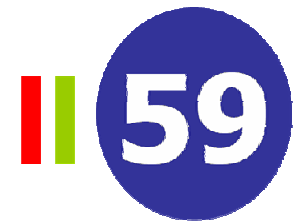




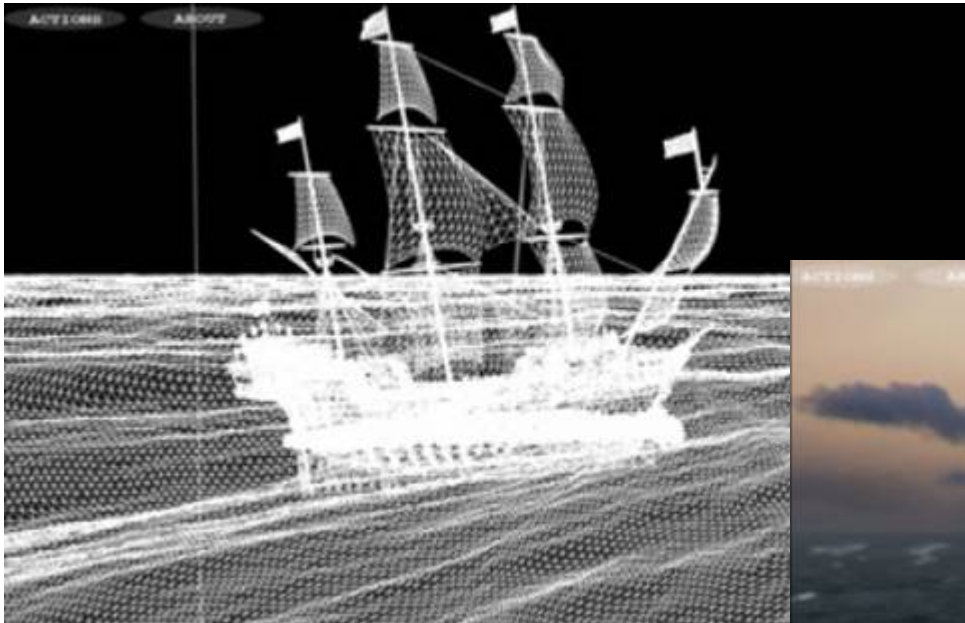
Aplicaciones
Evolución
Programación gráfica
Juegos por ordenador

Introducción



- **Visualización en tiempo real**
 - “Creación de imágenes sintéticas con la suficiente rapidez como para que el usuario pueda interactuar con un entorno virtual”.
Tomas Möller.
- **Características**
 - Gráficos interactivos
 - Desde 15 a 72 fps
- **Campo de rápida evolución**
 - Incremento de las prestaciones del Hardware gráfico
 - Disminución de su precio
 - Algunas librerías gráficas nacen, se desarrollan y con frecuencia desaparecen

Introducción



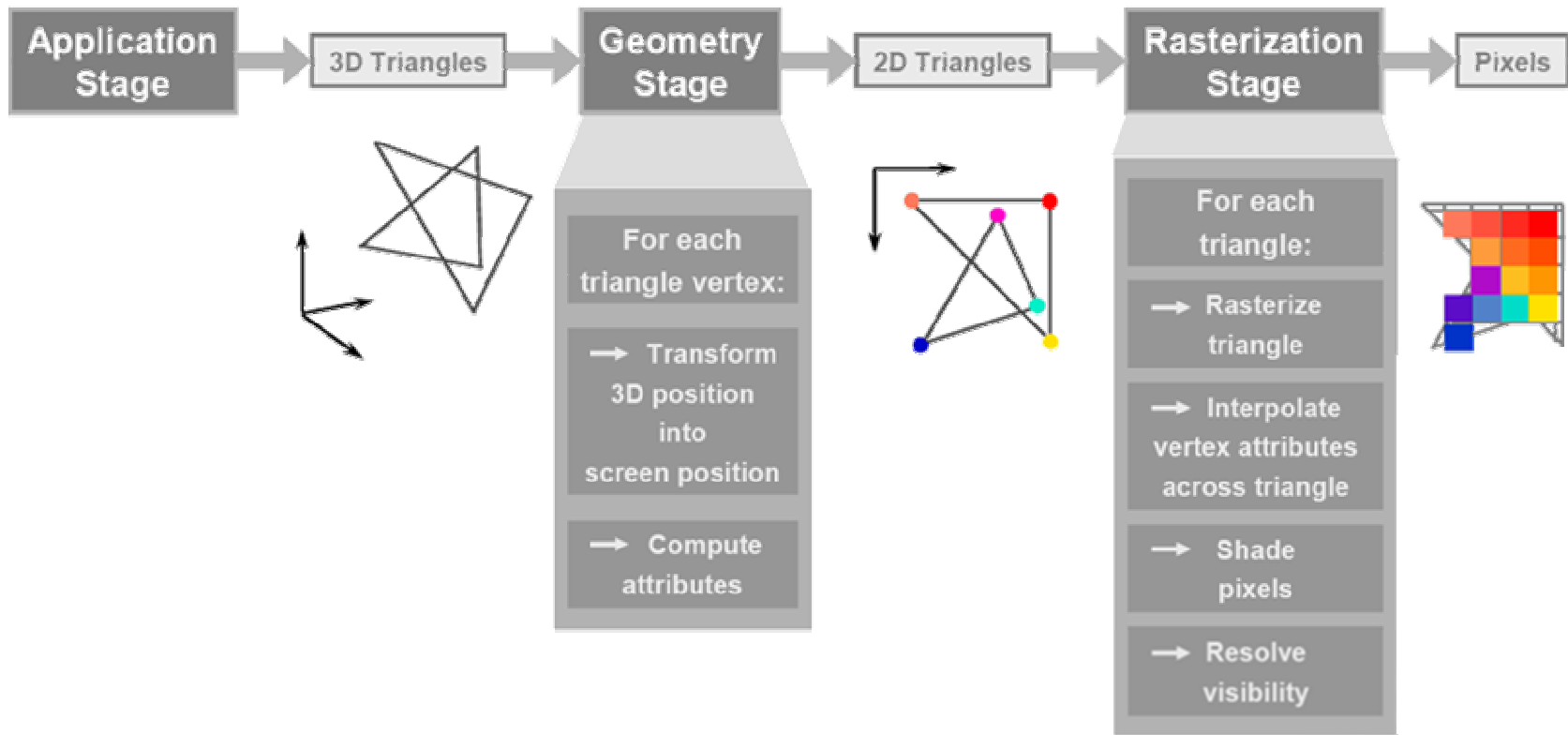
Escena 3D = colección de primitivas 3D (triángulos, líneas y puntos)



Imagen = matriz de píxeles

Introducción

- El sistema gráfico



Aplicaciones

- Realidad Virtual



Aplicaciones

- Simuladores y entornos de trabajo



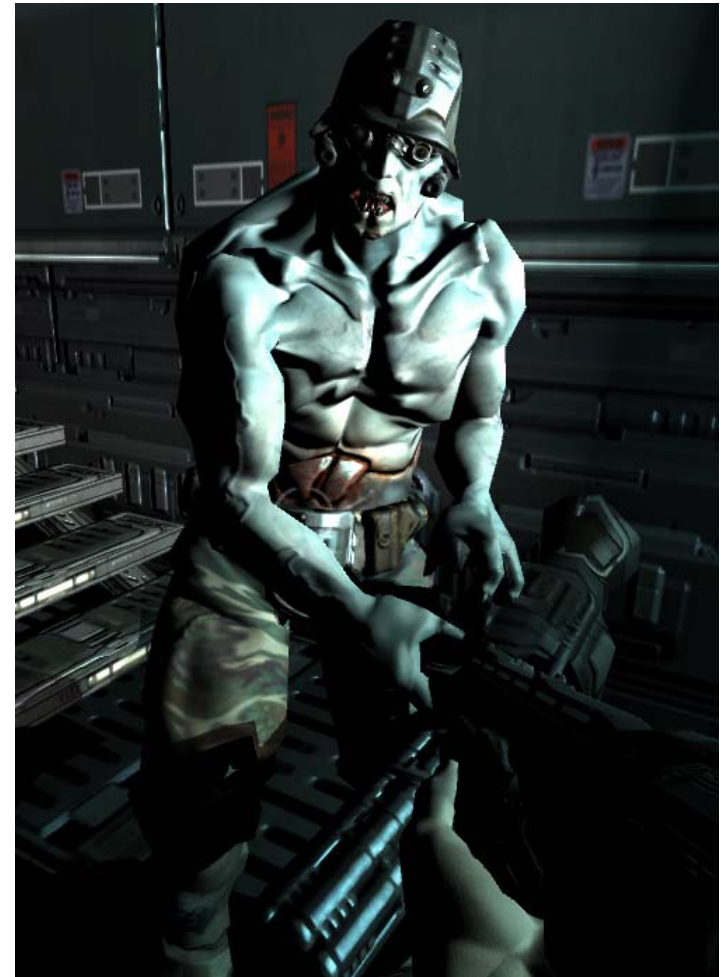
Aplicaciones

- Televisión (escenarios virtuales)



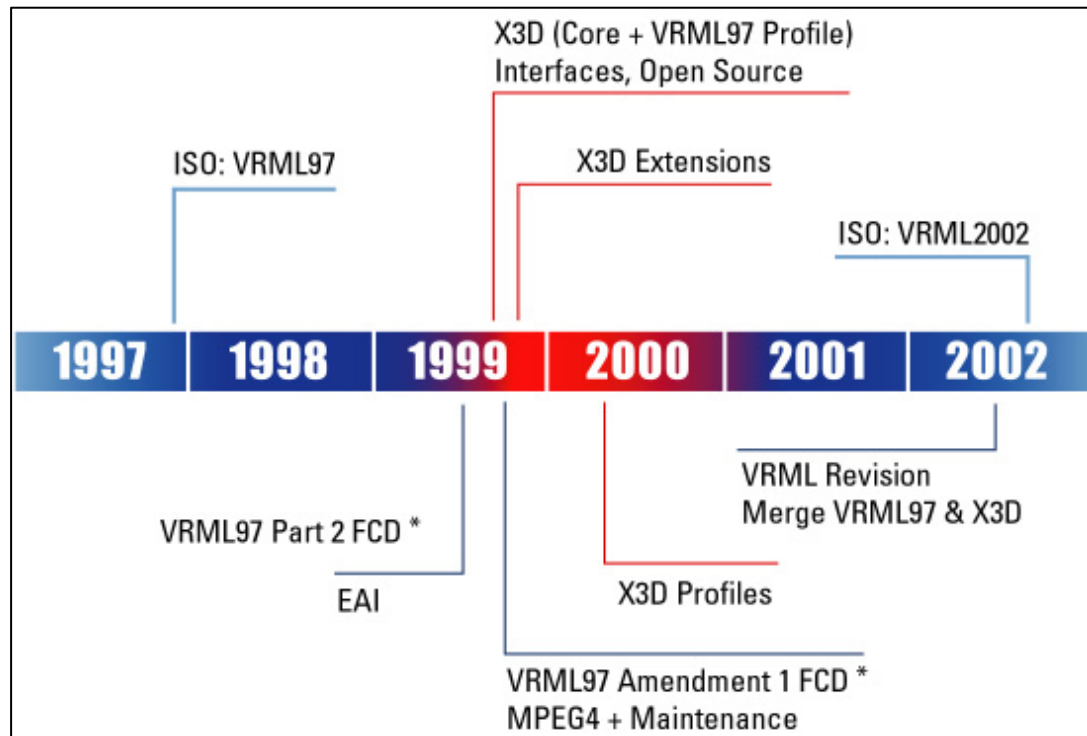
Aplicaciones

- Juegos por ordenador



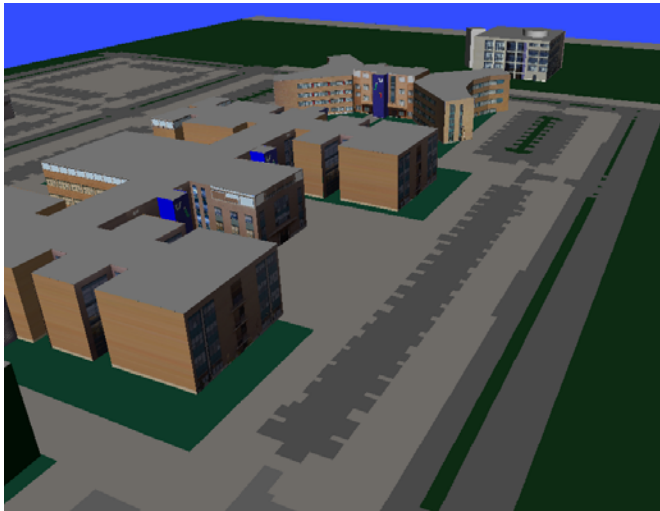
Aplicaciones

- Gráficos en Internet
 - <http://www.web3d.org>
 - <http://web3d.about.com/>



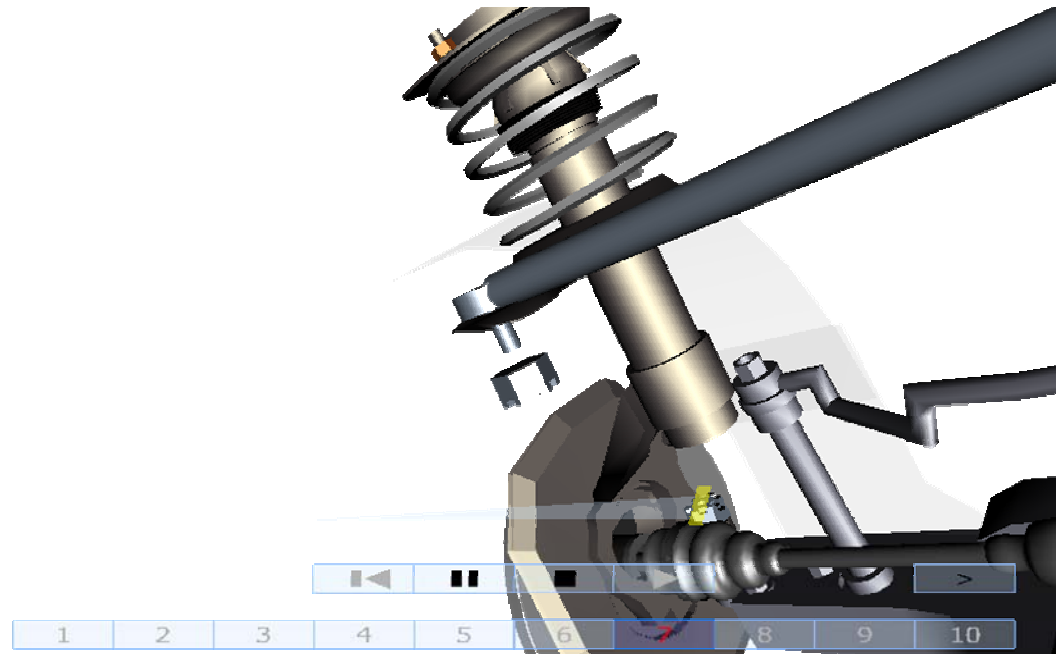
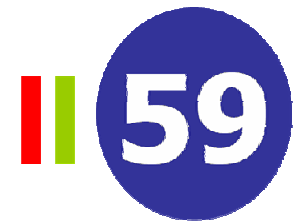
Aplicaciones

- Entornos multiusuario
 - Blaxxun
 - Active Worlds



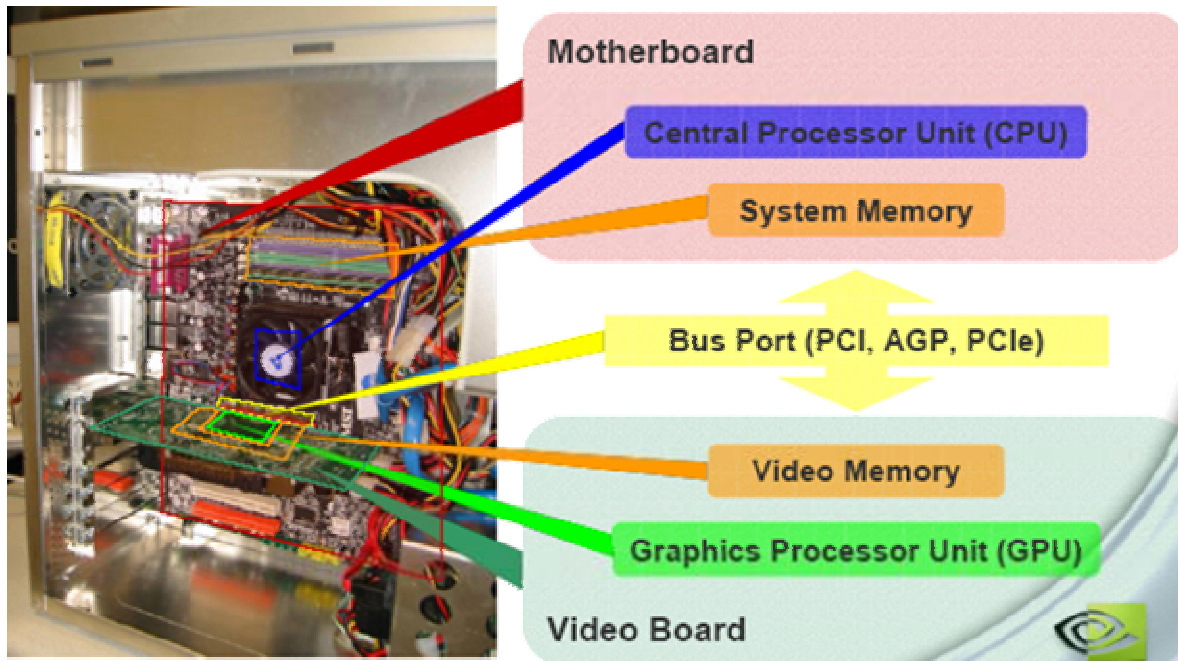
Aplicaciones

- Presentación de productos



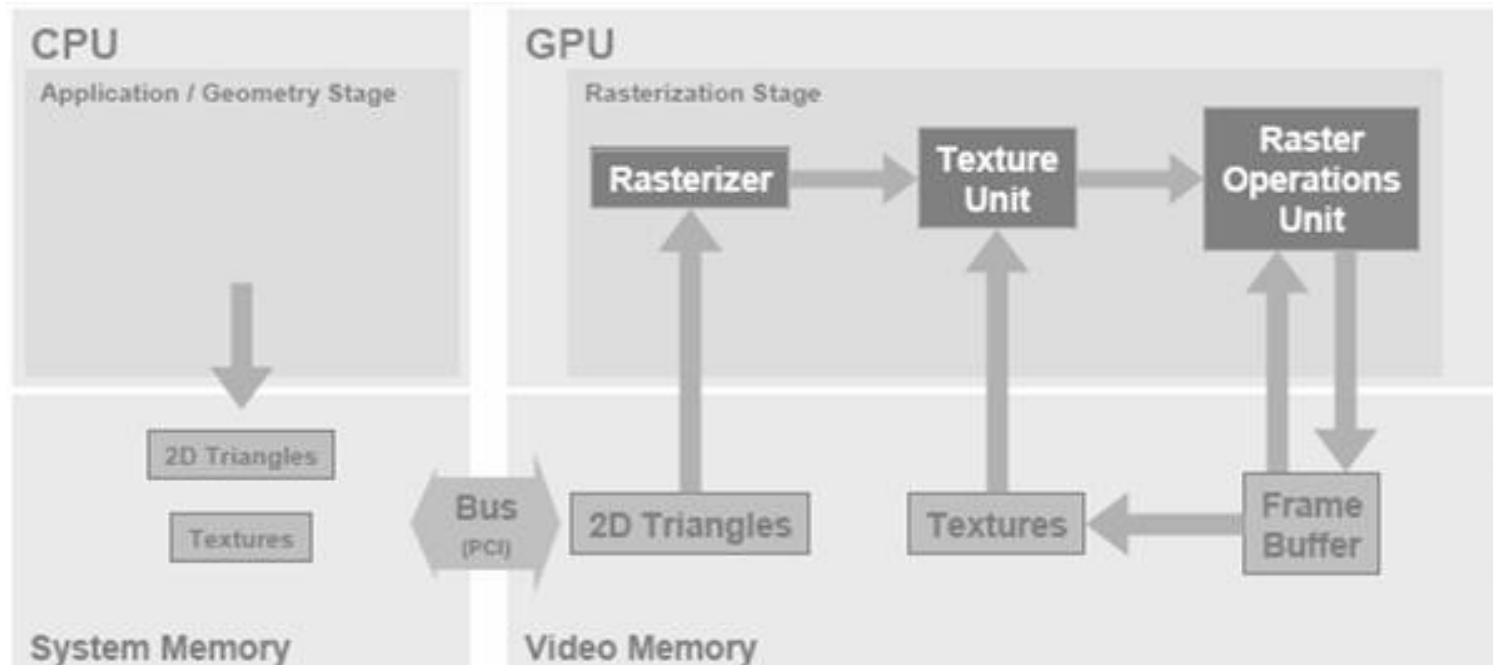
Evolución

- Evolución del hardware gráfico
 - 1995. Origen de la actual arquitectura PC



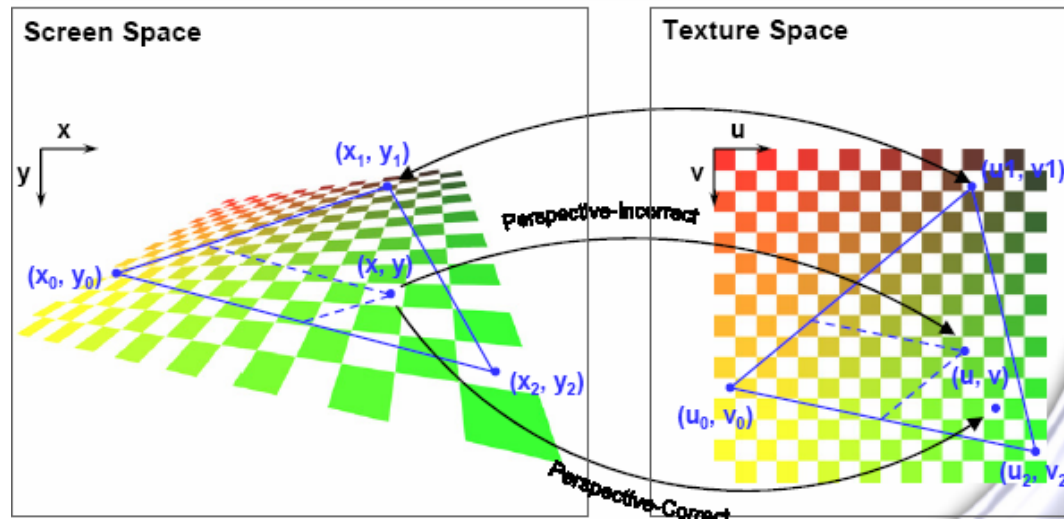
Evolución

- 1995-1998. Texturas y Z-Buffer
 - 3dfx Voodoo. Primer procesador gráfico para PC (triángulos 2D)
 - PCI: Peripheral Component Interconnect



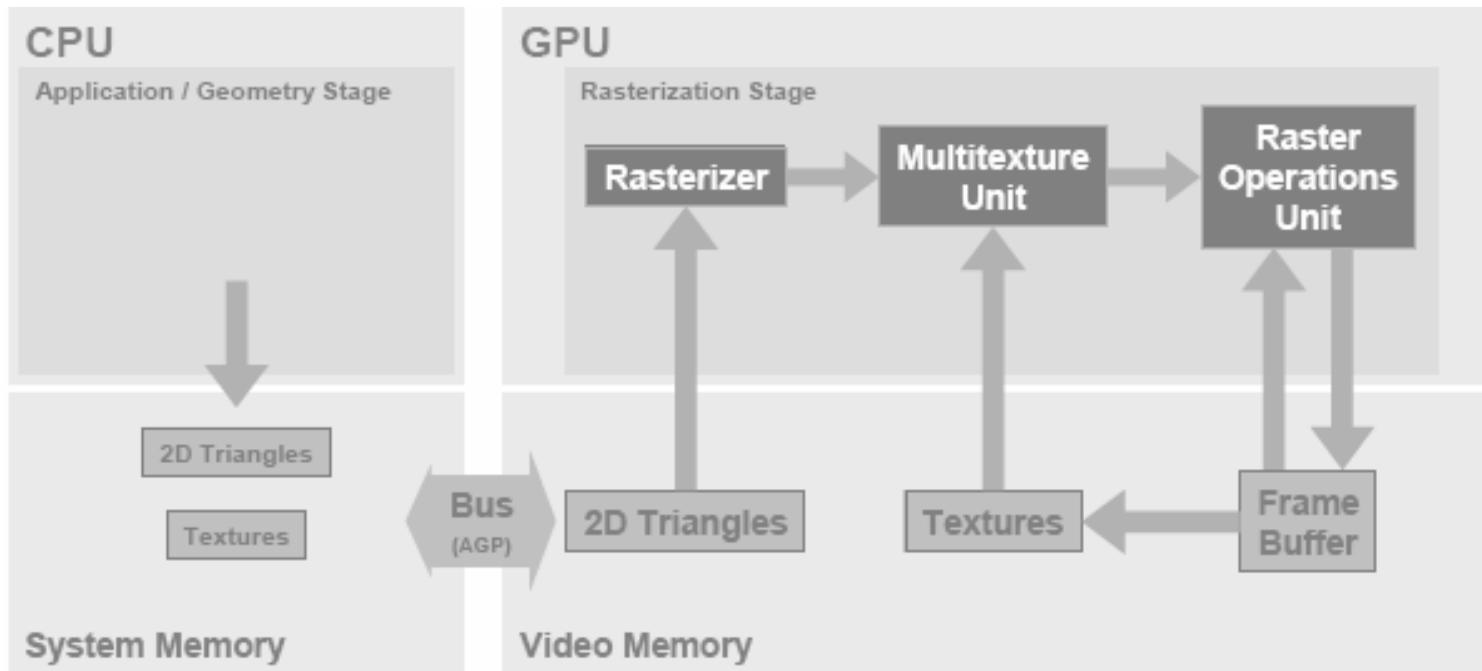
Evolución

- Aplicación de texturas con corrección perspectiva



Evolución

- 1998. Texturas múltiples
 - TNT (NVIDIA), Rage (ATI)
 - AGP: Accelerated Graphics Port



Evolución

- Ancho de banda. AGP= 2 x PCI
- Un píxel puede colorearse utilizando más de una textura



X

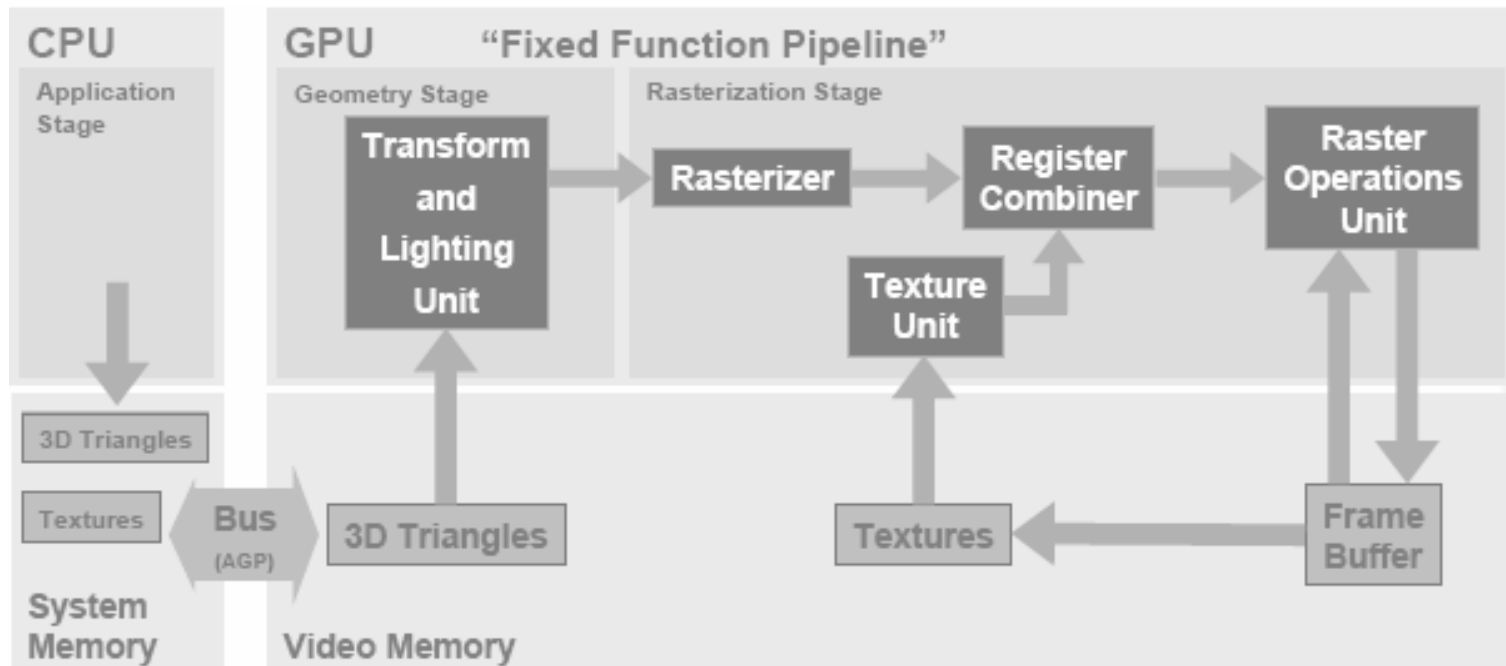


=



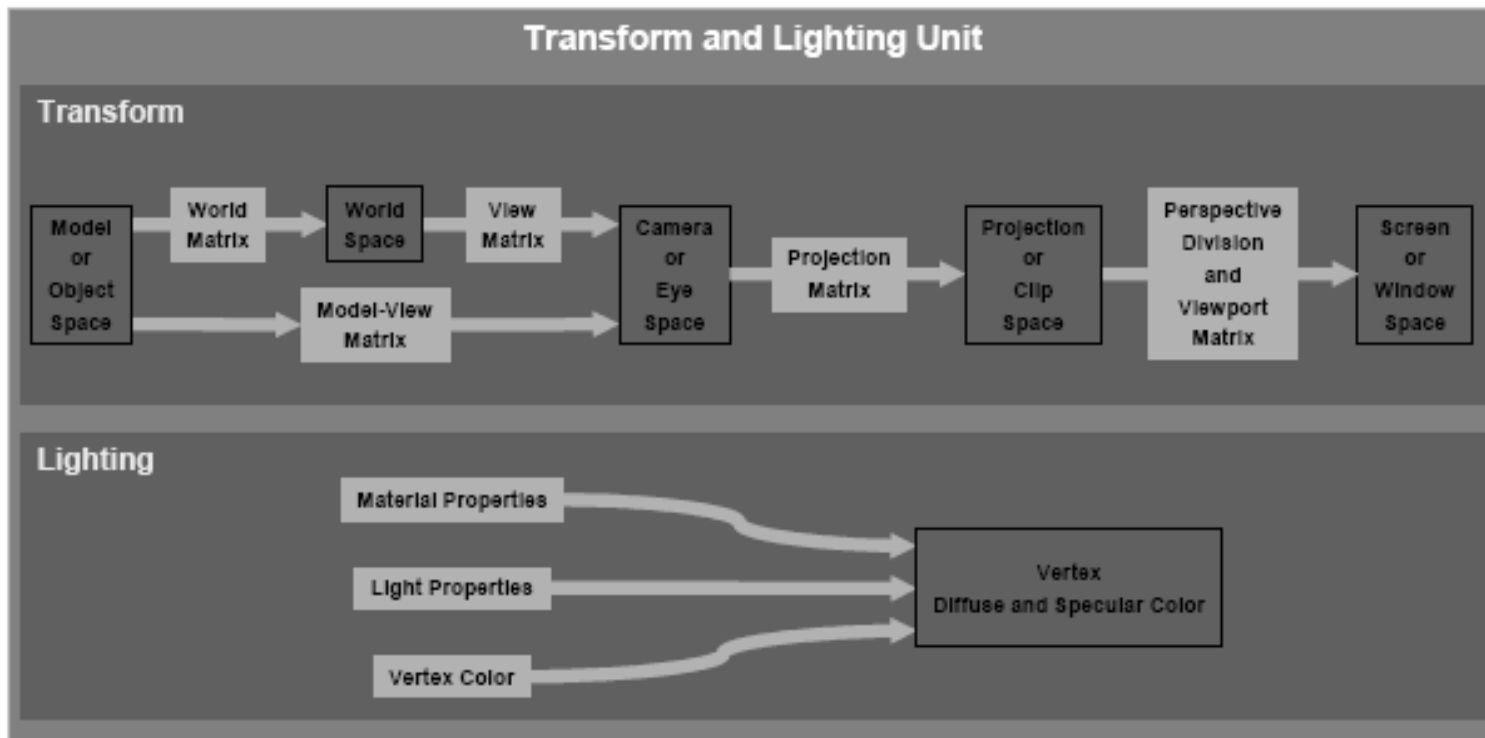
Evolución

- 1999-2000. Transformaciones e iluminación
 - GeForce 256, GeForce2 (NVIDIA), Radeon 7500 (ATI), Savage3D (S3)
 - Fase de geometría en la GPU (unidad TnL)



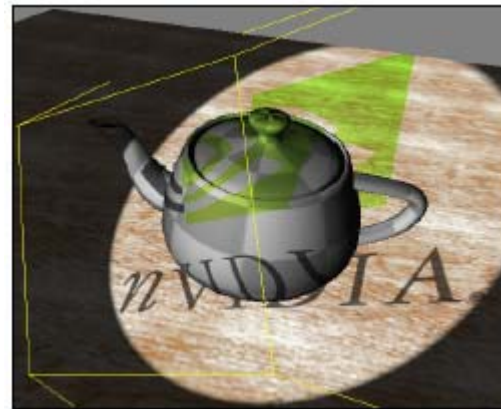
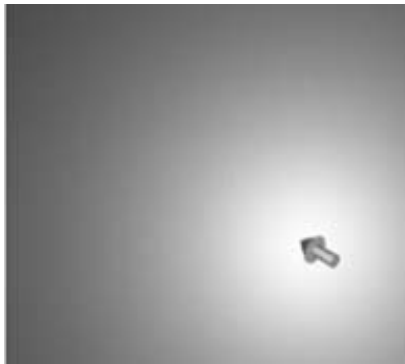
Evolución

- La GPU maneja los triángulos y toda la información para su iluminación



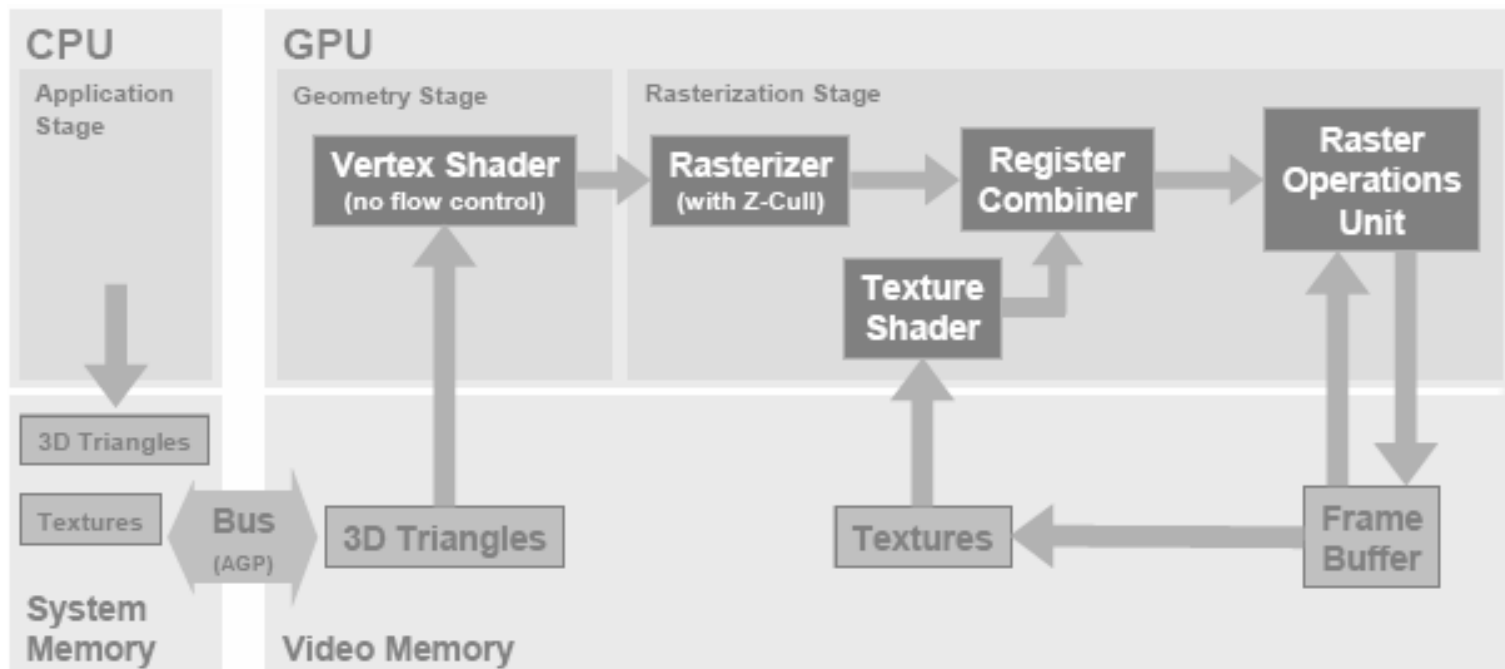
Evolución

- Más posibilidades a nivel de píxel:
 - Auténtico *Bump Mapping* con mapas de normales (sin interpolación)
 - Nuevos formatos de textura para *Environment Mapping* y *Shadow Mapping*



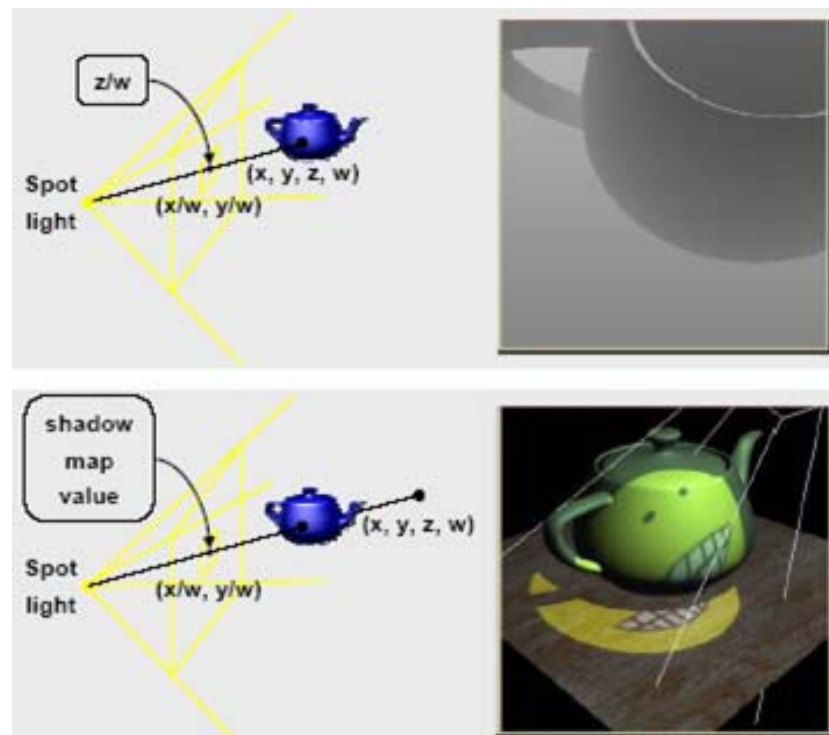
Evolución

- 2001. GPU programable
 - GeForce3, GeForce4 (NVIDIA), Radeon 8500 (ATI)
 - Programas de vértices (vertex shaders) sin control del flujo
 - Z-Cull: predice que fragmentos fallaran en el Z-buffer y los elimina
 - Texture Shader: más posibilidades y operaciones con texturas



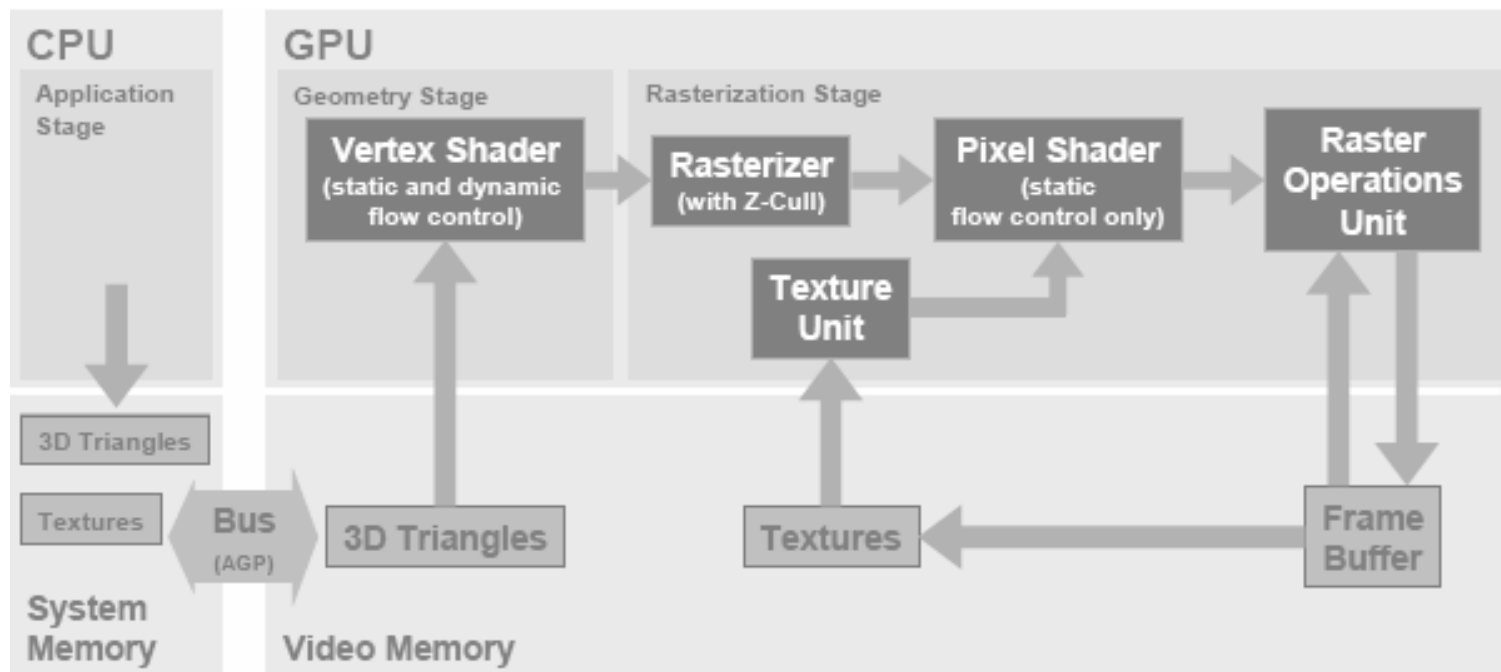
Evolución

- Shadow Mapping
 - Mapa de sombreado. Contiene el valor de profundidad de los puntos visibles desde la fuente de luz
 - Algoritmo de visualización. Un punto 3D está en sombra si su valor de z transformado es menor que el del mapa de sombreado



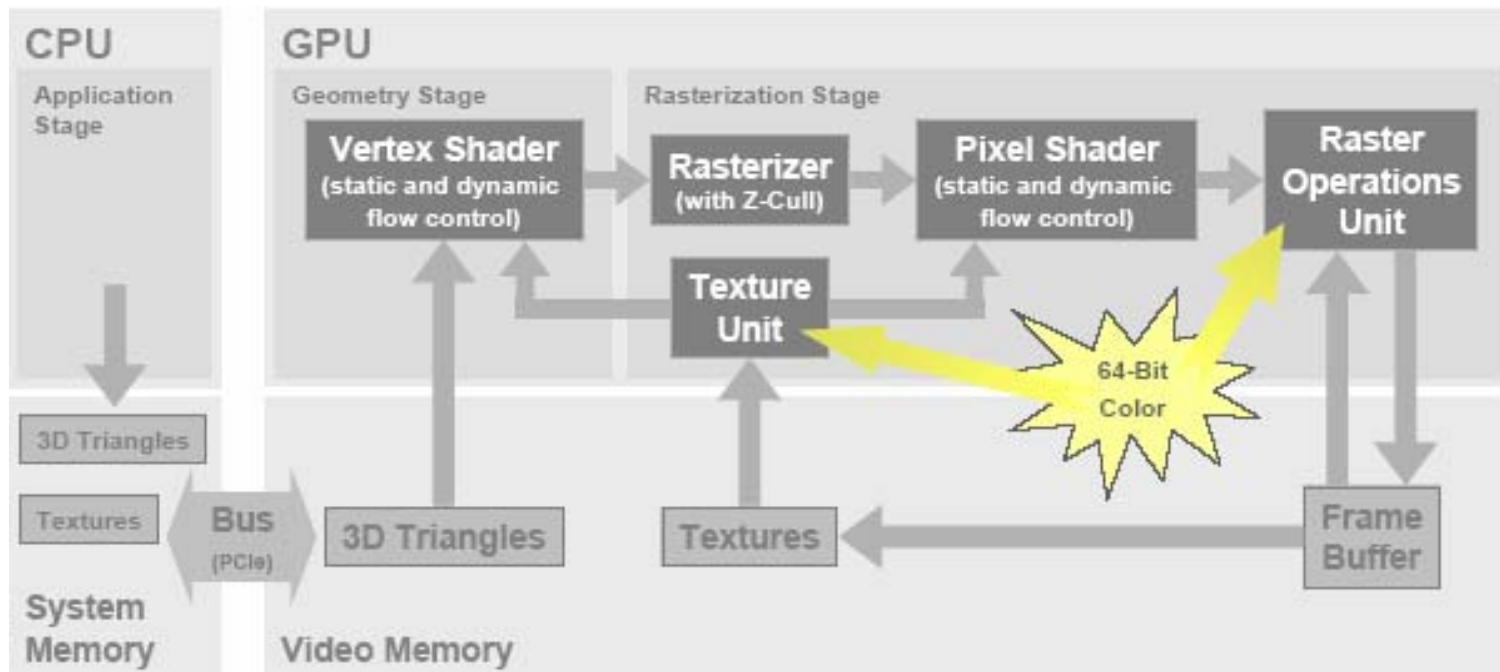
Evolución

- 2002-2003. Programas para píxeles
 - GeForce FX (NVIDIA), Radeon 9600 a 9800 (ATI)
 - Control de flujo para programas de vértices
 - Control de flujo estático para programas de píxeles (igual para todos)
 - MRT: Multiple Render Target



Evolución

- 2004. Shader Model 3.0 y soporte para color de 64-Bits
 - GeForce 6800 (NVIDIA)
 - Unificación del modelo de programación de la GPU (control de flujo total)
 - Acceso a las texturas desde los programas de vértices



Evolución

- Imágenes de alto rango dinámico
 - Color de 64-Bits. 16-Bits por componente RGBA
 - El rango dinámico de una imagen es la diferencia entre la mayor y menor luminancia

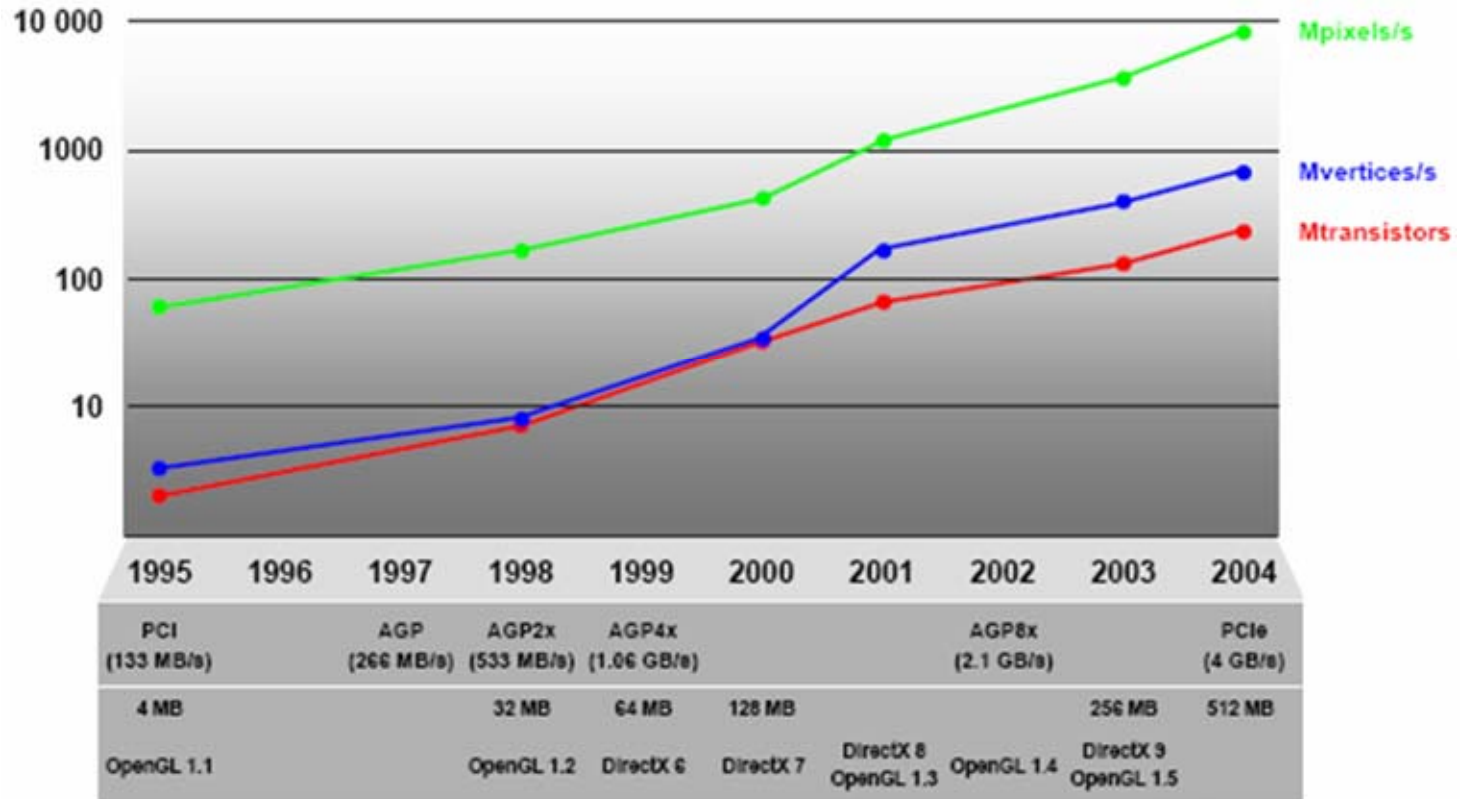


- PCIe: Peripheral Component Interconnect Express
 - Ancho de banda. PCIe = 2 x (AGP x 8)
 - Ancho de banda en ambas direcciones (CPU <-> GPU)

Evolución



– Resumen 1995-2004



Evolución

- Ejemplos de la evolución



Virtua Fighter
(SEGA Corporation)

NV1

50K triangles/sec
1M pixel ops/sec
1M transistors

1995



Dead or Alive 3
(Tecmo Corporation)

Xbox (NV2A)

100M triangles/sec
1G pixel ops/sec
20M transistors

2001



Dawn
(NVIDIA Corporation)

GeForce FX (NV30)

200M triangles/sec
2G pixel ops/sec
120M transistors

2003

Evolución

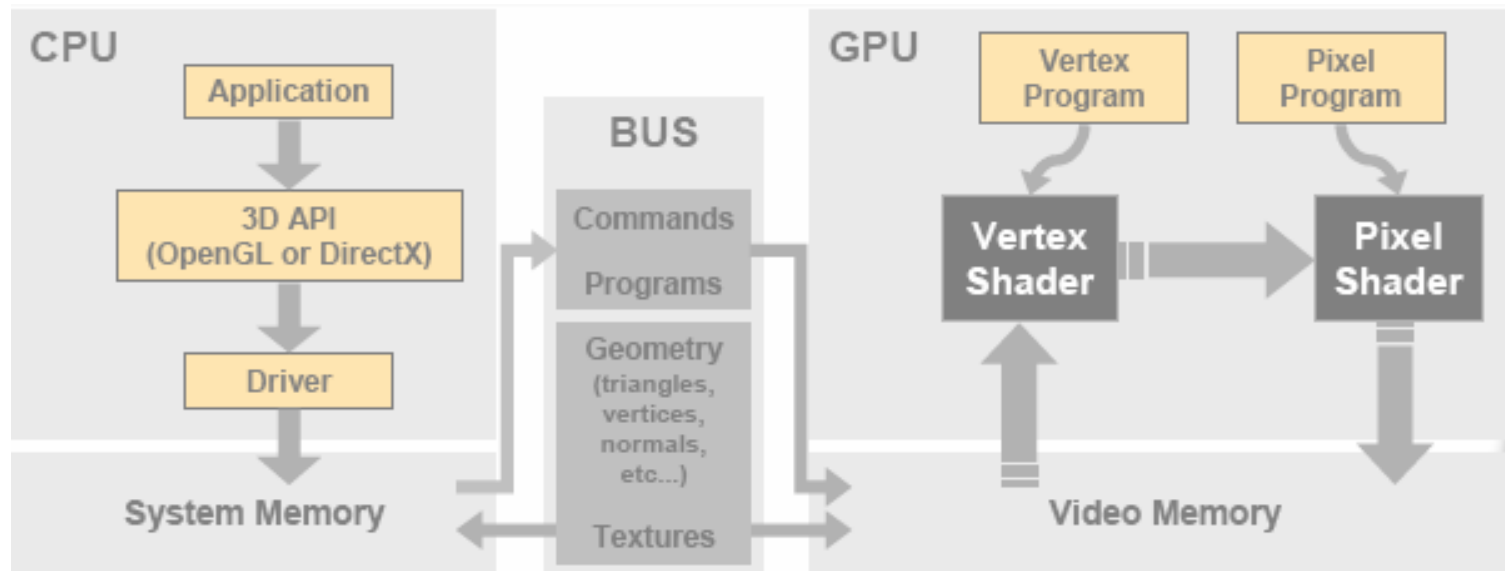
- El futuro.
 - Modelo de programación de la GPU más unificado
 - Más potencia de cálculo
 - Las últimas GPU disponen de 6 unidades para procesar vértices y 16 para procesar píxeles
 - Más memoria de vídeo
 - En la actualidad 512MB
 - Más ancho de banda entre la CPU y la GPU
 - El bus PCIe funciona actualmente a 4GB/s



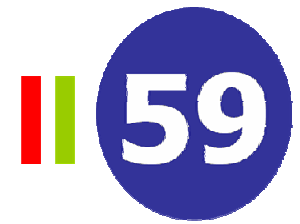
Programación gráfica

- **Arquitectura del sistema**

- La aplicación, el API 3D y el driver están codificados en C o C++
- Los programas de vértices y píxeles estas escritos en lenguajes de alto nivel (Cg, DirectX HLSL, OpenGL Shading Language)



Programación gráfica



- 3D APIs
 - DirectX (Microsoft Corporation)
 - Basada en C++
 - Una nueva versión cada año más o menos (DirectX 9.0)
 - Sólo para Windows
 - Popular en la industria de los videojuegos
 - OpenGL (OpenGL ARB - 3DLabs, Apple, ATI, Dell, IBM, Nvidia, SGI, Sun)
 - Basada en C
 - Evoluciona por medio de extensiones (que finalmente pueden adoptarse como parte de la librería)
 - Multiplataforma
 - Popular en el ámbito académico y en todas las industrias que no son de videojuegos

Programación gráfica

- Esqueleto de una aplicación

Inicialización

Para cada Frame:

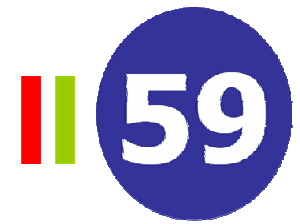
Dibujar sobre el Buffer trasero

Intercambiar Buffers

- Inicialización

- API: creación de la ventana y del contexto de dibujo (como se interactúa con el hardware gráfico). Dependencia del hardware. Existen librerías para facilitar la tarea como GLUT o AUX.
- Recursos
 - Zonas de dibujo (Render Targets). Memoria de video (RTT)
 - Programas de Vértices y Píxeles (Shaders)
 - Texturas
 - Vectores de Índices y Vértices (Buffers)

Programación gráfica



- Bucle de dibujo
 - Para cada frame , cada malla se dibuja así:

Para cada pasada de dibujo:

Activar buffer de vértices

Activar buffer de índices

Activar programa de vértices y parámetros

Activar programa de píxeles y parámetros

Activar estados de dibujo

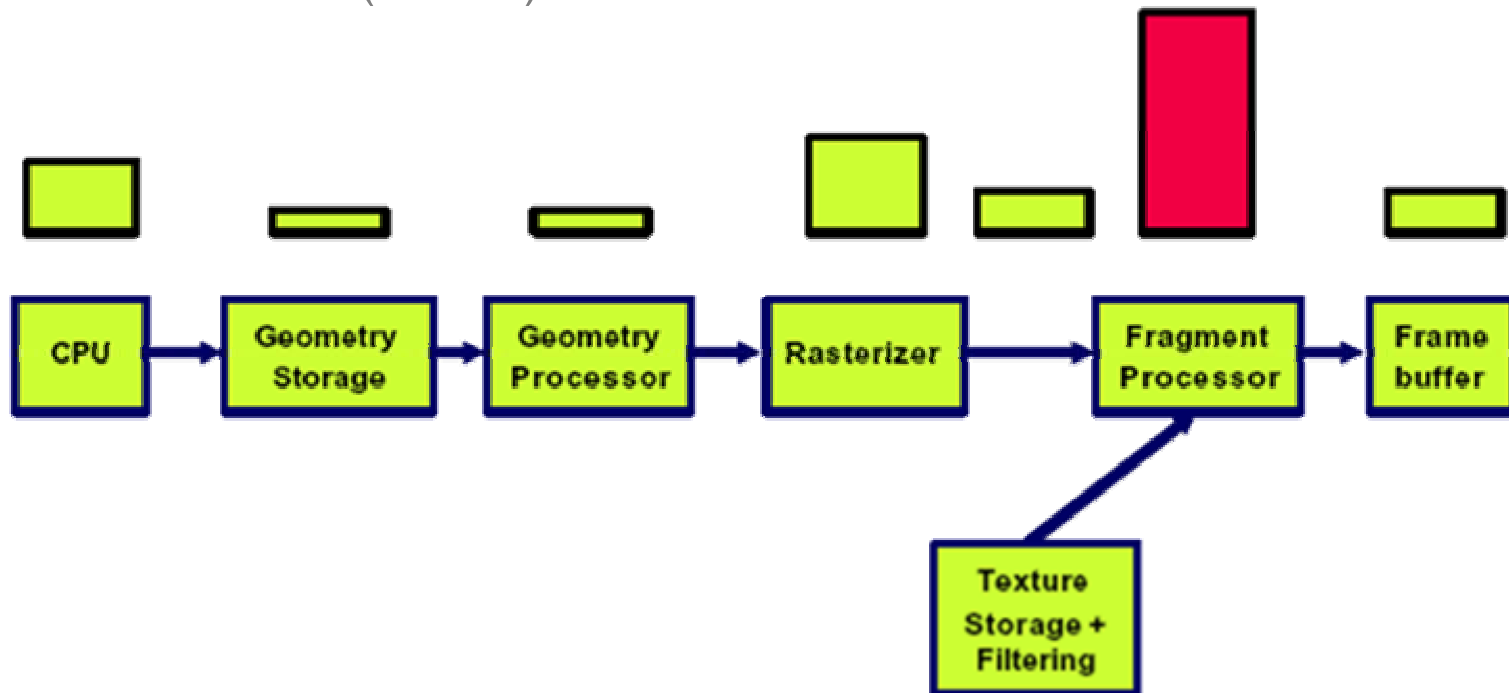
Activar destinos de dibujo (Render target)

Dibujar

- En ocasiones es necesario realizar varias pasadas de dibujo para obtener el efecto deseado

Programación gráfica

- Balancear la tubería gráfica
 - Rendimiento óptimo
 - NVPerfHUD (nVidia)



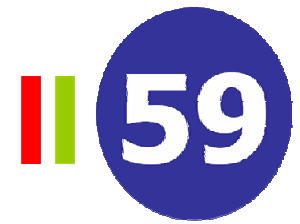
Juegos por ordenador

- ¿Qué es un motor de juegos?
 - El núcleo de una aplicación gráfica en tiempo real
- Elementos involucrados
 - Sonido
 - Física
 - Inteligencia Artificial
 - Comunicaciones
- Bucle básico de ejecución



```
Mientras no acabe el juego
  leer entrada de datos
  procesar entrada de datos
  producir sonidos
  simular física e I.A.
  redibujar
```

Juegos por ordenador



- **Sonido**
 - Sonido localizado para incrementar la inmersión
 - Soluciones
 - Variar el volumen con los cambios en posición
 - Modelo matemático más complejo. Head-Related Transfer Function (HRTF) para audio 3D con 2 o 4 altavoces
 - Los juegos utilizan:
 - Dolby 5.1: necesita muchos altavoces
 - Creative's EAX: "environmental audio"
 - Aureal's A3D: probablemente el mejor audio 3D
 - DirectSound3D: la solución de Microsoft
 - OpenAL: API multiplataforma de código

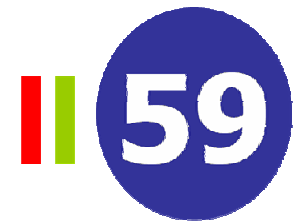
Juegos por ordenador

- Física

- La simulación física real tiene un alto coste computacional. Se suelen utilizar muchas simplificaciones (cajas de inclusión)
- Muchos elementos de un juego (explosiones, proyectiles) no están simulados sino descritos.
- Un buen motor de físicas es un requerimiento actualmente
 - Unreal (HAVOK), Doom III (propio)
- Se utilizan estructuras de datos espaciales para acelerar el cálculo de colisiones
- Se utilizan sistemas de partículas

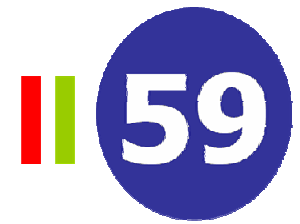


Juegos por ordenador



- Inteligencia Artificial
 - Soluciones
 - máquinas de estados finitos (FSMs)
 - Utilización de scripting, algoritmos de búsqueda de caminos A*
 - Crear comportamientos complejos y creíbles es difícil
 - Una solución inicial es trabajar con librerías
 - La carga esta en la CPU

Juegos por ordenador



- Comunicaciones

- Internet es impredecible muchos paquetes pueden perderse
- Utilización de “multithreading” y entrada salida sin bloqueos
- Se suele utilizar el protocolo UDP (con gestión de errores propia). TCP/IP es demasiado lento (demasiada corrección de errores, espera de confirmación)
- Se suele utilizar el modelo cliente servidor
 - QuakeWorld (1996) John Carmack
 - El servidor simula el juego a X Hz
 - Los clientes juegan a diferentes velocidades (normalmente $> X$)
 - Problema: El cliente se actualiza a diferentes velocidad que el servidor
 - Solución: Interpolación

Juegos por ordenador



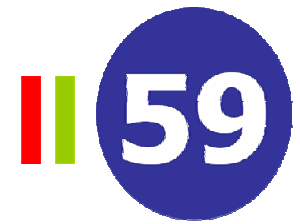
- **Creación de contenidos**
 - Un buen motor no hace que un juego sea bueno
 - Lo más importante son los contenidos creativos:
 - Personajes, entornos, sonido, música, animaciones, etc.
 - EA utiliza motores comerciales en sus juegos
- **Algunas de las herramientas más utilizadas en el sector**
 - 3DStudio Max, Maya y Blender para el modelado
 - RenderMonkey (ATI) y FXComposer para el desarrollo de shaders
 - PhotoShop para la creación de mapas de normales, bump mapping

Juegos por ordenador

- Modelado con pocos polígonos
 - Dificultad en manejar modelos complejos
 - La utilización de mapas de normales permite tener modelos con pocos polígonos que parecen muy complejos
 - Dos modelos uno complejo y otro simplificado. Generación automática de mapas de normales

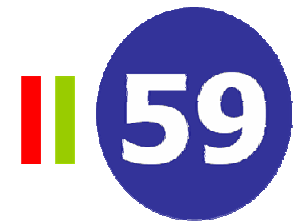


Juegos por ordenador



- Animación por cuadros claves
 - Similar a la utilización de cuadros claves en animación tradicional
 - Un cuadro clave representa la posición y orientación de un punto en el espacio
 - El modelo se divide en partes (cabeza, torso, etc.) cada parte tiene asociados vectores/matrices
 - Los cuadros claves especifican los valores de esas matrices en instantes de tiempo y la geometría asociada
 - La animación se produce por interpolación de los cuadros claves
 - Grandes necesidades de memoria porque es necesario almacenar la geometría en cada cuadro clave

Juegos por ordenador



- Animación con esqueletos
 - El modelo tiene un esqueleto que es el que se anima
 - Cada vértice sobre la “piel” esta conectado con un hueso. Utilización de jerarquía y matrices de transformación
 - Esta transformación de vértices se puede hacer en un vertex shader.
 - Los esqueletos permiten simulaciones realistas
 - Menos coste de almacenamiento
 - Ejemplos: Doom III, Half-Life 2
- Captura del movimiento
 - Muy importante en la industria de los juegos (son el mayor mercado)

Juegos por ordenador

- Colecciones de materiales
 - Cada vez es más importante la creación de librerías de materiales
 - Existe gente dedicada sólo a la creación de shaders (p.e. en Pixar)



Juegos por ordenador

- Ejemplos

Doom III



Halo 2

Juegos por ordenador

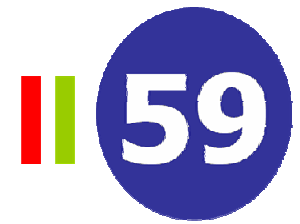
- Ejemplos

Black and White 2



Star Wars

Juegos por ordenador



- Salidas profesionales
 - **Fácil**. Inteligencia artificial, comunicaciones, grafismo, herramientas
 - **Difícil**. En gráficos ☹️
 - Otras salidas
 - Interfaces de usuario, visualización y educación
 - Internet (¿el futuro?)