

CUADERNOS

# Tecnológicos

Suplemento de Macworld España, septiembre 1995. Número 40

Macworld



# QuickDraw 3D

6

# Introducción

Una de las tecnologías que siempre han formado parte de sistema operativo de los Macintosh ha sido QuickDraw, encargada de gestionar en el Mac todos los aspectos relativos a la representación gráfica en la pantalla del monitor de la información correspondiente tanto a tipografías como a los elementos que conforman las aplicaciones (entre otros aspecto).

Tras la aparición de QuickDraw GX, que formando parte de la versión del Sistema Pro se encargaba de mejorar sobre todo los aspectos relativos al manejo de las tipografías ofreciendo al usuario un mayor rango de posibilidades, la versión de QuickDraw 3D aporta un sinfin de nuevas capacidades orientadas, en todos sus aspectos, al trabajo desde aplicaciones que deban gestionar información para la generación de imágenes tridimensionales.

Como ya viene siendo habitual en las tecnologías más recientes de Apple, la compañía de la manzana no sólo se ha centrado en ofrecer mayor potencia y flexibilidad en la implementación de esta tecnología en entornos Macintosh. Al igual que sucede con OpenDoc, QuickTime, u otras tecnologías, la naciente QuickDraw 3D también tiene un punto de apoyo muy importante en el aspecto de la compatibilidad con otras plataformas (en este caso basadas en Unix o Windows). De esta forma, los usuarios que trabajen con QuickDraw 3D, podrán utilizar sus proyectos sin mayor problema en otras plataformas; gracias, sobre todo, a la compatibilidad del estándar utilizado para el archivo encargado de almacenar la información con la que se está trabajando.

## Cuadernos Tecnológicos Macworld

es un suplemento de la revista Macworld España  
con la colaboración de Apple Computer España

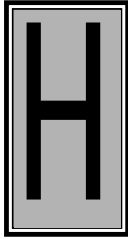
Número 6, QuickDraw 3D

**Autor: Javier González**

**Coordinador: Francisco Javier Rodríguez**

# QuickDraw 3D

## Las tres dimensiones de Apple



Hay ya mucho que Apple, con sus equipos Macintosh, representa la mejor elección para los profesionales del diseño gráfico, por su facilidad de uso, consistencia de interfaz de usuario y su cómodo y revolucionario entorno gráfico, a las que deben sumarse las cada vez más frecuentes tecnologías multimedia que complementan y amplían cada vez en mayor medida las posibilidades del Mac.

A lo largo de los años Apple ha ido introduciendo nuevas tecnologías, como los tipos TrueType, QuickDraw y QuickDraw GX. Tecnologías que han puesto al alcance de la mano del usuario un control y flexibilidad sin precedentes sobre campos como la impresión, el tratamiento de los gráficos en dos dimensiones, tratamiento de fotografía por ordenador y, ya más recientemente, el vídeo digital (tanto a nivel profesional como a nivel de producciones multimedia).

En este cuaderno encontrará otra de las recientes tecnologías de Apple, que si bien aún está en fase de desarrollo por parte de los ingenieros de Apple, ya está lo suficientemente avanzada como para ofrecer una información muy precisa sobre sus posibilidades. En concreto se trata de la tecnología QuickDraw 3D, con la que extiende estas innovaciones al complejo mundo de los gráficos en tres dimensiones; mejorando mediante su instalación como parte integrante de Sistema, el manejo de objetos desde aplicaciones para la generación de imágenes sintéticas (desde las cuales se obtiene un mayor potencial), así como otras partes del propio Sistema.

Por primera vez se integra en un Sistema Operativo de ordenador personal la tecnología de gráficos en tres dimensiones, aprovechando todas las ventajas de la plataforma RISC me-

dante el procesador PowerPC que actualmente está equipando alguna de las plataformas de gama alta y media de Apple, y que en un futuro próximo pasará a ser el corazón de todos los modelos Macintosh. También es el primero que, de la mano de Apple, se ha introducido en el mundo del ordenador personal.

La introducción de QuickDraw 3D, y el conjunto de aplicaciones que sacarán provecho de él, permitirá al usuario crear, manipular e incorporar gráficos en tres dimensiones en sus documentos (sean éstos del tipo que sean) con la misma facilidad y naturalidad con la que ahora lo hacen con gráficos en dos dimensiones, y de igual forma que se puede incorporar en la mayoría de los documentos otras tecnologías como es el vídeo digital. Además, las actuales aplicaciones 3D podrán ser modificadas para soportar QuickDraw 3D y sus normas de interfaz de usuario asociadas, que facilitará una aproximación más estándar a los gráficos 3D. De hecho, en el momento de escribir este Cuaderno Tecnológico, ya se ha hecho el anuncio de algunas aplicaciones (de síntesis de imagen) que soportan completamente la reciente tecnología de Apple.

### Un breve vistazo a QuickDraw

QuickDraw 3D elimina muchos de los tradicionales problemas de los gráficos en tres dimensiones, incorporando un nuevo y consistente formato, una completa interfaz de usuario estándar, un API (interfaz de programación) multiplataforma, y un conjunto de normas de uso y desarrollo que vienen a asegurar su consistencia y posibilidades de ser el próximo estándar del mercado.

3DMF (3D Metafile Format). Este es el nuevo formato de archivo multiplataforma, que



**QuickDraw 3D**  
permitirá a los  
usuarios crear,  
manipular  
e incorporar  
gráficos  
en tres  
dimensiones  
en sus  
documentos

soporta la información de objetos y trabajos realizados en un espacio tridimensional, no simple geometría, pero con toda la información que representa una vista (entendida desde la cámara, o punto de vista del usuario) o una escena.

El formato 3DMF pone al alcance del usuario la posibilidad de compartir imágenes entre aplicaciones con la misma facilidad con la que ahora lo hace con imágenes en otros formatos, como puede ser el caso de los archivos en formato PICT.

El API de programación, QuickDraw 3D facilita al desarrollador un conjunto de estándares para realizar operaciones con gráficos tridimensionales. Mediante este conjunto de APIs, los desarrolladores ya no tendrán que reinventar complicados algoritmos para el tratamiento de las siempre complejas funciones 3D; sobre todo en el caso de las funciones más comunes. Este estándar incluye una geometría de espacio tridimensional, una arquitectura para el tratamiento de sombras (shading) y generación de imágenes sintéticas (rendering), así como gestores específicos para dispositivos de aceleración en el tratamiento de este tipo de información.

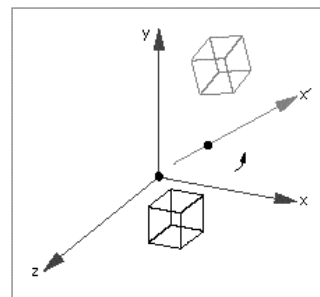
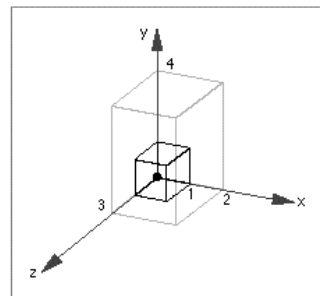
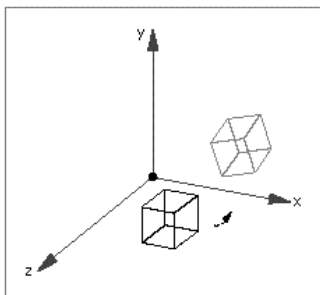
La interfaz de usuario y sus "normas" de utilización e implementación son la clave para que la tecnología QuickDraw 3D represente el primer conjunto realmente estándar para el tratamiento y gestión de gráficos en tres dimensiones. Con este estándar el usuario podrá aplicar sus conocimientos previos en el manejo de gráficos bidimensionales a los gráficos tridimensionales, y también extender esta familiaridad en el trabajo con gráficos 3D a las diferentes aplicaciones que trabajen con este tipo de gráficos.

### Ampliabilidad y plug and play

Tal vez la facilidad de ampliar la arquitectura, y la sencillez de hacerlo mediante el siempre sencillo método de "plug and play" (enchufar y listo) hacen que este sea uno de los aspectos más importantes de QuickDraw 3D. El que todos sus componentes sean ampliables hace de QuickDraw 3D una de las arquitecturas más abiertas de la industria informática (en lo que se refiere a nivel de Sistema).



Ejemplos de transformaciones y translación de un objeto 3D por coordenadas cartesianas. ▼



El gestor de aceleración 3D hace posible aplicar, como en tantos otros aspectos del Macintosh, cualquier tipo de ampliación estando de ese momento lista para trabajar con ella. Gracias este gestor, el usuario podrá añadir a su equipo tarjetas de aceleración, y las aplicaciones podrán sacar partido de ellas inmediatamente.

### Soporte multiplataforma y requerimientos

QuickDraw 3D asegura y protege en cierta forma el trabajo del desarrollador, tanto en tiempo como en lo que se refiere a su apuesta por la innovación. QuickDraw 3D ha sido desarrollado para ser independiente de la máquina en que se ejecuta. Así, QuickDraw 3D para Windows aparecerá poco después de que lo haga la versión Macintosh. Este API para Windows disfrutará también de las tradicionales ventajas del software de Apple: facilidad de uso, consistencia de interfaz, etc.

Cabe mencionar que este es el primer API de Apple específico para el nuevo microprocesador PowerPC, con lo cual a sus posibilidades en cuanto al tratamiento de infor-

mación para el cual ha sido desarrollada esta tecnología, se garantiza el máximo de velocidad. Eso sí, en el aspecto negativo, el que este optimizado para la ejecución sobre procesadores PowerPC hace que los usuarios de los Macintosh basados en procesadores 680x0 no puedan disfrutar de las posibilidades de esta nueva tecnología. Esto es así porque, entre otros aspectos, QuickDraw 3D requiere del procesador PowerPC por su óptimo rendimiento en operaciones de coma flotante; y no hay que olvidarse que precisamente en el tratamiento de objetos tridimensionales se maneja una enorme cantidad de cálculos en este tipo de operaciones.

Por tanto, el equipo mínimo con el que deberá contar cualquier usuario para poder obtener los beneficios de la tecnología QuickDraw 3D es: cualquier modelo Macintosh que implemente un procesador PowerPC, versión del Mac OS 7.1.2 o superior; disco duro, sin grandes requerimientos de capacidad para la instalación de QuickDraw 3D; y contar con un mínimo de 16 MB de memoria RAM.

### OpenDoc y QuickDraw 3D

QuickDraw 3D dará sus servicios a cualquier aplicación que haya sido preparada para soportar este API, o el formato 3DMF. En un futuro próximo, OpenDoc (la arquitectura de documentos codesarrollada por Apple, Novell WordPerfect, IBM y otros) proveerá de estos servicios para el tratamiento de la información en 3D en cualquier documento. Los futuros editores de terceras partes que soporte QuickDraw 3D permitirán al usuario incorporar gráficos 3D en sus documentos con las técnicas de arrastrar y soltar (drag & drop).

Lo que está claro en todo momento, como Apple ha venido demostrando a lo largo de este tiempo, es que mediante el lanzamiento de una nueva tecnología se busca la máxima integración posible no sólo con el resto de las tecnologías de Apple, sino también entre plataformas, con lo cual el usuario será cada vez más independiente de la plataforma que se utilice.

### Soluciones reales a problemas reales

Los usuarios en cuyo trabajo cotidiano esté involucrado como parte fundamental la creación de gráficos de negocios; usuarios cuyo trabajo este basado fundamentalmente en el diseño gráfico, o los ilustradores tradicionales, han estado buscando una forma fácil de incorporar imágenes 3D en sus trabajos, pero se han tropezado con evidentes dificultades para hacerlo con herramientas 2D. La creación de gráficos 3D ha sido, durante mucho tiempo, el coto privado de artistas gráficos altamente especializados, ilustradores técnicos e ingenieros.

Para suplir estas deficiencias, y otorgar de una forma sencilla a los usuarios la posibilidad de la generación de imágenes tridimensionales a partir de imágenes bidimensionales, salieron al mercado una serie de herramientas capaces de generar (de forma más o menos satisfactoria) este tipo de transiciones.

QuickDraw 3D se ha diseñado para eliminar muchos de los obstáculos con los que tradicionalmente se encuentran los usuarios no familiarizados con aplicaciones que manejen gráficos 3D, y una mayor productividad en los que ya están familiarizados con este tipo de aplicaciones.

### Facilidad de uso

Una de las objeciones más comunes a la hora de trabajar en 3D es que se requiere el aprendizaje de nuevas técnicas. Sin embargo, en el diseño de QuickDraw 3D se ha partido de las técnicas de manipulación de imágenes 2D, extendiéndolas a la tercera dimensión. Gracias a esta aproximación, los profesionales que ya trabajaban con imágenes 2D podrán reutilizar todos sus conocimientos y aplicarlos al manejo de las imágenes 3D.

### Características generales del API QuickDraw 3D

QuickDraw 3D instala unas librerías compartidas, a las que cualquier aplicación puede acceder a través de la toolbox. Es decir, una vez que se han instalada, cualquier aplicación escrita con soporte de QuickDraw 3D, puede



**QuickDraw 3D  
requiere del  
procesador  
PowerPC  
para ofrecer  
resultados  
óptimos  
en el trabajo  
operaciones  
de coma  
flotante**



QuickDraw 3D

acceder a este código. Este API, como ya se ha dicho, define multitud de objetos geométricos (alguno de los cuales son: polígonos, triángulos, líneas, polilíneas, puntos, elipses, cilindros, conos, etc.). Además, gracias a su arquitectura extensible, QuickDraw 3D permite a los desarrolladores que sus aplicaciones añadan su propia información a los objetos QuickDraw 3D. Esta interesante característica puede usarse, por ejemplo, para añadir códigos identificativos en inventarios.

El API QuickDraw 3D soporta cuatro tipos de iluminación: ambiente, direccional, de punto y "spot." En futuras versiones de QuickDraw 3D será posible también que el desarrollador cree sus propios tipos de iluminación. De todos estos tipos se hablará más adelante. Adicionalmente se incorporan tres tipos de estilo de iluminación: nulo (null), a utilizar cuando no se necesiten este tipo de efectos; Lambertian y Phong, que proveen de diferentes niveles de refinamiento y complejidad.

Se incorporan también un buen número de diferentes tipos de transformaciones que pueden realizarse con los objetos QuickDraw 3D (rotación alrededor de un eje, rotación alrededor de un punto, escalado, translación de puntos a lo largo de un eje específico, etc.).

En cuanto a los procesos de creación, modificación, ensamblado y colocación de un objeto en un determinado escenario, debe volver-

se a destacar la importancia del esfuerzo de estandarización en la interfaz de usuario, y la disponibilidad para el desarrollador de un kit conteniendo los elementos básicos para implementar esta interfaz en sus aplicaciones. La combinación del API QuickDraw 3D con su interfaz de usuario estándar, el procesador PowerPC, el "renderer" interactivo (conversión de la representación matemática de un objeto 3D a píxeles en la pantalla, que es de dos dimensiones), junto a otras características, hacen posible que el usuarios pueda manipular objetos 3D directamente en la pantalla, de la misma manera que actualmente lo hace con los objetos 2D. Además, QuickDraw 3D disfruta de una arquitectura abierta en lo que se refiere al soporte de nuevos dispositivos de entrada, y a la posibilidad de asociar determinadas herramientas a un objeto concreto, de tal forma que al seleccionar dicho objeto, automáticamente, se habilitarían las herramientas adecuadas para su correcta manipulación.

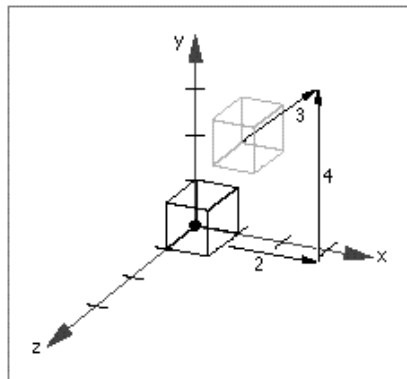
En otro orden de cosas, y en lo que se refiere a la presentación en pantalla, los objetos 3D resultan ser sólo una descripción matemática de la geometría del volumen de una figura o modelo. El objeto, por sí mismo no dispone de cualidades o apariencias específicas (entendidas como textura de los objetos: metálico, mineral, madera, brillo, lustre, etc.). QuickDraw 3D puede aplicar sobre los objetos 3D este tipo de



El API  
QuickDraw 3D  
soporta cuatro  
tipos  
de  
iluminación:  
ambiente,  
direccional,  
de punto  
y "spot"



Con QuickDraw 3D puede manejarse el objeto tridimensional en modo render. ▲



matices (shader o sombreador). Y además, y nuevamente, incorpora un arquitectura abierta que permite a los desarrolladores implementar sus propios algoritmos de texturización, modelos de iluminación, efectos especiales (como ahumados y nieblas). También pueden crear "shaders" que usen un lenguaje tal que el usuario (y no el desarrollador) pueda alterar estos matices sin complejidad.

Como ya se ha dicho, las operaciones de "rendering," convierten la representación matemática de objetos 3D a puntos en la pantalla, incorporando información de sus diferentes atributos, matices y texturas. Aunque muchas operaciones de "rendering" terminan al completar la representación de la escena, hay ocasiones en que el usuario necesita una mejor percepción de como quedará el objeto en la imagen final.

QuickDraw 3D añade un "renderer" interactivo que puede producir, rápidamente, una imagen de buena calidad que el usuario puede continuar editando y manipulando. Pero además, también hace posible a los desarrolladores crear sus propios "renderers" como librerías compartidas.

Debido a que las operaciones de rendering requieren de operaciones intensivas en el ordenador, pueden verse tremendamente beneficiadas de aceleradores especializados (en el mercado hay varias tarjetas, programables, de aceleración en procesos matemáticos).

QuickDraw 3D incorpora una aceleración por puntos de ruptura que, automáticamente, puede redireccionar las tareas de rendering del procesador de propósito general al acelerador especializado. La característica "plug & play" del componente software encargado de realizar la gestión de aceleración permite disponer de un gran número de hardware potencialmente utilizable en los sistemas Macintosh.

Las operaciones de rendering pueden llegar a ser extremadamente complejas, llegando incluso a tardar varios días en realizarse (en el caso de los trabajos más complejos, compuestos de un buen número de objetos, con varios focos de luz y gran cantidad de texturas aplicadas sobre cada uno de los objetos). Una posible solución es la utilización de sistemas con varios pro-

cesadores, bien conectados en paralelo, o bien distribuidos a través de la red. QuickDraw 3D soporta ambos tipos, paralelo y distribuido.

## En un futuro próximo

Esta primera versión de QuickDraw 3D no es sino la base para los futuros desarrollos de Apple en la tecnología de gráficos 3D. En este sentido las líneas maestras de estos esfuerzos serán:

- Después de esta primera versión de QuickDraw 3D para Macintosh aparecerá la versión Windows de este API.
- Arquitectura expandible de "shaders" y "rendering."
- "Renderer" de previsualización. Para hacer más natural el trabajo con imágenes 3D, QuickDraw 3D incorporará en próximas visiones un "renderer" de previsualización de alta calidad, que soportará sofisticados efectos de iluminación, matices y texturas.
- Adiciones y mejoras en las definiciones geométricas.
- Nuevo kit de herramientas para el desarrollador. En próximas versiones se incluirá un conjunto de herramientas para hacer todavía más fácil al desarrollador la implementación de la interfaz de usuario estándar e QuickDraw 3D.

## Modelado y rendering

En la creación de imágenes en tres dimensiones deben realizarse dos importantes tareas: el modelado y el "rendering" o posterior generación de la imagen sintética. El modelado es el proceso con el que se crea una representación de un objeto real o abstracto. El "rendering" es el proceso final para la creación final de una imagen. QuickDraw 3D subdivide, a su vez, cada una de estas tareas en una serie de subtareas. El modelado incluye:

- Creación, configuración y posicionamiento de los objetos geométricos básicos, o grupos de objetos. QuickDraw 3D define un buen número de objetos geométricos básicos y también gran cantidad de medios para transformarlos.
- Asignación de conjuntos de atributos a los objetos, o a partes de los objetos.



Los desarrolladores pueden implementar sus propios algoritmos de texturización, aumentando de esta forma la potencia



- Aplicación de texturas a las superficies.
- Configuración de un modelo de iluminación y calidad de material a los objetos.

Mientras que el módulo encargado de realizar las labores de “rendering” incluye:

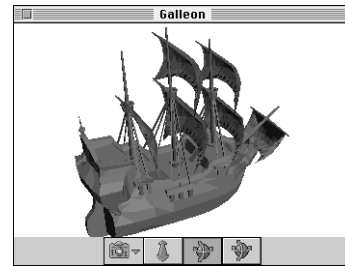
- Especificación del tipo la posición de la cámara. El tipo de cámara se define por el método de proyección del modelo en una superficie plana. QuickDraw 3D dispone de dos tipos de cámaras para este fin.
- Especificación de un “renderer” o de un método para la ejecución de este tipo de operaciones.
- Creación de una vista, o colección de un grupo de luces, una cámara y un “renderer” y sus estilos; así como presentar el modelo dentro de la vista, para crear la imagen final.

Estas tareas de modelado y rendering no son fácilmente separables, particularmente en aplicaciones que soporten el modelado 3D interactivo. Cuando el usuario, por ejemplo, selecciona una esfera y la arrastra a otra posición, la aplicación deberá efectuar cambios en el modelo (recolocar la esfera y crear una nueva imagen), en realidad, la aplicación puede generar una serie de nuevas imágenes para mostrar la esfera cambiando de posición mientras el usuario la mueve. QuickDraw 3D soporta una tercera e importante tarea, la interacción con un modelo; es decir, la selección y manipulación de objetos en el modelo.

## Capacidades adicionales de QuickDraw 3D

Efectivamente, sus capacidades pueden aumentarse fácilmente de diferentes formas. A continuación están recogidas las principales forma de realizar esta tarea:

- Definiendo tipos de objetos propios que aumentan el estándar de tipos ofrecidos por el API QuickDraw 3D.
- Definiendo atributos propios y asignándolos a cualquier objeto.
- QuickDraw 3D es multiplataforma. Si bien en un principio sólo está disponible en entorno Mac OS, también lo estará para otros sis-



**D**iferentes transformaciones de una misma imagen 3D, utilizando una sencilla interfaz de usuario.



temas operativos. Esta portabilidad a otros sistemas gráficos se realizará aislando todo aquello que sea específico de dicho sistema en una capa denominada “draw context.” QuickDraw 3D manejará todas las posibles variables dependientes del sistema de forma transparente.

- QuickDraw 3D es capaz de utilizar aceleradores hardware, si están presentes en la máquina.

Además se define un formato de archivo independiente de la plataforma en la que se utilice, para guardar e intercambiar información 3D entre aplicaciones y sistemas diferentes.

## Orientación a objetos

QuickDraw 3D está orientado a objetos, prácticamente todo lo que el desarrollador puede hacer con este API lo hará creando y manipulando objetos QuickDraw 3D. Un objeto se entiende como ocurrencia de una clase, que dispone de una serie de métodos y propiedades. En otras palabras, los objetos de QuickDraw 3D son un conjunto de información que define las características específicas del objeto, y un conjunto de métodos que definen su comportamiento.



Veremos a continuación las clases de objetos más importantes.

- **Element.** Es cualquier objeto que puede formar parte de un conjunto de objetos. Generalmente guardan información sobre colores, posiciones o datos definidos por la aplicación.

- **Pick.** Se usa para especificar y retornar información relativa a la selección de objetos en un modelo. En general se usa para obtener información sobre los objetos seleccionados por el usuario.

- **Shared.** Es aquel objeto que puede ser compartido por otros dos o más objetos. Por ejemplo un determinado renderer puede asociarse a diversas vistas, o un determinado mapa de píxeles puede usarse como textura en muchos objetos de diferente tipo.

- **View.** Recolecta información acerca de la apariencia y posición de los objetos en el momento de "rendering."

Del objeto "shared" se definen una serie de subclases que también deben citarse:

- **Draw context.** Mantiene información específica sobre un sistema gráfico en particular, o sobre el dispositivo en el que deben efectuarse las operaciones de dibujo.

- **File.** Utilizado para acceder al disco, o a la memoria.

- **Renderer.** Utilizado para crear una imagen desde una vista y un modelo. Controla varios aspectos del modelo y de la imagen resultante.

- **Set.** Colección de uno o más elementos.

- **Shape.** Afecta a cómo el "renderer" llevará a cabo su trabajo. Por ejemplo, una luz es un objeto de tipo "shape," porque afecta a la iluminación del objeto en el modelo.

Finalmente, del objeto "set" se definen las siguientes subclases:

- **Camera.** Se usa para definir un punto de vista, un rango de objetos visibles y un método de proyección para generar una imagen en un dispositivo de dos dimensiones, como la pantalla del ordenador, desde un modelo de tres dimensiones.

- **Geometric object.** Describe un tipo particular de figura geométrica, como por ejemplo un triángulo.

- **Group.** Utilizado para recolectar objeto en listas, o en jerarquías de modelos.

- **Light.** Provee de iluminación a las superficies en una escena.

- **Shader.** Provee de calidad de material a los objetos (en función de la texturización y la luz asociada. Por ejemplo, metálicos, madera, etc.).

- **Style.** Determina algunas de las características básicas usadas en las operaciones de "rendering" de curvas y superficies en una escena.

- **Transform.** Empleado para modificar la apariencia o comportamiento de un objeto.

## Modos retenido e inmediato

Un sistema gráfico opera en modo retenido si retiene una copia de toda la información que describe el modelo gráfico con el que está tra-



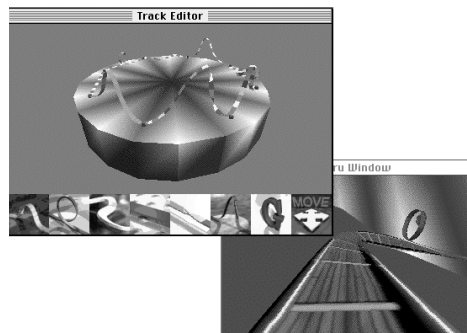
El formato de archivo

es

completamente independiente a la plataforma utilizada a la hora de crear el proyecto



Con QuickDraw 3D es posible editar en una ventana, y en otra poder visualizar en un breve lapso de tiempo el nuevo render.



bajando. Es decir, un sistema gráfico en modo retenido requiere que el desarrollador, en sus aplicaciones, especifique el modelo con el que va a trabajar pasándosele a través de unas estructuras de datos ya predeterminados.

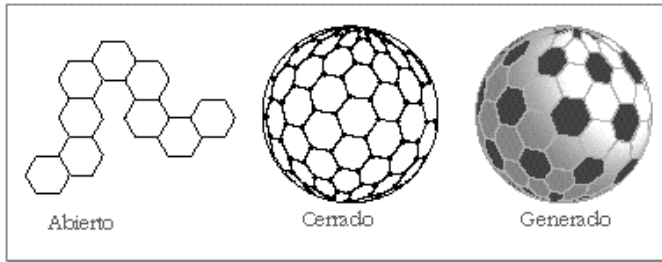
El sistema organiza estos datos internamente, generalmente en una base de datos jerárquica. Una vez que el objeto se ha añadido a esta base de datos, el objeto puede ser modificado, pero sólo a través de rutinas y servicios específicos ofrecidos por el sistema gráfico.

Por el contrario, un sistema gráfico opera en modo inmediato si la aplicación es la que mantiene la información que describe el modelo con el que se está trabajando. Por ejemplo el QuickDraw inicial es un sistema gráfico en dos dimensiones que opera en modo inmediato, el desarrollador dibuja objetos en la pantalla con QuickDraw llamando a rutinas que especifican por completo el objeto a ser dibujado, QuickDraw no mantiene ninguna información internamente acerca de él.

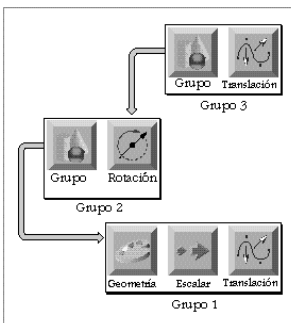
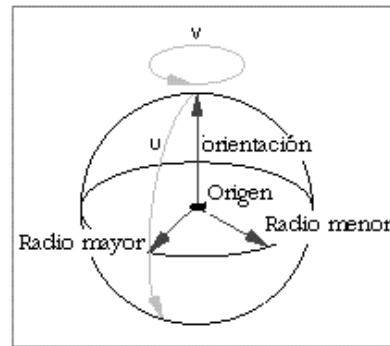
QuickDraw 3D soporta ambos modos, inmediato y retenido, para la especificación y dibujo de los modelos con que trabaja.

La principal ventaja del modo inmediato es que la información acerca del modelo está disponible inmediatamente, y no está duplicada por el sistema gráfico, se guarda en el formato que el desarrollador quiera y puede cambiarla en cualquier momento, sin restricciones. Su principal desventaja es que el desarrollador debe mantener siempre, y constantemente, y que las operaciones geométricas a realizar con esta información también serán por su cuenta.

En cuanto al modo retenido, soporta más altos niveles de abstracción que el modo inmediato y se adapta mejor al hardware de aceleración y caché. Además la estructuración jerárquica del modelo permite al sistema gráfico efectuar las operaciones de actualización de forma extremadamente rápida cuando esta información cambia. Su principal desventaja es que los datos se encuentran duplicados, en la aplicación y en la base de datos del sistema gráfico, además es mucho más complicado cambiar el modelo de datos.



parametrización estándar de un elipsoide



orden en que se aplican las transformaciones



### 3D Viewer

Esta librería provee de un método de visualización sencillo y rápido a las imágenes 3D. Incorpora un pequeño conjunto de herramientas para una limitada interacción del usuario con dichas imágenes.

Una instancia de 3D Viewer es un objeto viewer. Cada uno de estos objetos está asociado con una ventana, en la que la imagen debe estar enteramente contenida. El objeto viewer puede ocupar por entero la región de la ventana, o sólo una pequeña porción.

Un objeto viewer, en su creación y al mostrarlo al usuario, consiste en un área de dibujo que contiene la imagen a visualizar y una barra de herramientas (denominada “controller strip”), que puede contener una o más herramientas de este tipo. También puede encontrarse un “badge,” esto es, un elemento visual que se muestra cuando la barra de herramientas no es visible, y que se hace visible al hacer un clic sobre ella.

El “badge” informa al usuario de que la imagen visualizada es del tipo 3D, y no una imagen estática. Una vez que aparece la barra de herramientas, ya no hay forma de volver a ocultarla y hacer que el “badge” reaparezca.

La barra de herramientas está compuesta por una serie de elementos para la manipulación y orientación, desde el punto de vista del usuario, de la imagen 3D. Los controles que incorpora son:

- **Botón Cámara.** Permite seleccionar una cámara. Haciendo clic sobre él un menú desplegable muestra las cámaras disponibles.
- **Botón “Truck.”** Con este botón es posible redimensionar el tamaño de la imagen.
- **Botón Orbits.** Con él pueden efectuarse rotaciones de los objetos.
- **Botón de Zoom.** Permite al usuario alterar el campo de visión de la cámara actualmente utilizada.
- **Botón “Dolly.”** Utilizando este botón podrá mover el objeto a través de la ventana.

## Objetos geométricos

Un objeto geométrico bajo QuickDraw 3D es una instancia de una clase de objetos QuickDraw 3D. Actualmente QuickDraw 3D incorpora muchos tipos de primitivas para objetos geométricos, sin embargo aún no se encuentran totalmente documentados.

Cualquier objeto geométrico puede contener uno o más conjuntos de atributos, que definen las características de parte o de todo el objeto, como por ejemplo color o cualquier otra propiedad de material.

Pueden asociarse estos conjuntos de atributos prácticamente a cualquier parte claramente distinguible de un objeto geométrico, por ejemplo a la cara de un triángulo, a uno o todos sus vértices.

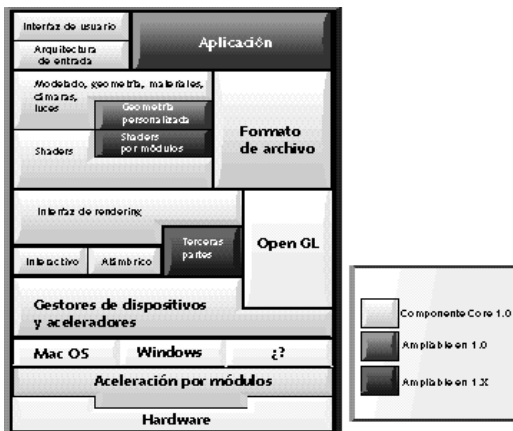
## Las mallas de red

Se denominan “mesh,” y es una colección de vértices, caras y lados que representan una figura sólida compuesta por caras poligonales. Una cara de la malla es una figura poligonal, que forma parte de la superficie de la malla. No se requiere que estas caras sean planos, pero pueden obtenerse resultados inesperados al realizar generaciones sintéticas sobre caras no planas. Estas caras se definen por una matriz que contiene los vértices de la malla.

Debido a que dinámicamente pueden añadirse, o borrarse, lados y vértices, la malla es siempre un objeto en modo retenido.



**QuickDraw 3D** incorpora una gran cantidad de primitivas básicas para la composición de objetos en tres dimensiones



OpenGL es un buen ejemplo de cómo los APIs de terceras partes pueden obtener las prestaciones ofrecidas por QuickDraw 3D. En este caso, la integración está compuesta por el formato de los archivos y el soporte de los gestores de aceleración.



### Curvas NURB

QuickDraw 3D soporta curvas y superficies que pueden ser definidas por una clase de ecuaciones que utilizan ratios paramétricos no uniformes y polinomiales.

Una curva en dos dimensiones representada mediante una ecuación NURB se denomina curva NURB, y una superficie compuesta por este tipo de curvas se denomina patch NURB.

Este tipo de ecuaciones pueden usarse para definir curvas y superficies verdaderamente complejas. Las superficies patch son utilizadas intensivamente en las imágenes 3D, porque las transformaciones de sus puntos de control no varían con la escala, rotación, traslación y perspectiva.

### Parametrización de superficies

Para muchas operaciones de modelado, particularmente las que afectan a su textura, QuickDraw 3D necesita realizar un mapeo entre la textura y la superficie. Las texturas se especifican, típicamente, por un mapa de puntos ("pix-map," matriz rectangular de puntos).

QuickDraw 3D define una parametrización estándar de superficies para muchos de sus objetos geométricos.

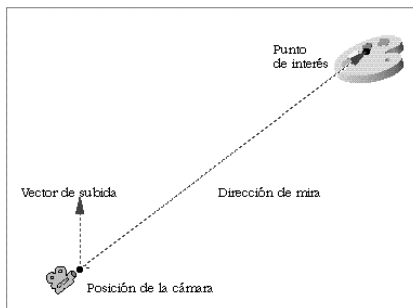
### Atributos de objetos

Es un tipo de objeto tal que determina alguna de las características de un modelo, como pueda ser el color o transparencia. En general los atributos definen propiedades del material de las superficies de un objeto en un modelo. Se definen muchos tipos de atributos, entre los que se encuentran color difuso, especular, transparencia, etc.

En general no se aplican a objetos individualmente. Normalmente se crea un conjunto de atributos, colección de uno o más atributos.

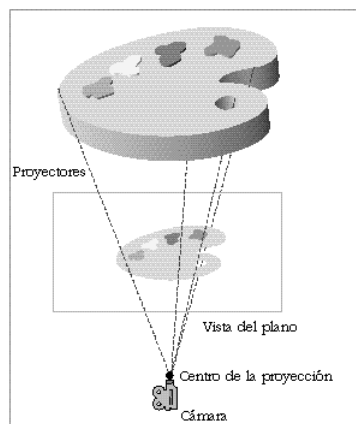
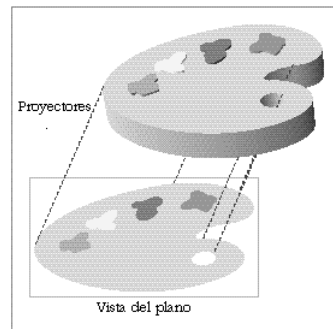
### Tipos de atributos y conjuntos de atributos

Como ya se ha dicho se definen un buen número de tipos de atributos básicos. Si éstos no son suficientes para las necesidades del desarrollador, puede definir sus propios tipos de atributo.



**C** Cámara en perspectiva

**P** Proyección paralela de un objeto



**P** Proyección en perspectiva de un objeto partiendo de un objeto plano.

También puede definirse un tipo de atributo denominado "metahandler," que contiene métodos para manejar información de los atributos creados por el desarrollador (QuickDraw 3D define "metahandlers" para todos sus tipos de atributos básicos). Puede ligarse un conjunto de atributos a una vista, grupo de objetos, un único objeto geométrico, la cara o vértices de un objeto. Para cada uno de estos niveles se definen una serie de atributos naturales.

## Herencia de atributos

Durante el rendering de los objetos en una vista, los conjuntos de atributos de objetos más altos en la jerarquía son heredados por los objetos más bajos de la misma. Por ejemplo, si el conjunto de atributos de una lista específica un determinado color difuso, todos los objetos de dicha vista serán mostrados, al efectuar el rendering, con el color difuso.

La herencia de atributos siempre ocurre en el orden vista, grupo, objeto geométrico, cara, vértice.

Puede decirse entonces que los atributos de vista son heredados por todos los grupos de objetos en el modelo, a no ser que un grupo contenga otro conjunto de atributos que anulen dicha herencia. De forma similar, cualquier atributo asignado a un objeto geométrico será heredado por todas sus caras, a no ser que alguna de ellas contenga otro conjunto de atributos que lo anulen.

Esta herencia se aplica sólo a los atributos naturales contenidos en cualquier conjunto de atributos. Así pues, si un conjunto de atributos de una vista contiene un atributo de superficie no natural para los conjuntos de atributos de vista, este atributo no será heredado por ninguno de los objetos más bajos que la superficie en la jerarquía.

## Objetos estilo

Este es un tipo de objeto que determina algunas de las características básicas del dibujado de curvas y superficies en una escena. Algunos de los tipos de estilo definidos son: interpolación, relleno, subdivisión, sombreado, etc.

Al contrario que los atributos, que definen características sobre el aspecto de superficies individuales y sólo pueden aplicarse a parte de un modelo, los estilos definen características de un renderer y generalmente, no siempre, a un modelo completo.

## Objeto Transform

Los objetos "transform" pueden usarse para modificar o transformar la apariencia o comportamiento de los objetos QuickDraw 3D dibujables. Las transformaciones son muy usa-

das, ya que no alteran la representación geométrica de los objetos. Se definen, por ejemplo, transformaciones de traslación, escala, rotación, rotación alrededor de un punto, etc.

Estas transformaciones se producen en un espacio, en un sistema de coordenadas, que puede definirse como una asignación, plana o espacial, de posiciones a objetos. Usualmente QuickDraw 3D trabaja con sistemas de coordenadas cartesianas, en los que la posición de un punto en el plano, o en el espacio, está determinada por la proyección del punto dentro del eje de coordenadas, que son líneas mutuamente perpendiculares, y que intersectan en un punto denominado origen.

También puede utilizarse otro tipo de sistemas de coordenadas, como el polar (que asigna posiciones a los objetos en términos de distancias desde el origen y a lo largo del radio que forman un ángulo 'F' con una coordenada fija). También está disponible el sistema de coordenadas esférico (que asigna posiciones a los objetos en función de sus distancias desde el origen a través el radio el ángulo 'f' con una coordenada fija y otro ángulo con otra coordenada fija).

QuickDraw 3D dispone de rutinas de conversión entre estos tres tipos de sistemas de coordenadas.

## Objetos Light

Proveen de iluminación a las superficies en una escena. La iluminación de superficies, en general, se ve afectada por varias fuentes de luz. Así pues, una vista se asocia a un grupo de luces.

QuickDraw 3D soporta múltiples fuentes de luz, y múltiples tipos de luz en una escena. Se definen cuatro tipos de luces: ambiente, direccional, de puntos y de mancha.

La iluminación ambiente es una cantidad de luz, de un determinado color, que se añade a la iluminación de todas las superficies en la escena. Si no se utiliza otro tipo más que éste, la iluminación ambiente ilumina la escena con una luz plana y uniforme. No puede producirse con ella efectos de atenuación en función de la distancia ni sombreados.



QuickDraw 3D  
trabaja  
con un  
sistema  
de  
coordenadas  
cartesianas.  
la posición  
de un punto  
es su propia  
proyección



La iluminación direccional parte de una fuente que emite rayos de luz paralelos en una dirección específica. Este tipo de luz no tiene localización, y no pueden utilizarse efectos de atenuación, pérdida de intensidad en función de la distancia, ni sombreados).

La iluminación por puntos emite rayos de luz en todas direcciones desde una posición específica. El efecto de iluminación conseguido es dependiente de su intensidad y color, así como de la orientación de la superficie a iluminar y su distancia a la fuente de la iluminación.

La iluminación de mancha emite un cono circular de luz en una dirección específica, desde un determinado punto.

### Objetos cámara

Usado para definir un punto de vista, un rango de objetos visibles, y un método de proyección para la generación de imágenes en dos dimensiones de estos objetos partiendo de un modelo tridimensional.

QuickDraw 3D define tres tipos de cámaras: "orthographic", "view plane" y "aspect ratio." La localización de la cámara es su posición en el sistema de coordenadas, su ubicación es la localización con una orientación y dirección (la orientación la indica su vector superior, qué está encima de ella; la dirección la indica el punto de interés, el punto hacia el que la cámara apunta).

Los diferentes tipos de cámaras se diferencian por sus métodos de proyección. La proyección de un objeto es el conjunto de puntos en donde los rayos que parten del ob-

jeto (denominados proyectores) intersectan un plano ("view plane").

La proyección creada cuando todos los proyectores son paralelos a otros, se denomina proyección paralela, mientras que la proyección creada cuando los proyectores intersectan en un punto se denomina proyección en perspectiva.

Las cámaras "ortographic" usan una proyección paralela para generar las imágenes. Este tipo de proyecciones son menos realistas que las de perspectiva. Las dos proyecciones más comunes con este tipo de cámaras son la isométrica y la de elevación.

Las cámaras "view plane" utilizan proyección en perspectiva. Generalmente se usan para crear una imagen en perspectiva de un objeto específico de la escena.

Las cámaras "aspect ratio" también son de proyección en perspectiva, definidas en función del ángulo de visión y el ratio de horizontal a vertical.

### Objetos Group

Se usa para recolectar objetos en una lista jerárquica de modelos. Los objetos que se colocan en un grupo no se copian a él, sino que el grupo mantiene referencias a dichos objetos. Además, un grupo puede contener otros grupos.

Existen tres subclases del objeto group: light, display (grupo de objetos dibujables, que incluyen objetos geométricos, estilos, transformaciones y atributos). Esta última subclase, a su vez "ordered display," en la que al colocar el grupo en una vista, los objetos se visualizan en



Los objetos de cámara se utilizan para definir el punto de vista, un rango de objetos visibles y un método de proyección



Los componentes del API de QuickDraw 3D han sido diseñados para ofrecer herramientas a los desarrolladores. El sistema de búsquedas ha sido potenciado con la posibilidad de efectuarlas en función de más de un criterio, así como por la forma de mostrar el resultado de la búsqueda.

Apple	Desarrollador	Usuario
Arquitectura de interfaz	Toolkit	Sencillo de usar
Formato multiplataforma	Tipo de datos comunes	Cortar y pegar
API extensible	Oportunidades de negocio	Soluciones escalables
	Aceleración Hardware	Flexibilidad

## Microsoft y las tres dimensiones

Microsoft ha añadido a Windows NT 3.5 (y promete hacerlo también para todos sus Sistemas Operativo de 32 bits) una librería gráfica denominada OpenGL, que incorpora generación sintética de imágenes 3D y capacidades de animación.

OpenGL es un estándar de la industria informática, basado en una librería gráfica interna de Silicon Graphics. Ha sido diseñado, y mantenido por una alianza de fabricantes compuesta por SGI, Microsoft, IBM, Intel y DEC.

Hasta hace poco sólo podía encontrarse en estaciones de trabajo UNIX. Ahora, gracias a la disponibilidad de esta interfaz para gráficos 3D, además de la utilización de hardware especializado en el proceso de rendering de imágenes 3D, es posible ya crear complicadas y realistas escenas en Windows.

### Primitivas OpenGL

OpenGL provee de una serie de primitivas para puntos, líneas y polígonos, en cualquier caso el desarrollador puede crear basándose en estas primitivas. La librería también incorpora rutinas para el dibujo de curvas, superficies o texto; también pueden crearse polígonos rellenos.

Una vez creada una escena, pueden especificarse efectos de iluminación (como transparencias o nieblas) y el ángulo de visión, OpenGL cuida del resto. Si se cambia el punto de vista, OpenGL se encarga de recalcularse la escena. Una vez que se han creado los objetos a utilizar, pueden alterarse dinámicamente su localización y rotación, el punto de vista y efectos de iluminación.

OpenGL ha sido diseñado para ser eficiente en el modelo cliente/servidor. Un ejemplo de configuración típica: un potente ordenador genera los comandos de dibujo (el servidor), mientras que una estación de trabajo en la red (el cliente), recibe estos comandos y ejecuta el "rendering" en su pantalla. Pero OpenGL también trabaja correctamente en una única máquina, haciendo ella sola ambos papeles, cliente y servidor. La parte difícil es aprender a crear una escena OpenGL y la interfaz adecuada con Windows, ya que OpenGL (como librería independiente de hardware que es) no conoce nada acerca de Windows y sus dispositivos.

### Librerías OpenGL

Con la versión NT de OpenGL se distribuyen tres librerías: `opengl32.lib` y `glu32.lib` son rutinas de apoyo para OpenGL, que dan servicios como creación de esferas, ejecución de operaciones con matrices. La tercera librería no forma parte estricta de OpenGL, y es una librería auxiliar; y no sería necesario incluirla en muchos programas OpenGL, pero permite ignorar los detalles específicos de Windows para concentrarse en las partes puramente OpenGL. Se dispone también de una serie de rutinas de interfaz específicas que permiten a OpenGL trabajar en una plataforma Windows.

un determinado orden (transformaciones - estilos - conjuntos de atributos, etc). Y "proxy display group," que contiene varias representa-

ciones de un objeto geométrico. De esta forma pueden encapsularse, en un metafichero, dos o más descripciones de un objeto.

Esto es muy adecuado cuando una aplicación, al leer de un archivo, no es capaz de entender alguna descripción. Por ejemplo, si el desarrollador sabe que otras aplicaciones no soportan los "patch NURB," pero sí la red de mallas, puede preparar un "proxy display group" con las dos descripciones de la superficie. Posteriormente, la aplicación que lea el metafichero podrá utilizar cualquiera de las dos descripciones de la superficie, o la que sí sea capaz de entender. El orden de las descripciones es de la preferida a la menos preferida.

### Objetos renderer

Es el objeto utilizado para poner un modelo, esto es, crear una imagen desde una vista y un modelo. Los objetos "renderer" controlan varios aspectos del modelo y de la imagen resultante, como por ejemplo los tipos de objetos geométricos que debe dibujar sin descomponerlos en objetos simples, las partes de objetos a dibujar (por ejemplo sólo los lados, o las caras rellenas), los tipos de luz disponibles y el modelo de iluminación a utilizar.

Nuevamente, están disponibles diversos tipos de "renderers," concretamente dos: "wireframe" e "interactive".

El primero crea las líneas de dibujo de los modelos, opera de forma extremadamente rápida y con una carga en memoria comparativamente pequeña. Debido a que es un simple dibujador de líneas no puede iluminar las superficies, o dotarlas de características de material. Por tanto este tipo de "render" ignora los grupos de luz asociados a una vista.

El segundo, "interactive," emplea un algoritmo rápido para el dibujo de elementos sólidos y de superficies. Sin embargo es más lento que el anterior, y requiere mayor cantidad de memoria que el "wireframe" o alámbrico. Cuando el tamaño del modelo es razonable, y sólo se realizan operaciones sencillas de iluminación, funciona con suficiente rapidez. Este "renderer" es capaz de dibujar los modelos con un alto grado de detalle, aunque

a expensas de tiempo de ejecución y cantidad de memoria.

### Objetos Draw Context

El API QuickDraw 3D es capaz de direccionar su salida (una imagen) en uno o más destinos. Por ejemplo, puede poner una imagen 3D en una ventana estándar de Macintosh, y podrá hacerlo también en una de Windows. Para cumplir con esta capacidad multiplataforma se usan los objetos "draw context," que mantienen información específica acerca de las ventanas de un determinado sistema gráfico, o destino del dibujo.

En general, QuickDraw 3D no duplica los métodos ya existentes para crear y manejar los dispositivos destino del dibujo. Así no contiene ninguna función para crear una ventana, capturar los eventos que se producen en ella, o modificar su tamaño y posición en la pantalla. Los objetos "draw context" contienen una mínima cantidad de información sobre los sistemas gráficos o dispositivos destino.

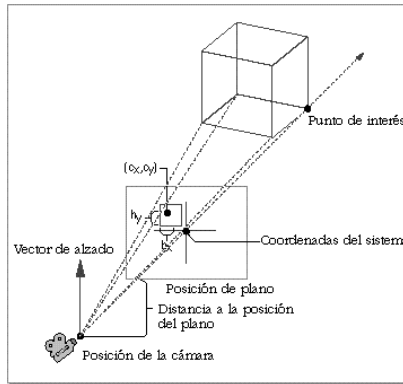
Actualmente sólo se soportan dos tipos de "draw contexts": Macintosh draw contexts y pixmap draw contexts. No todos los posibles destinos del dibujo son una ventana, también puede ser una región de memoria (un pixmap).

### Objetos View

Mantiene la información necesaria para colocar una escena o imagen de un modelo. Esencialmente es una colección de cámara, grupo de luces, "draw context" y un "renderer." Como ya se ha visto, la cámara define un punto de vista dentro del modelo tridimensional y un método de proyectarlo, en dos dimensiones, en un plano de dos dimensiones. El grupo de luces aporta la iluminación de los objetos en el modelo, el "draw context" define el destino de la imagen en dos dimensiones, y el "renderer" determina el método de generación de la imagen desde el modelo.

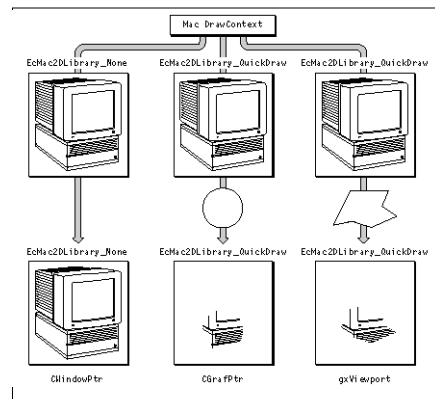
### Objetos Shader

Manipulan los efectos visuales que dependen de la iluminación, el color y otras propiedades del material como reflectancia o textura.



**C** Cámara definida como posición de plano.

**U** so de una librería gráfica en un "Draw Context"



Algunos tipos de "shader" son: el de superficie, aplicable al calcular la apariencia de una superficie; Texturas e iluminación.

QuickDraw 3D soporta shaders de textura, que permiten realizar operaciones de mapeo, técnica en la cual una imagen predefinida, la textura, se mapea a la superficie del objeto en un modelo.

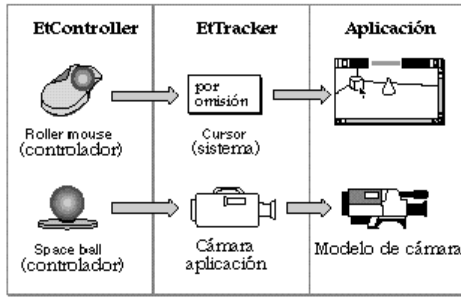
### Objetos Storage

Utilizado para representar una porción física de almacenamiento en el ordenador. Puede tratarse de cualquier tipo de dato. Actualmente QuickDraw 3D soporta tres tipos de formatos: guardados en memoria, en el data fork de un archivo Macintosh y en un archivo con formato UNIX.

### Objetos File

Se usa para leer y escribir los datos que conforman el formato 3DMF (QuickDraw 3D





**D**ispositivos de entrada, control y "trackers."

Object Metafile), un formato de archivo estándar diseñado para facilitar el intercambio de datos tridimensionales entre aplicaciones. Puede usarse en formato guardar datos y en formato intercambiar datos.

La relación entre objetos "file" y objetos "storage" es similar a la existente entre objetos "view" y objetos "draw context," que proveen de una capa de abstracción que lo independiza del sistema operativo sobre el que se esté trabajando.

### Gestor de dispositivos apuntadores

Se trata de un conjunto de funciones usadas para manejar estos dispositivos. Para que el usuario pueda interactuar sin problemas con los objetos en un modelo tridimensional, es necesario disponer de medios para la manipulación de los posiciones del objeto a través de los tres ejes, correspondientes a cada dimensión.

La mayoría de los ordenadores sólo soportan dispositivos de entrada de dos dimensiones, como ratones o tabletas gráficas. QuickDraw 3D dispone de una interfaz estándar entre aplicaciones y dispositivos que permiten al usuario trabajar con cualquiera de los dispositivos 3D.

Se entiende por dispositivos apuntadores cualquier dispositivo físico capaz de controlar movimientos, o especificar posiciones en un espacio tridimensional. Un dato que debe tener en cuenta es la posibilidad de utilizar simultáneamente más de un dispositivo de este tipo.

Todos estos dispositivos son capaces de controlar posiciones, y algunos son capaces de controlar orientaciones. Algunos, como por ejemplo el ratón, pueden disponer de más de un botón; en consecuencia el controlador del dispositivo

debe ser capaz de leer el estado de estos botones (si están pulsados o no lo están).

Un "tracker" es un tipo de objeto que controla la posición, orientación y estado de los botones de un elemento específico de la interfaz de usuario en una aplicación. QuickDraw 3D incorpora un "tracker" que controla la posición y orientación del cursor del sistema. El desarrollador puede crear "trackers" adicionales y ligarlos a otros elementos visibles de la interfaz de usuario. Por ejemplo, a la vista de una cámara, y entonces se permite a usuario controlar la posición y orientación de la cámara usando el dispositivo.

### Utilidades matemáticas

QuickDraw 3D dispone de un importante número de útiles funciones para realizar operaciones matemáticas básicas con objetos como puntos, vectores, matrices, etc. Alguna de las cosas que pueden hacerse con estas utilidades son: ajustar los componentes formados de puntos y vectores; convertir las dimensiones correspondientes de puntos y vectores; restar puntos de puntos; calcular distancias entre puntos; sumar y restar puntos y vectores; escalar vectores; determinar longitud de vectores; normalizar vectores; sumar y restar vectores; transformar puntos y vectores; convertir vectores de su forma cartesiana a polar o esférica; manejo de matrices; y cálculos trigonométricos.

### ¿Para cuándo estará disponible?

Apple ha anunciado como posible fecha de lanzamiento de la tecnología QuickDraw 3D, en su primera versión, el verano de 1995; y se incorporará en los nuevos ordenadores Apple que se lancen al mercado a partir de este periodo de tiempo.

También estará disponible en los kits de Sistema Operativo, y también se servirá como componente extra con aquellas aplicaciones que soporten QuickDraw 3D a través de licencias de desarrollador.

Por otro lado, Apple ofrecerá un completo kit de desarrollo (SDK) disponible a través de CDIS, mediante el cual cualquier usuario podrá implementar las características y el soporte a QuickDraw 3D en sus propios desarrollos.