**WebGL**

WebGL es una especificación estándar que está siendo desarrollada actualmente para mostrar gráficos en 3D en navegadores web. El WebGL permite mostrar gráficos en 3D acelerados por hardware (GPU) 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 API para javascript que permite usar la implementación nativa de OpenGL ES 2.0 que será incorporada en los navegadores. WebGL es gestionado por el consorcio de tecnología sin ánimo de lucro Khronos Group .

¿Que es WebGL?. WebGL (Web-based Graphics Library) es un estandar web multiplataforma para una API de gracos 3D de bajo nivel basado en OpenGL ES 2.0 y expuesto a traves del elemento canvas de HTML5 como interfaces DOM (Document Object Model). Esta API provee enlaces de JavaScript a funciones OpenGL haciendo posible proveer contenido 3D acelerado en hardware a las paginas web. Esto hace posible la creacion de graficon 3D que se actualizan en tiempo real, corriendo en el navegador.

Como en cualquier biblioteca de gráficos 3D, en WebGL, necesita ciertos componentes que estar presente para crear una escena 3D. Los componentes que nos referimos son los siguientes:

Lienzo o cambas: Es el marcador de posición donde se representará la escena. Es un estándar Elemento HTML5 y como tal, se puede acceder mediante el Document Object Model (DOM) a través de JavaScript.

Objetos : Estas son las entidades en 3D que forman parte de la escena. Estas entidadesse componen de triángulos.

Luces : Nada en un mundo en 3D se pueden ver si no hay luces. Este elemento de cualquierAplicación WebGL.

Cámara: Los actos de lona como la ventana al mundo 3D. Vemos y exploramos una escena 3D a través de él.

**HTML5 Elemento Canvas**

WebGL usa el elemento Canvas introducido en el estándar de HTML5 para renderizar. El elemento Canvas se introdujo como herramienta para poder pintar gráficos de forma Scriptable, mayoritariamente con Javascript. Aparte también ha servido para el alojamiento de WebGL.

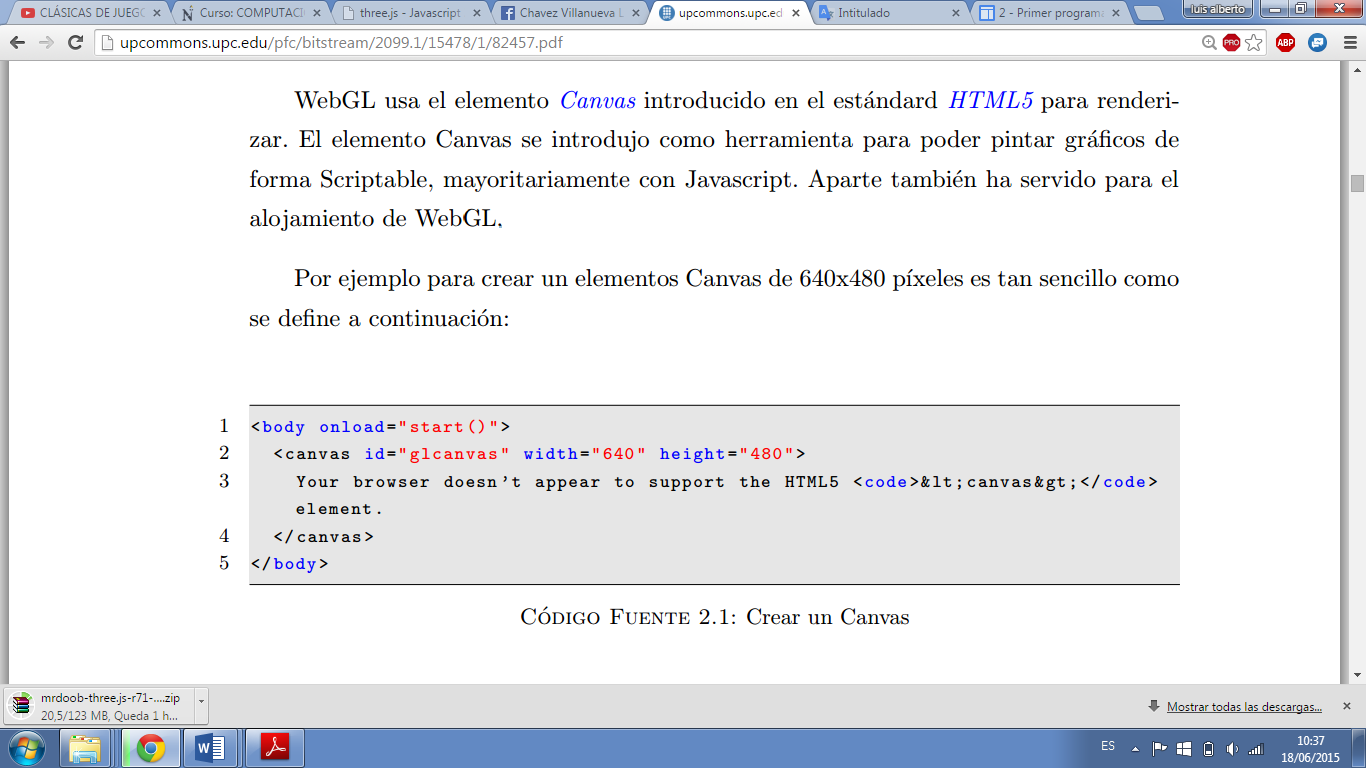


Figura 1. Creacion del camvas

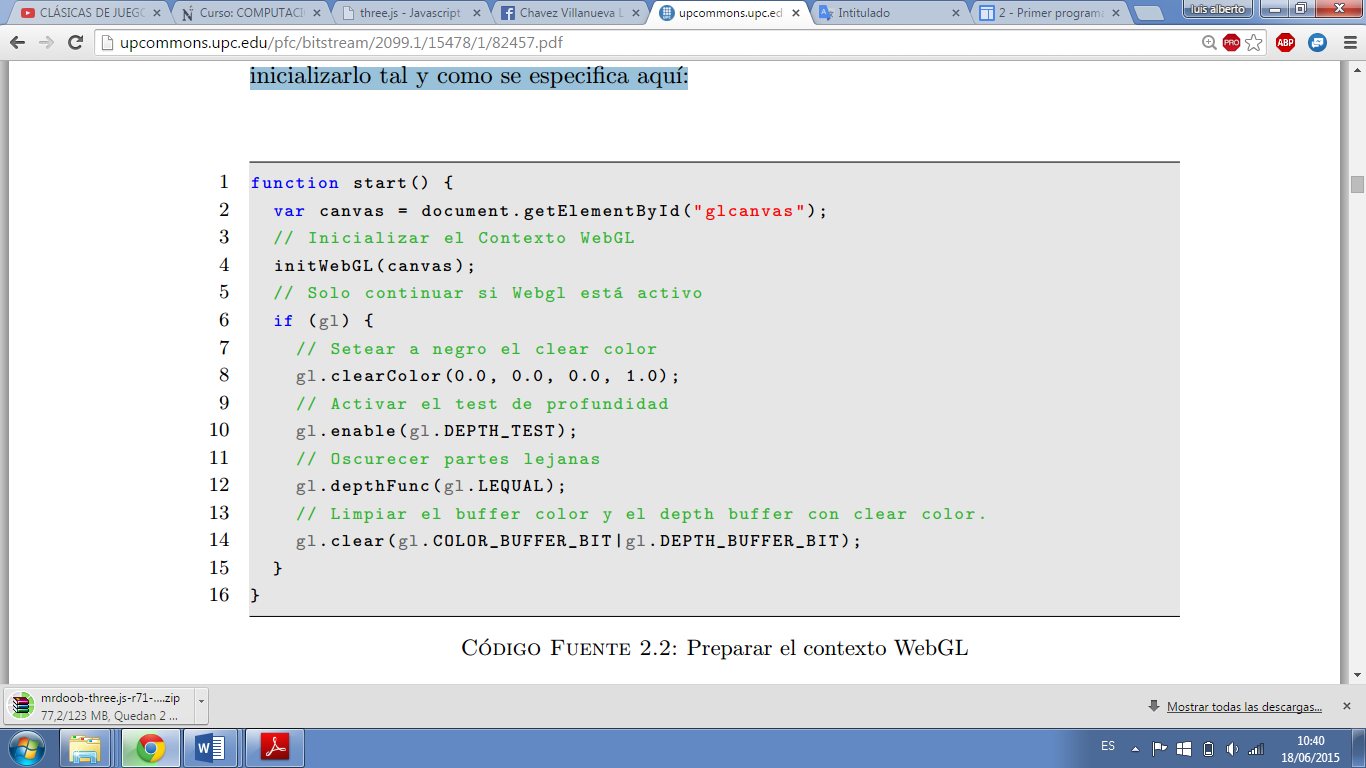
Una vez tenemos un elemento Canvas dentro del entorno DOM hace falta recoger un contexto WebGL e inicializarlo. Cuando recibimos un contexto WebGL es necesario inicializarlo tal y como se especifica aquí:

Figura 2. Preparar el contexto WebGL

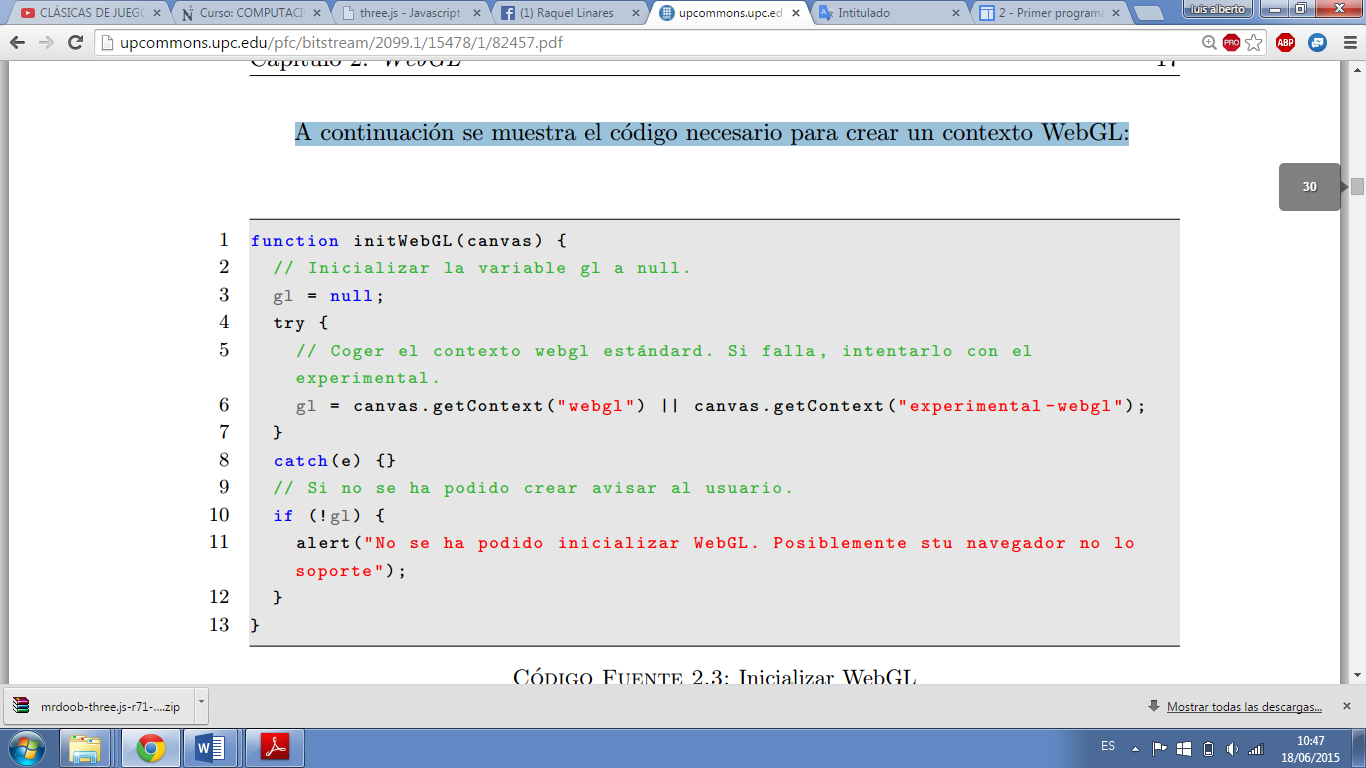
A continuaci´on se muestra el código necesario para crear un contexto WebGL:

Figura 3. Inicializar WebGL

Con éstos pasos ya tenemos un Canvas WebGL listo para empezar a recibir llamadas.

****

Figura 4. Modelo conceptual de la arquitectura WebGL.

# Bibliografía

Cantero, C. (s.f.). *Web-based Graphics Library.* universidad catolica nuestra señora de la asuncion.

Cantor, D., & Brandon, J. (2012). *WebGl.* birmingham: Packt.

Crespo, J. (2012). *diseño y desarrollo de un juego en webGl.* Barcelona: universidad politecnica de catalunya.

wikipedia, c. d. (15 de junio de 2015). *WebGL*. Obtenido de WebGL: https://es.wikipedia.org/w/index.php?title=WebGL&oldid=81429026