



---

## ¿Que es DirectX?

Microsoft DirectX es una colección avanzada de interfaces de programación de aplicaciones(APIs) integrada a los sistemas operativos Microsoft Windows.

Este conjunto de APIs mantiene una plataforma de desarrollo de aplicaciones multimedia estándar para aplicaciones Windows, permitiéndoles a los programadores del software acceder al hardware especializado sin tener que escribir código específico de cada tipo de hardware.

Cuando una aplicación o un juego es escrito para DirectX, el programador no tiene que preocuparse por exactamente qué tarjeta de sonido o por el adaptador gráfico que el usuario final podría tener en su máquina. DirectX se encarga de eso por el.

DirectX juega un papel en muchas funciones, incluyendo renderizacion 3D , reproducción de video, interfaces para joysticks y ratones, gestión de redes para multi-jugador y muchos más.

Sin embargo, DirectX tiene una gran desventaja: no es portable, es decir, una aplicación programada con DirectX esta condenada a trabajar solamente en Windows, lo cual no es nada deseable a menos que sepas que el único mercado al que va dirigido la aplicacion son personas con una computadora con Windows.

## ¿Porque Utilizar DirectX?

DirectX proporciona a los programadores una manera estandarizada y amigable de acceder a los recursos de la computadora para programar aplicaciones y juegos aprovechando las ultimas tecnologías de hardware de manera generalizada.

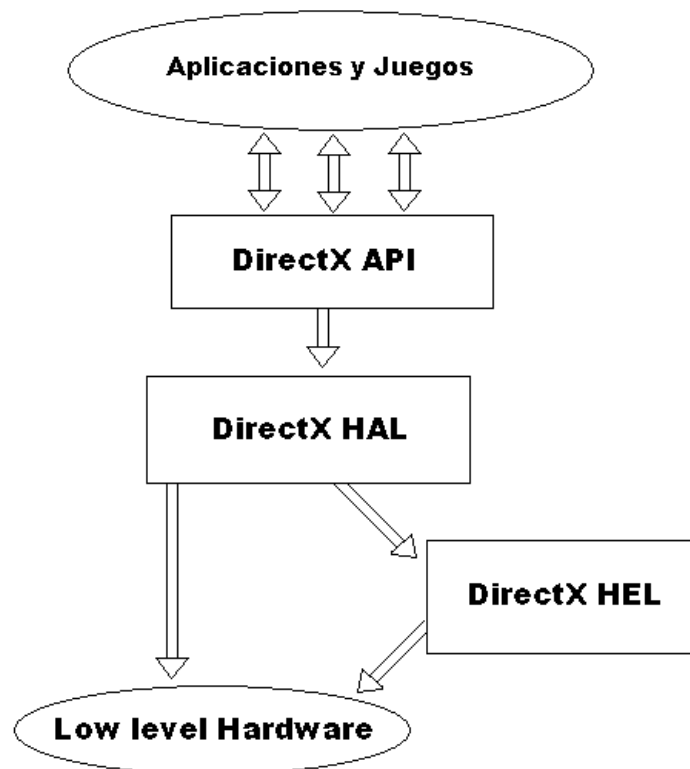
Otra de las principales ventajas de trabajar con DirectX es que no solo soporta la parte de gráficos y de 3D sino posee todas las herramientas para construir aplicaciones completas de alto nivel de una manera en la que el hardware no es una limitación, sino que el programador solo debe conocer el API y este es el que se encarga de saber como realmente funcionan los distintos tipos de hardware.

## ¿Como Funciona DirectX?

Básicamente el programador tiene a mano el API de DirectX de donde se tienen las funciones con las que uno dispone para programar la aplicación.

Este API se comunica con un HAL (capa de abstracción del hardware) diciéndole a esta lo que la aplicación esta queriendo hacer, el HAL es el que realmente conoce que es lo que el hardware puede o no realizar y como lo realiza así que es este el que se comunica con el hardware y de acuerdo a esto le asigna el trabajo, en caso de que nuestro hardware no sea compatible con alguna de las funcionalidades de la versión de DirectX que es esta utilizando existe un HEL (capa de emulación de hardware) que se encarga de emular las funciones que nuestro hardware no puede manejar.

De esta manera se puede mantener la compatibilidad de DirectX con todo tipo de hardware mas viejo y nos permite utilizar aplicaciones nuevas con estos hardwares, pero con cierta penalización en performance o en calidad.



---

## Alternativas a DirectX

### OpenGL ( <http://www.opengl.org> )

OpenGL fue creado por Silicon Graphics en 1992, y se ha convertido en uno de los principales ambientes de programación para desarrollar aplicaciones gráficas que existen.

OpenGL esta diseñado para poderse usar en una gran variedad de sistemas operativos, es decir, que el mismo código que se utilice para desarrollar una aplicaron o juego en Windows puede ser utilizado usarlo para desarrollar en Mac, Linux, etc. sin tener que modificar o modificándolo muy poco, esto puede significar una gran ventaja para cualquier compañía en especial en estos días donde se busca abarcar el mayor mercado posible.

Otra gran ventaja que ofrece OpenGL es que su arquitectura fue pensada para poder evolucionar conforme avanza el tiempo, de modo que pueda adaptarse con facilidad a las novedades que van surgiendo en la industria.

En general, desde que apareció, OpenGL ha sido un ambiente de programación bastante estable y confiable, y quizás la razón por la que es más conocido es porque fue la primera en ofrecer la posibilidad de desarrollar aplicaciones 3D; fue precisamente con OpenGL que se hicieron los primeros juegos 3D como Wolfenstein y Doom, hoy todavía es muy utilizado en muchos de los ultimos juegos que salen al mercado.

### SDL ( <http://www.libsdl.org> )

**Simple DirectMedia Layer** es una librería multimedia diseñada para proveer acceso rápido al buffer gráfico y el sistema de sonido. Es usado por MPEG, emuladores y varios juegos populares, incluyendo el premiado "Civilization: Call To Power" portado en Linux. Otra característica que tiene es que fue diseñado para ser portable a diferentes sistemas operativos como Windows, Linux, BeOS, MacOS, Solaris, IRIX, y FreeBSD sin tener que cambiar ni siquiera una sola línea de código.

Esta librería fue escrita en C, pero trabaja en forma nativa con C++, y puede utilizarse fácilmente con otros lenguajes de programación como Ada, Eiffel, ML, Perl, Python y Ruby.

Una de las características que tiene SDL es que se trata de un proyecto "Open Source", es decir el código fuente está disponible.



---

## **Allegro** ( <http://www.allegro.cc> )

La Allegro Gaming Library es un API que fue originalmente creado por Shawn Hargreaves, y está enfocado específicamente hacia la programación de video juegos.

Una de las principales ventajas que tiene es que fue específicamente diseñado para la programación de video juegos, por lo tanto todas las rutinas que utiliza están hechas para ser fáciles de manejar y sobretodo eficiente desde el punto de vista de un juego.

Con Allegro podrás manipular prácticamente toda la multimedia del juego, tal como la Entrada/Salida, gráficas, midis, efectos de sonido y el tiempo.

Esta API es muy usada especialmente entre la gente que gusta de programar video juegos por hobby o gente que apenas comienza en este pequeño mundo de programación de video juegos, principalmente esto se debe a que es muy fácil e intuitivo de usar, además de que hace el código del programa muy legible.

## **RenderWare**

API para manejo de gráficos 3D desarrollado por Criterion Software. Posee un API muy amigable que incluye funciones para sombreados, mapeado y filtrado de texturas, realizar transformaciones de geometría, clipping, e iluminación. RenderWave incluye un lenguaje de scripts utilizado para guardar objetos en archivos de texto. También incluye una gran librería de primitivas y modelos, desde la versión 2, incluye drivers para utilizar aceleración por hardware, aunque tiene un muy buen desempeño inclusive sin la ayuda del hardware.

## **Historia de DirectX**

### **Los años oscuros**

Antes de 1995 virtualmente todos los juegos se ejecutaban desde DOS, sin ninguna aceleración de hardware de video.

DOS mismo presentaba una interfase hostil para el usuario.

Dificultades para la configuración

Aparece Windows 3.1 con una interfase más amistosa pero como una pobre plataforma para los juegos.

Era Virtualmente imposible lograr una renderizacion customizada rápida.

### **Aparece Windows95**

Diseñado para reemplazar a DOS/ Windows 3.1

Podía correr la mayoría de las aplicaciones de DOS de forma nativa, lo que permitió que las aplicaciones pudieran migrar sin problemas.

Se introdujo un SDK para juegos, mas tarde conocido como DirectX 1.

### **Las metas de DirectX**

Una de las razones primarias para crear DirectX era promover el desarrollo de los juegos para la plataforma Windows. La mayoría de juegos desarrollados para la computadora personal eran basados en MS-DOS. Sin embargo, al diseñar los juegos basados en DOS, los diseñadores, debían programar varias implementaciones de hardware para una variedad de tarjetas que complicaban la instalación

DirectX fue desarrollado para garantizar a los programas basados en Windows un alto rendimiento en el acceso en tiempo real al hardware disponible en las computadoras. Este provee interfaces consistentes entre en los desarrolladores de hardware y software para dar a los programadores acceso de bajo nivel a las capacidades del hardware, como acceso directo a la memoria de video, acceso a la aceleración de video y sonido por hardware, etc, todo esto mediante una capa de abstracción entre el software y los distintos dispositivos de hardware.



---

A partir de este punto, DirectX fue evolucionando, agregando cada vez más funcionalidades, y mejorando otras, en la actualidad, la última versión de DirectX es la 9.0b.

En general DirectX ha venido ganando mucha popularidad en los últimos años entre los programadores de vjuegos, en especial desde que Microsoft comenzó a mejorar la parte de Direct3D haciéndolo más amigable como eficiente, y se ha vuelto una seria competencia para OpenGL, otro de los estandares graficos de gran aceptación.

## **Componentes de DirectX**

**El API de DirectX esta compuesto de los siguientes components:**

### **DirectDraw**

DirectDraw permite el acceso directo a la memoria del adaptador de video, hardware blitters, hardware overlays, and page flipping. DirectDraw proporciona esta funcionalidad manteniendo la compatibilidad con los todos los programas basados en versiones previas de Windows y los drivers de distintos dispositivos de hardware.

Esencialmente, DirectDraw es un administrador de la memoria de video. Usando DirectDraw, un programa puede manipular la memoria de video con facilidad, mientras aprovecha todo tipo de capacidades de los tipos diferentes de adaptadores de video sin depender de un tipo particular de hardware.

### **Direct3D**

Direct3D es un conjunto completo de servicios gráficos 3D en tiempo real, que se encarga de todo el renderizado basado en software-hardware de todo el Pipe-line grafico (transformaciones, iluminación, y rasterización) y el acceso transparente a la aceleración del hardware. Direct3D es completamente escalable, permitiendo que todo o una parte del 3D Pipe-line sea acelerado por el hardware. Direct3D expone las avanzadas capacidades de los aceleradores del hardware 3D, incluyendo el z-buffering, el anti-aliasing, alpha blending, mipmapping, efectos atmosféricos, y perspective-correct texture mapping. La integración con otras tecnologías de DirectX permite a Direct3D tener otras características tales como video mapping, hardware 3D rendering in 2D overlay planes, e inclusive sprites, proporcionando capacidades de gráficos 2D y 3D en aplicaciones Multimedia.

### **DirectSound**

DirectSound es la parte del Microsoft Windows Game SDK (Kid de Desarrollo de Software para juegos) que proporciona utilidades de mezcla de sonido a baja-latencia, aceleración por hardware, y acceso directo al dispositivo de sonido. DirectSound proporciona esta funcionalidad manteniendo la compatibilidad con los todos los

programas basados en versiones previas de Windows y los drivers de distintos dispositivos de hardware. DirectSound 3D permite la aceleración por hardware de



---

sonidos 3D utilizando sistemas surround y un subconjunto de la norma de EAX de Creative.

### **DirectInput**

DirectInput proporciona acceso rápido y consistente a palancas de mando analógicas y digitales (Joysticks). El API de DirectInput mantiene la compatibilidad con Kit de Desarrollo de Software (SDK) Win32 de Microsoft, pero ha mejorado la capacidad de respuesta y la fiabilidad cambiando el modelo del driver de dispositivo.

Los drivers de DirectInput también utilizan el registro de Windows para guardar las configuraciones para las palancas de mando normales, y la información de la calibración para las palancas de mando previamente configuradas.

### **DirectPlay**

DirectPlay es una interfase de software que simplifica el acceso a los servicios de comunicación. DirectPlay provee a los juegos una manera de comunicarse ente si que es independiente del medio de transporte subyacente, protocolo, o el servicio en línea. DirectPlay también hace posible lanzar y correr juegos fácilmente a través de Internet usando el API de DirectX.

### **DirectMusic**

Proporciona a los programadores un API unificado para el manejo de música, aprovechando las características específicas de diversos tipos de hardware y manteniendo la compatibilidad con aplicaciones mas antiguas.

### **Otros componentes:**

**DirectX HAL** (Capa de abstracción de hardware).

Es la Parte de DirectX que representa las capacidades del hardware de manera generalizada y no de una forma dependiente del hardware, en otras palabras hace de puente entre las funciones del API y en hardware en si, de manera a estandarizar las capacidades de los distintos componentes de hardware.





---

### **DirectX HEL** (Capa de emulación de hardware).

Es la parte de DirectX que permite a los programadores desarrollar aplicaciones sin tener que preocuparse de las funcionalidades de los tipos de hardware, ya que el HEL puede emular estas funciones de manera a mantener la compatibilidad con hardware mas antiguo.

## **Evolución de DirectX**

DirectX se ha lanzado como parte del Kit de desarrollo de software para juegos de Microsoft (SDK). DirectX es la etiqueta genérica para cualquier revisión del sistema de DirectX.

### **DirectX 1.0**

Consistió en DirectDraw, DirectInput, DirectPlay, y DirectSound.

### **DirectX 2.0**

Introduce Direct3D.

### **DirectX 3.0**

Marcó la introducción de los archivos especiales de DirectInput, de un applet especial en el panel de control para la administración de joysticks, y un controlador actualizado para el Coprocesador matemático virtual, para obtener las ventajas de la tecnología MMX de Intel. El API de DirectSound fue actualizado con extensiones para 3D.

### **DirectX 3.0a**

Era una actualización de menor importancia a 3.0 que (sobre todo) corrigió algunos problemas en las máquinas MMX.

### **DirectX 3.0b**

Era una actualización muy de menor importancia a 3.0a que corrigió un problema cosmético con la versión japonesa de Windows 95.



### **DirectX 5.0**

Agregó muchas mejoras incluyendo el soporte para dispositivos de retorno de fuerza (force feedback), soporte para múltiples monitores (en Windows 98), un nuevo panel de control de los controladores de juego (joysticks), un mejor soporte de las extensiones MMX, y una interfaz del todo mejorada. DirectX 5.0 se encuentra incorporado en Windows 98 y Windows 98 Segunda Edición.

### **DirectX 5.2**

Incorporó la última versión de DirectPlay, así como otros cambios misceláneos.

### **DirectX 6.0**

Agregó las nuevas características 3D, algunas de las cuales son el soporte para single-pass multitexturing, Bump mapping, la compresión de texturas, y los stencil buffers. Había también un aumento en el performance sobre DirectX 5 y soporte para la tecnología 3DNow! de AMD.

### **DirectX 6.1**

Fue lanzado a principios de febrero de 1999. La característica más grande era la adición del API de DirectMusic, que originalmente fue descartado de DirectX 6.0.

### **DirectX 7.0**

Fue lanzado en septiembre de 1999. DirectX 7 agregó un soporte mejorado para gráficos 3D y para sonido, además de mejorar notablemente el rendimiento.

Aceleración por hardware para DirectMusic en los adaptadores que soportan la tecnología.

También introdujo al API de Direct3D la capacidad de aprovechar transformación, iluminación y compresión de texturas por hardware.

Mejoras en la aceleración de sonidos por hardware y los algoritmos de sonido 3D mejorados usando tecnología EAX de Creative.

También incluyó soporte para Visual Basic en el API, permitiendo a programadores utilizar el API en un ambiente más familiar.

### **DirectX 7.0a**

Era una actualización de menor importancia a DirectX 7 que proporcionó las actualizaciones mejoradas del funcionamiento de dispositivos de retorno de fuerza (force feedback), para usuarios de Windows 95/98.

### **DirectX 8.0**

Cambio radical en la arquitectura de video, DirectDraw y Direct3D se fusionan, se introduce un modelo de "shaders" para manejar el procesado de vértices y de píxeles, con funciones fijas o con un lenguaje parecido al código ensamblador. Esto significa que los



programadores ya no solo tenían un API que les brindaba toda la capacidad del hardware, sino que ellos mismos podían de una manera básica, programar la tarjeta de video, para que produzca los efectos visuales que ellos querían lograr.

También hubo una mayor integración entre DirectSound y DirectMusic.

Se integro la habilidad de insertar filtros de procesamiento dentro del pipe-line de audio para lograr efectos especiales

- Reverb
- Chorus
- Etc.

Grandes cambios estructurales en DirectPlay, se adopto el modelo “Push” remplazando al “poll”, permitiendo el uso efectivo de las capacidades de multilectura, también se agrego soporte para conversaciones de voz.

Se introduce DirectShow Como parte de DirectX, DirectShow proporcía al programador un API unificado para manejar las capacidades de reproducción de video de los diversos tipos de hardware disponible en el mercado.

### **DirectX 9.0 (Actual)**

El principal cambio en esta ultima versión de DirectX es la introducción del HLSL (High Level Shading Language), con la introducción de este lenguaje basado en C, los programadores pueden crear desde animaciones ultra realistas, hasta efectos visuales sorprendentes, sin tener que preocuparse por el tipo específico de hardware que se este utilizando.

Se fueron agregando también, nuevas técnicas de procesamiento grafico, como son el Depth Adaptive Tessellation (reducción dinámica en la complejidad de objetos según distancia) y el displacement mapping (capacidad de dar profundidad a un objeto 3D de acuerdo a un mapa de colores).

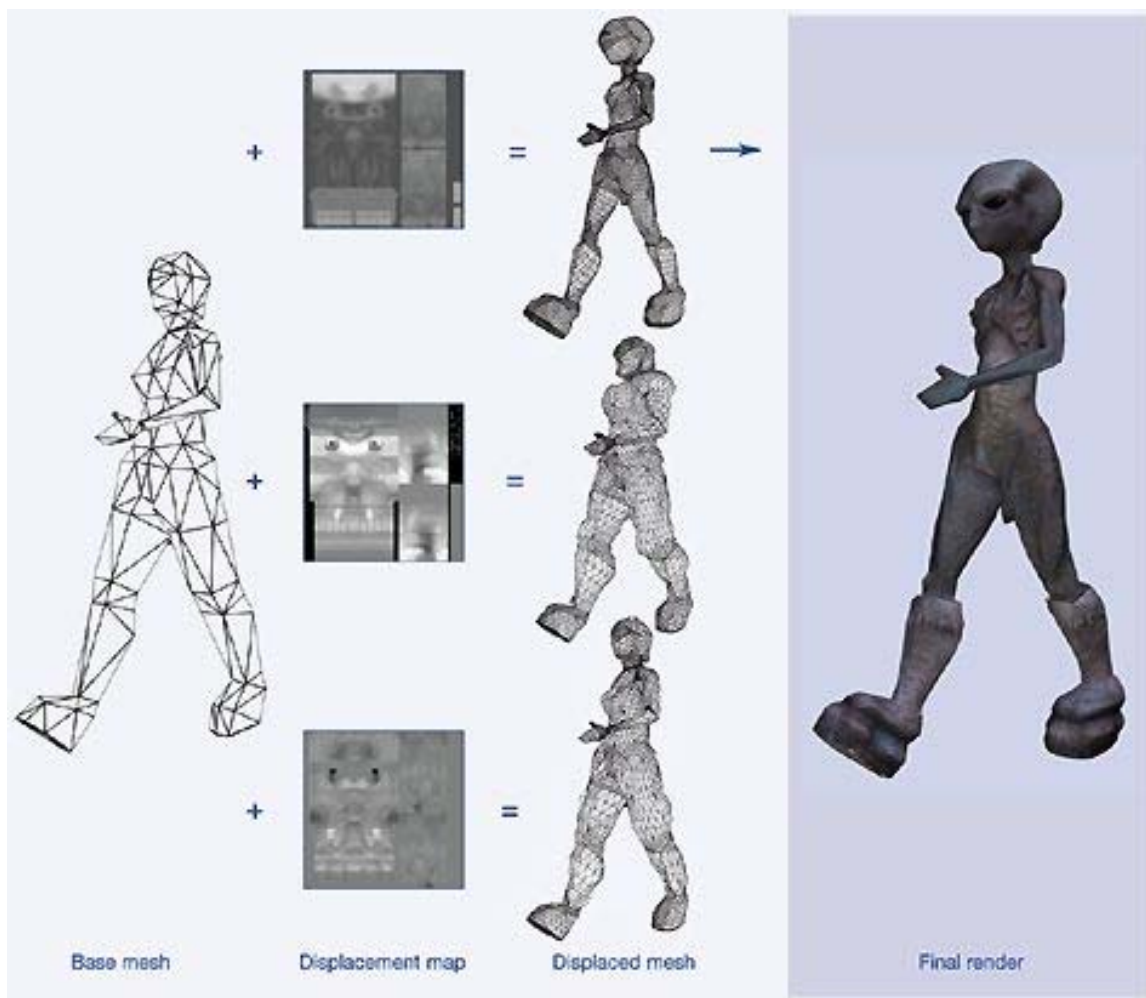
Ahora en verdad se puede hablar de una verdadera GPU, ya que con la introducción del HLSL que permite saltos, loops y manejo de subrutinas, se tiene un procesador grafico totalmente programable.

En esta versión, el api viene con unas utilidades nuevas facilitan la visualizacion y manipulación de texturas asi como unos plug-ins actualizados para exportar – importar modelos 3d de aplicaciones como 3D Studio o Maya.

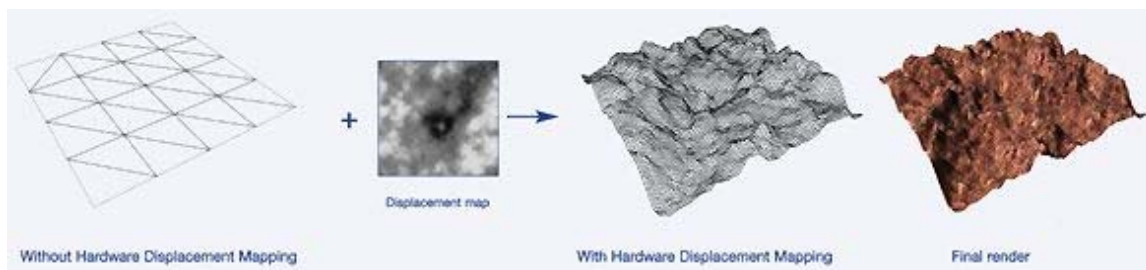
DirectPlay tuvo mejoras en cuestiones de seguridad, tambien incluye el soporte para Internet Protocol v.6.

DirectInput no sufrió cambios en el API, pero se trabajo para lograr mayor compatibilidad con el hardware actual.

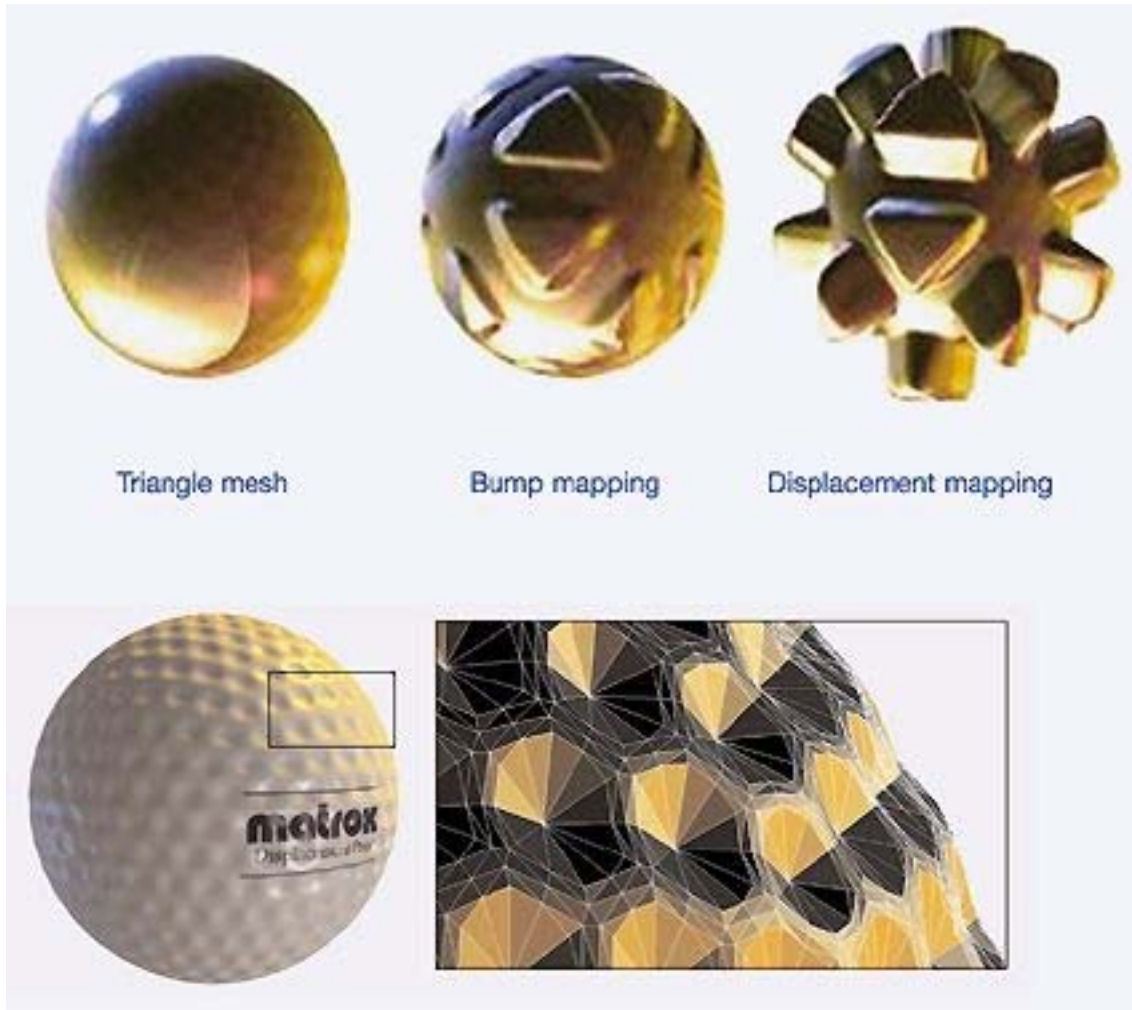
DirectSound tubo mejoras en el rendimiento.



Generación de diferentes modelos 3D basados en un modelo simple usando Displacement Mapping.

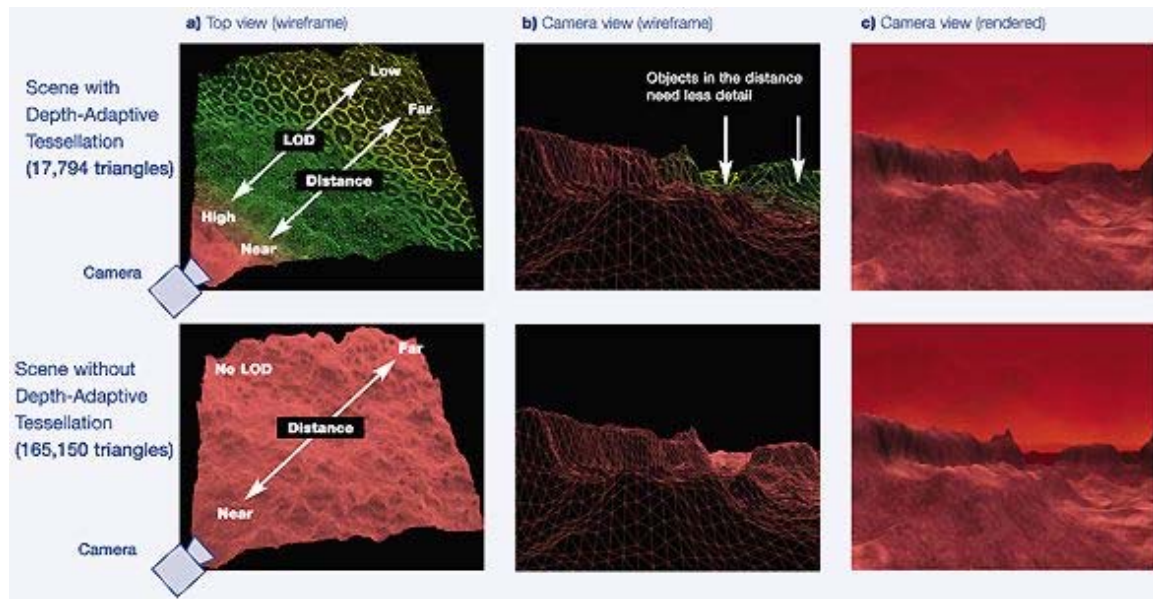


Generación de terreno 3D con Displacement Mapping.



Diferencias entre Bump Mapping y Displacement Mapping.

Notar que el Displacement Mapping crea geometría 3D real sobre el modelo base, mientras que el Bump Mapping utiliza efectos de sombreado para crear el efecto de geometría 3D.



Reduccion de Poligonos sin perdida de calidad con Depth Adaptive Tessellation.

## El Futuro - DirectX/LH

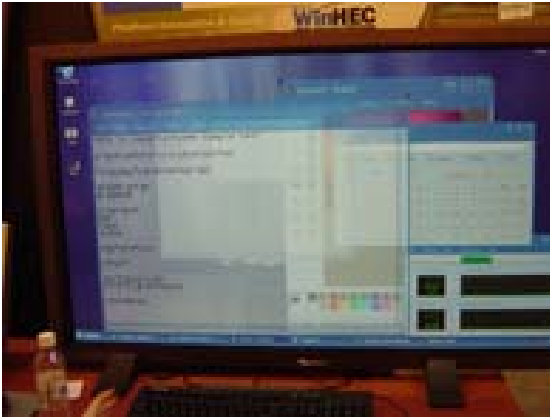
No hay mucha información sobre lo que será el futuro de DirectX, lo que se sabe es que lo que se está desarrollando tiene como nombre clave DirectX/LH ya que el próximo sistema operativo de Microsoft lleva como nombre clave Windows Longhorn.

Con su nueva interfase de usuario, de nombre clave AERO, Windows Longhorn permitirá aplicar todo tipo de animaciones, efectos visuales y animaciones a las diferentes ventanas del escritorio, sin sobrecargar a la cpu, el sistema de video utiliza buffers individuales para cada ventana del escritorio y puede aplicar efectos visuales a cada una de estas separadamente.

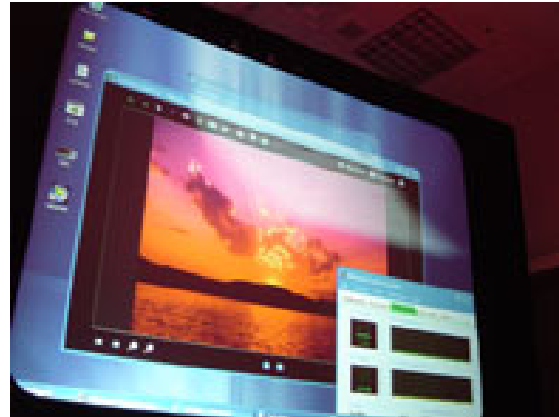
Efectos como animación y efectos de ventanas, escalamiento de ventanas (no solo el tamaño del borde), ventanas transparentes, fuentes de video animadas y muchas otras serán cosas comunes en el futuro cercano de las posibilidades de DirectX.

Otras funcionalidades que están previstas para DirectX/LH, al menos en la parte de 3D son:

- Vpu (video procesing unit) totalmente programable.
- Sombras dinámicas en tiempo real.
- Unificación en los programas de vértices y píxeles.



El escalamiento de ventanas de altos Dpi será algo común en un futuro cercano.



Se puede notar como una ventana tiene un efecto de deformación al ser arrastrada y la cpu no es cargada.



Aquí se pueden ver múltiples ventanas de video siendo animadas independientemente sin perder interactividad.

## Bibliografía

- <http://www.tomshardware.com>
- Evolution of DirectX and the PC Games Development Scene  
Tony Cox  
Microsoft Corporation  
[tonycox@microsoft.com](mailto:tonycox@microsoft.com)
- <http://support.microsoft.com/support>  
Artículos: **Q150651, Q155002, Q158961.**
- <http://msdn.com>