

UNIVERSIDAD PRIVADA DEL NORTE
FACULTAD DE INGENIERÍA

CARRERA:
INGENIERÍA DE SISTEMAS COMPUTACIONALES

CICLO 2015 - 1

COMPUTACIÓN GRÁFICA Y VISUAL
“WebGL”

Autor:

Chávez Hinostroza, Carlos

DOCENTE:

Ing. Luis Miguel Cotrina Malca.

CAJAMARCA – PERU

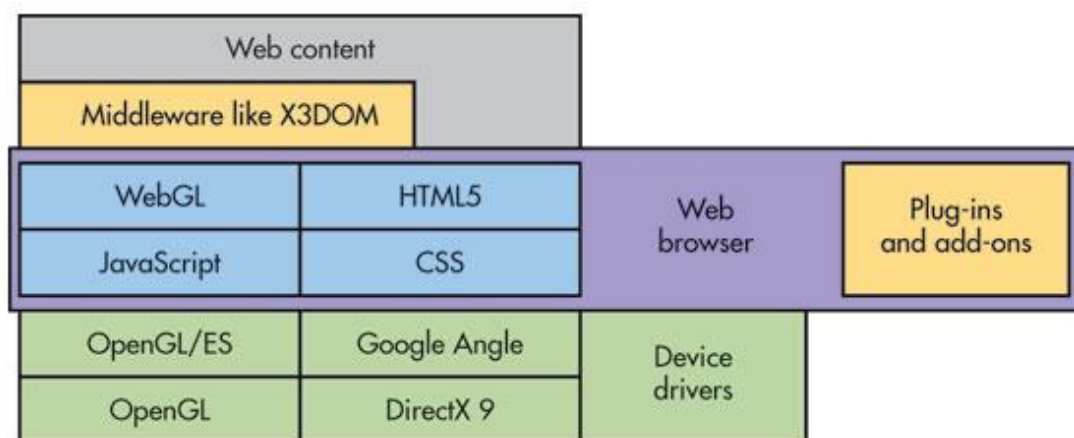
2015

WebGL

WebGL es una especificación estándar que sirve para desplegar gráficos en 3D en navegadores web. WebGL permite activar gráficos en 3D acelerados por hardware en páginas web, sin la necesidad de plug-ins en cualquier plataforma que soporte OpenGL 2.0 u OpenGL ES 2.0. Técnicamente es un enlace (binding) para javascript para usar la implementación nativa de OpenGL ES 2.0, que será incorporada en los navegadores.

Algunas bibliotecas en desarrollo que se están incorporando WebGL incluyen el C3DL y el WebGLU. Utiliza el elemento canvas del HTML 5.

Los propietarios de los navegadores más famosos y utilizados, Google (**Chrome**), Opera (**Opera**), Mozilla (**Firefox**) y Apple (**Safari**), son miembros del grupo de trabajo del consorcio Khronos para el desarrollo de WebGL, además de muchos otros programadores independientes de aplicaciones 3d.



WebGL complements HTML5 and fills out the Web browser, providing access to accelerated 3D display technology.

WebGL complementa HTML5 y rellena el navegador Web, facilitando el acceso a la tecnología de visualización 3D acelerado.

DIFERENCIAS	
OPENGL	WEBGL
Computadora de escritorio centrada API (como Direct3D).	Se deriva de OpenGL ES 2.0 (destinado a dispositivos móviles) que tiene menos capacidades y es más sencillo de usar.
Tiene shaders, shaders de teselacion y shaders de cómputo.	Está diseñado para ejecutarse en un navegador, y tiene, por tanto, algunas limitaciones más de OpenGL ES 2.0.
	Solo soporta vértices y sombreados de fragmentos.
	No tiene texturas 3D.

Trabajando con WebGL

1. Iniciamos con un archivo HTML:
2. Definir en la etiqueta <body> la función encargada de gestionar el interface gráfico.
3. Cargar los scripts externos que vayamos a necesitar (en este caso los propios de la API WebGL y otro necesario para trabajar con matrices.)
4. Una vez definidos estos aspectos ya podemos comenzar a programar nuestro propio script en WebGL.
5. Indicar que función hemos de cargar a la vez que el código, para nuestro caso será webGLStart(). En dicha función vemos como en primer lugar se ha de inicializar el canvas, llamando a la función initGL() donde, entre otras cosas, se comprueba que el navegador soporta WebGL;
6. A continuación se inicializan los shaders, así como los buffers.
7. Una vez inicializados los buffers tenemos que pasar a inicializar la escena con las funciones que podemos encontrar en webGLStart, para finalmente hacer una llamada a setInterval, donde se especifica que se ha de redibujar la escena cada 15 ms, la función que contiene la escena, drawscene.
8. A continuación definimos una perspectiva, esto es importante puesto que al definirla conseguimos que según la posición de los objetos en el espacio estos se verán de distintos tamaños.
9. Una vez definida la perspectiva hemos de crear una matriz identidad, y lo hacemos con loadIdentity().
10. Luego de esto se pasa a dibujar los objetos en la escena. Y finalmente con estos pasos ya se podrá dibujar una escena.

Bibliografía

- Antuña,E.,Diaz,A. & Gonzalo,D. (13 Diciembre del 2010). *Primer Programa* (Desarrollo en WebGL).Recuperado de:
<https://sites.google.com/site/desarrolloenwebgl/tutorial-1/2---primer-programa>
- S/N. (24 Octubre del 2011). *Tutoriales WebGL*.Recuperado de:
<http://www.jlabstudio.com/webgl/tutoriales-webgl/>
- Santiago,A.(13 Diciembre del 2010). *Leccion 1:Primeros Pasos con WebGL*.Recuperado de:
<http://ayudasprogramacionweb.blogspot.com/search/label/Guia%20WebGL>