporte de pantalla virtual para una animación fluida y de calidad.
- Efectos de iluminación y sombreado:  
Cálculo del ángulo de incidencia de una fuente de luz sobre las superficies de los objetos. Esto determina el sombreado de los mismos, que además ofrece la posibilidad de escoger el tipo de sombreado que queremos aplicar (facetado, Gouraud, Phong, etc).
- Propiedades de los materiales:  
Capacidad para modificar el brillo o reflectividad de los objetos, pudiendo dar efectos de metalizado, transparencia, pulimiento, etc.
- Canales alfa:

Esta es una característica importante, que consiste principalmente en poder dar la propiedad de invisibilidad a un color determinado, por lo que dicho color no interferiría con los colores que se encuentran detrás de él.

- **Matrices de transformación:**

Capacidad de poder cambiar la localización, tamaño y perspectiva de un objeto en 2D o 3D en el espacio.

OpenGL es un interface diseñado, mantenido y mejorado por la “OpenGL Architecture Review Board (ARB)”, que engloba a importantes compañías de software y hardware, como son Digital Equipment Corporation, Evans and Sutherland, IBM, Intel, Intergraph, Microsoft y Silicon Graphics. Además hay en el mercado una gran variedad de tarjetas de vídeo que interpretan directamente las instrucciones de OpenGL, lo que permite aumentar la rapidez de ejecución entre veinte y cincuenta veces, que para una imagen de menos de 100.000 polígonos crea una ilusión real de movimiento.

#### **4.- PRÁCTICAS EN VRML**

VRML es un lenguaje creado para la representación de mundos virtuales a través de Internet, si bien sus posibilidades no se restringen exclusivamente a la Red, sino que posibilita la creación de mundos en 3D que pueden interactuar con el usuario.

El lenguaje es una especie de script con una serie de primitivas que permiten representar objetos sencillos que se pueden combinar para formar otros objetos más complejos.

Como principales características de VRML destacamos las siguientes:

- **Primitivas de transformación:**

Implementa tanto la rotación como la traslación y el escalado.

- **Tratamiento preciso del color:**

Incorpora una serie de instrucciones para el manejo del color, y además permite establecer matices de transparencia y reflexión, que aporta un mayor realismo a los objetos.

- **Completo manejo de la luz:**

Implementa una gran cantidad de tipos de iluminación, como luces direccionales, focales y puntuales, que además da la posibilidad de controlar el color, intensidad, anchura del cono de iluminación, etc.

- **Manejo de texturas:**

Incorpora un completo conjunto de instrucciones para texturizar objetos, pudiéndose controlar aspectos como el ratio, y al cual se puede ajustar con instrucciones de transformación sobre la misma textura.

- **Inserción de código en JAVA:**

Una característica muy importante del VRML es su capacidad para dotar a los objetos de las escenas virtuales de movimiento o comportamiento propio, de forma que puedan interactuar entre ellos o con el usuario. Estos comportamientos se programan en JAVA, mediante pequeñas rutinas que se asignan a los diferentes objetos. Estos movimientos se pueden ejecutar de forma automática (para objetos con movimientos fijos) o en respuesta a algún evento (como colisiones, pulsación del ratón sobre ellos, etc).

Las prácticas se dividen en dos bloques: en el primero el alumno aprende a conocer y programar con este lenguaje, realizando una serie de ejercicios sencillos basados en la representación de primitivas. Las representaciones de los objetos son estáticas y no interactúan con el usuario ni entre ellos. Todavía no se incluye código en JAVA, limitándose al alumno a observar el resultado de su trabajo a través del navegador.

La segunda parte de las prácticas de VRML son algo más avanzadas, ya que se programan mundos virtuales más complejos, utilizando objetos más elaborados y añadiendo un cierto dinamismo a las escenas, dinamismo que se consigue programando el comportamiento de los diferentes objetos con pequeñas rutinas en JAVA, lenguaje éste que los alumnos ya conocen en grado suficiente.

El objetivo de esta segunda parte es la realización de mundos que respondan a una serie de eventos externos, dirigiendo al alumno para que

construya escenas que sean capaces de ser recorridas a través del navegador, con objetos que posean movimientos fijos e inalterables, mezclados con otros de movimientos guiados por el usuario.

El tipo de escena a desarrollar se deja a la imaginación del alumno (especie de ciudad tridimensional, campos de golf, mundos fantásticos, etc.)

En todos ellos tendremos que ser capaces de interactuar con el entorno (no poder atravesar paredes, poner en marcha diversos objetos, etc).

## **5.- CONCLUSIONES**

Con este programa de prácticas hemos conseguido que a lo largo del curso el alumno vaya utilizando un amplio abanico de herramientas, adecuadas en cada caso para el resultado esperado, que abarca diferentes formas de enfocar la programación de gráficos.

Durante el curso el alumno realiza programas en C++, ensamblador, programación visual bajo Windows utilizando OpenGL y desarrolla mundos tridimensionales bajo VRML y JAVA.

OpenGL y VRML son unos buenos aliados a la hora de practicar con los conceptos más avanzados de la Informática Gráfica, ya que reduce notablemente el tiempo de programación, lo que permite que los ejercicios prácticos sean más complejos y realistas, hecho que motiva al alumno y fomenta su creatividad.

El resultado de esta experiencia nos ha indicado que un porcentaje muy elevado de alumnos realizan los ejercicios con un nivel de perfección por encima del mínimo exigido, investigando ellos mismos algunas de las posibilidades que ofrece VRML, y que por problemas de tiempo no se explican en clase.

La tendencia a seguir en los próximos años será la de ir aumentando progresivamente el espacio dedicado a la programación en Windows'95 con OpenGL y VRML, (que podría ser complementada con librerías DirectX), e incluso ampliar VRML con aplicaciones para Internet.

## 6.- BIBLIOGRAFÍA

- [1] Fostner, Ron. "*Programming with OpenGL*". Dr.Bobb's Journal (julio 1995)
- [2] Neider, Jackie, Tom Davis, and Mason Woo. "*OpenGL programming Guide: The official Guide to Learning OpenGL, Release 1.*" Addison-Wesley, 1993 (conocido como "Red Book").
- [3] OpenGL Architecture Review Board. "*OpenGL Reference Manual: The Official Reference for OpenGL*" Addison-Wesley, 1992. (conocido como "Blue Book").
- [4] Prosser, Jeff. "*Advanced 3-D Graphics for Windows NT 3.5: Introducing the OpenGL Interface, part I*" Microsoft Systems Journal (Octubre 1994).
- [5] Crain, Dennis. "*Windows NT OpenGL: Getting Started*" (Abril 1994) Microsoft Developer Network CD. (Este es el primer artículo sobre OpenGL bajo Windows).
- [6] Jed Hartman, Josie Wernecke, "The VRML 2.0 Handbook" Addison-Wesley, 1996.