

# OpenGL

Reynaldo Martell

5 de agosto de 2019

# ¿Que es OpenGL?.

OpenGL es un API que nos proporciona un gran conjunto de funciones con las que podemos usar para manipular gráficos e imágenes.

## En el principio ...

- Los datos estaban representados solo por impulsos eléctricos que son invisibles al ojo.
- Un método de visualización de estos datos tuvo que ser inventado, su nombre se llamó tarjetas perforadas.
- Se necesitaba que su interpretación fuera legible.
- La salida de las maquinas estaba lejos de ser óptima.

# Las primeras interacciones.

- En un inicio CRTs fue utilizado para mostrar el texto del estado de la computadora.
- En 1961 Ivan Sutherland desarrollo un programa llamado Sketchpad.
- Sketchpad permitía a los usuarios dibujar formas geométricas con una pluma de luz en tiempo real.
  - Definió los gráficos por computadora.
  - Introdujo las interfaces de usuario.
  - Sentó las bases de la programación orientada a objetos.



# Las primeras interacciones.

- En 1968 Ivan Sutherland y Bob Sproull diseñaron "La espada de Damocles".
- Precursor de lo que ahora llamamos Realidad Virtual.



# La primera década de OpenGL.

- Silicon Graphics (SGI) empresa fundada en 1981 especializada en gráficos 3D.
- Construyo software y hardware específicamente para este propósito.
- Desarrollo una biblioteca de software llamada IRIS GL (Integrated Raster Imaging System Graphical Library).
- IRIS GL fue una biblioteca muy popular durante la década de 1990 por su alto desempeño.
- IRIS GL tenía un problema por que se trataba de un sistema propietario que funcionaba con plataformas propias de SGI.

# La primera década de OpenGL.

- SGI libera IRIS GL de todas las funciones que no sean gráficos por computadora y lanza al público en 1992 OpenGL.
- Los proveedores de software tenían que proporcionar sus propias implementaciones del estándar en sus programas y hardware.
- Permitía comunicarse con el hardware de gráficos llamados “device drivers”.

# Flexibilidad.

- OpenGL proporciona una especificación de como debe funcionar el API.
- La mayor ventaja es el soporte para sus extensiones.
- Las extensiones son especificaciones que no son soportadas por OpenGL, pero el proveedor de software y hardware añaden estas funcionalidades.



# An Open Standard.

- En 1992 se crea el ARB.
- Consistía en una serie de proveedores de software y hardware que decidían el futuro de OpenGL.
- Determinaban las nuevas características de OpenGL.
- Decidieron que las extensiones eran las principales características de las próximas versiones de OpenGL.
- ARB tenían que aprobar a través de los testing.
- Esta prueba consistía en una prueba de compatibilidad.

# OpenGL en Windows.

- OpenGL se aplicaba en UNIX.
- En 1993 se introduce el Windows NT el cual no tenia aplicaciones de gráficos nativa en sus sistema.
- En 1994 se implementa OpenGL en Windows NT 3.5.
- No era completamente compatible con las implementaciones de OpenGL.
- Las aplicaciones fueron muy lentas.

# DirectX.

- Windows busco su propio API de gráficos 3D al ver oportunidad en el mercado de los videojuegos.
- Su primer intento consistía una serie de comandos a la GDI (Graphics Device Interface).
- En 1995, para poder proporcionar funcionalidades 3D Microsoft adquiere RenderMorphics.
- Produjeron un API llamado Reality Lab, el cual ha sido renombrado como Direct3D y así crean un SDK llamado DirectX.
  - DirectDraw
  - DirectInput
  - DirectPlay
  - DirectSound
- En las primeras versiones era muy incomodo para los desarrolladores, por esta razón aún seguían introduciendo OpenGL y así hacer su API mas competitivo.

## El comienzo “La guerra de los APIs”.

- 1996 John Carmack desarrolla el famoso videojuego Quake el cual utiliza el API de OpenGL.
- Existía una gran diferencia entre el código necesario para ambas APIs.
- Direct3D convirtió mas utilizable su API con su versión 5.0.
- En este punto ambas APIs eran igual de utilizables.

# Buffers.

- La metodología utilizada por los programadores consiste en emitir una lista de comandos que serán interpretados por la GPU (Modo inmediato).
- Se introdujo un nuevo método que consistía en objetos de Buffers.
- Se almacenaba en memoria y se trasladaban a la GPU en cada llamada.
- Se mantenían ahí hasta que ya no son necesarios.
- Estos objetos se le denominan Vertex Buffer Objects (VBOs).

# Shaders.

- En el año 2000 Microsoft lanza la versión Direct3D 8.0.
- Soporta una nueva característica llamada Shaders.
- Shaders son pequeños programas que corren sobre la GPU.
  - Vertex shaders. Se ejecuta una vez por vértice.
  - Pixel shaders. Se ejecuta una vez por pixel.
- Shaders permiten un mayor rendimiento mediante la eliminación del CPU.
- Eran difíciles de programar debido a que se parecían al ensamblador.
- En 2003 se lanza la versión Direct3D 9.0 que incorpora Shaders llamados HLSL.
- Programación de alto nivel basado en C.

# OpenGL se estanca.

- OpenGL se estanca debido a que no soporta shaders.
- En el 2004 es lanzada la versión de OpenGL 2.0 y se introduce GLSL.
- OpenGL cae drásticamente ante Direct3D en términos de características básicas.
- Del 2004 al 2006 Direct3D dominaba el mercado y se incremento mas cuando el Xbox 360 fue lanzado en 2005.
- En 2006 se lanza la versión OpenGL 2.1 con muy pocas mejoras.
- Microsoft lanzó Direct3D 10.0 junto con su sistema operativo Windows Vista.
- Se crea una nueva metodología mas programable que OpenGL carecía.

# El Nuevo OpenGL.

- En el 2006 es administrado por el grupo Khronos.
- El grupo Khronos es un consorcio de proveedores de hardware y software que se encargan de crear y mantener APIs abiertos.
- Dos versiones de OpenGL fueron anunciados.
  - Estas versiones prometieron erradicar el renderizado de modo inmediato y hacerlo únicamente con buffers y shaders.
  - Crear objetos con pocas funciones.
- Longs Peak debería ser compatible con el hardware de esa época y conservar la compatibilidad de las versiones anteriores.
- Mt. Evans debería eliminar la compatibilidad hacia el pasado y ver hacia el futuro.



# OpenGL 3.0.

- En 2008 se lanza el API de OpenGL 3.0, no había rastros de haber cambiado mucho.
- Las compañías decidieron cambiar hacia DirectX.
- En marzo de 2009 fue lanzada la versión del API OpenGL 3.1, donde elimino toda funcionalidad del modo inmediato.
- Meses mas tarde OpenGL 3.3 fue liberado con la inclusión de shaders de Geometría.

# OpenGL 4.0.

- En 2010 lanzan OpenGL 4.0 como una API de generación de GPUs al igual que Direct3D 11.
- Tessellation, permite un control mas preciso de superficies y detalles automáticos en la escena.
- Teléfonos utilizan OpenGL ES.
- Cross-browser utilizan el API llamado WebGL.

# Bibliografía.

- <https://www.opengl.org/>
- OpenGL 4.0 Shading Language Cookbook, David Wolff, Packt publishing, July 20
- <https://www.opengl.org/documentation/books/>
- <http://openglbook.com/the-book.html>