



Ingeniería en Producción Multimedia

Física III

DOCUMENTACIÓN

Paula Remy Ochoa

Romina Verdugo Corral

Blanca Lidia Castellanos Solis

Romina Gpe. Vega Alvarado

Marianne Cordova Amador

Cd. Obregón, Sonora, 24 de abril de 2020

WebGL

WebGL nació en 2011, basado en la tecnología OpenGL 2.0, incluye una API con funciones similares a su precursora en su versión 1.0. La diferencia con la tecnología anterior, es que WebGL está diseñado para ejecutar operaciones de renderizado pesadas, para llevar a cabo una representación 3D a través del elemento de HTML5. en los navegadores que lo soporten sin el uso de plug-ins. Al día de hoy WebGL ya está presente en las builds de Mozilla Firefox, Mozilla Fennec, Google Chrome y Safari 5.1. WebGL utiliza la placa de video del usuario, como acelerador hardware, para desarrollar la presentación de los gráficos y mejorar el rendimiento. Para este propósito, las características de la tarjeta gráfica son importantes, si la placa de video no fuera compatible con la tecnología, se realizaría la aceleración por software, pero de forma menos eficaz.

Los programas escritos en WebGL tienen dos partes, código de control escrito en lenguaje JavaScript y el código de efectos especiales (código shader).

Los elementos WebGL se pueden mezclar con otros elementos HTML y componerse con otras partes de la página o página de fondo. Algunos ejemplos de las primeras aplicaciones de WebGL incluyen Google Maps y Zygote Body. También podemos encontrar Unity Engine, que permite crear contenido (Videojuegos, apps, etc...) para navegadores usando esta tecnología.

La compatibilidad con los navegadores web modernos es muy buena, tanto en navegadores de escritorio como en dispositivos móviles, aunque en ciertos casos será necesario que el usuario actualice los drivers de la placa de video de su dispositivo para su correcto funcionamiento. En la siguiente ilustración se muestra las compatibilidades para los distintos navegadores web, numerando sus versiones funcionales. En la parte superior derecha, se puede observar también el porcentaje de compatibilidad tanto en navegadores en nuestro país y a nivel global.

WebGL es un estándar de web abierta para representar gráficos en cualquier navegador compatible, sin necesidad de plugins adicionales. WebGL se integra totalmente con todos los estándares web del navegador y permite el uso acelerado por GPU de procesamiento de imágenes y efectos como parte del lienzo de la página web. Los elementos WebGL pueden incrustarse en otros elementos HTML y combinarse con otras partes de la página.

WebGL es una API de bajo nivel, por tanto, está en contacto directo con el hardware del usuario. Esto significa que el programador debe ocuparse de controlar una serie de aspectos complejos sobre el procesamiento computacional de los gráficos, que en otras librerías ya pueden estar resueltos automáticamente. Por este motivo, la API de WebGL no es fácil de aprender.

¿Qué se puede hacer en WebGL?

WebGL se puede aplicar, por ejemplo, para la creación de videos musicales interactivos, juegos, visualización de datos, arte, entornos de diseño 3D, modelado 3D de espacio, modelado 3D de objetos, trazado de funciones matemáticas o creación de simulaciones físicas.

Actualmente, una de las principales aplicaciones es el soporte de marketing para un servicio, una película o productos, principalmente en el campo de lujo. Un experimento WebGL, es un gran activo en una campaña digital: ayuda a contar una historia y permite al usuario interactuar con ella.

Los museos y el campo cultural tenían un interés temprano en esta área, construyeron muchas instalaciones digitales y aplicaciones a su alrededor.

La juventud de la tecnología permite su originalidad. La mayoría de las veces vinculado con un toque de gamificación, WebGL ayuda a que aumente el 3D en la Web. La tecnología a menudo se ve como una experimental, que necesita dispositivos realmente rápidos, pero no hay que confundirnos, se pueden obtener algunos resultados bastante rudos.

WebGL tiene tres ventajas distintas sobre escribir código que simplemente manipula el DOM:

- **Tareas:**

Dibujar materiales reflectantes o iluminación compleja genera una tonelada de sobrecarga, y al ver que JavaScript tiene un solo subproceso y está vinculado a la CPU, ¿por qué no descargar algo de eso a la GPU en su dispositivo y hacer que haga el trabajo pesado?

- **Rendimiento:**

Utilizando la aceleración de hardware (la GPU integrada en su dispositivo), WebGL es ideal para juegos o visualizaciones complejas.

- **Shaders:**

Los efectos visuales complejos se pueden producir con pequeños programas conocidos como "shaders". Esto puede ser tan simple como producir un efecto de coloración sepia o simulaciones más complejas, como agua o llamas.

JavaScript

JavaScript el lenguaje más importante y potente en la web.

JavaScript, es uno de los más potentes e importantes lenguajes de programación en la actualidad, por tres enfoques claros: es útil, práctico y está disponible en cualquier navegador web.

JavaScript es creado por Brendan Eich y vio la luz en el año 1995 con el nombre de LiveScript, que luego fue nombrado JavaScript, nace como un lenguaje sencillo destinado a añadir algunas características interactivas a las páginas web. Sin embargo, hoy en día ha crecido de manera acelerada y es el lenguaje de programación que se utiliza en casi todos los sitios web en el mundo.

El poder de JavaScript está disponible principalmente en lado frontend, agregando mayor interactividad a la web, también puedes usar las librerías y framework como: jquery, angular, backbone, react y demás, escritas sobre JavaScript, y que te ayudan a crear una mejor experiencia de usuario en nuestros sitios web.

De igual manera JavaScript se puede utilizar en los servidores web. Node.JS es tu mejor opción para usar este lenguaje del lado del servidor.

Conozcamos ahora las características de JavaScript que hacen de este lenguaje, uno de los más populares en la actualidad.

1. Es Liviano.
2. Multiplataforma, ya que se puede utilizar en Windows, Linux o Mac o en el navegador de tu preferencia.
3. Es Imperativo y estructurado, mediante un conjunto de instrucciones indica al computador qué tarea debe realizar.
4. Prototipado, debido a que usa prototipos en vez de clases para el uso de herencia.
5. Orientado a objetos y eventos.
6. Es Interpretado, no se compila para poder ejecutarse.

Estas son las características que hacen de javascript un lenguaje que te permite desarrollar aplicaciones gigantes y potentes, como lo es: google doc, facebook, twitter e incluso capaz de ejecutarse en el servidor como un servidor Web muy rápido, gracias a nodejs.

¿Qué se puede hacer en JavaScript?

JavaScript brinda a los diseñadores HTML una herramienta de programación. Los diseñadores HTML normalmente no son programadores, pero JavaScript es un lenguaje con una sintaxis muy simple. Casi todo el mundo puede poner pequeños "fragmentos" de código en las páginas HTML.

Realmente, las cosas que se pueden hacer con JavaScript son muy variadas, entre las y las más destacadas son:

- Puede trabajar con diálogos emergentes e interactivos creados con elementos div, en lugar de ventanas emergentes, que por razones de seguridad y diseño han dejado de utilizarse.
- JavaScript puede poner texto dinámico en una página HTML, un comando de JavaScript como: `document.write("<h1>" + nombre + "</h1>")` puede escribir un texto o variable en una página HTML.
- Puede crear algunas animaciones como transiciones de imágenes y objetos de una página web.
- Obtener la información sobre el navegador que está usando el usuario, la versión de este, el sistema operativo sobre el que se está ejecutando e, incluso la resolución de pantalla que tiene configurada en su computadora.
- Para crear cookies, almacenar y recuperar información en equipo del visitante.
- Puede crear sistemas de menú sofisticados con submenús emergentes que se activan con la acción del usuario.
- Se pueden validar los valores que se ingresan en los campos de formulario antes de que sean enviados al servidor.
- Puede crear árboles de navegación para facilitar a los usuarios moverse de una página a otra por su sitio web.
- Puede crear efectos de sustitución de imágenes controlados por la acción de colocar el puntero del ratón o retirarlo de la misma.
- Puede cambiar la posición de elementos HTML en la página web de forma dinámica o controlada por el movimiento del puntero del ratón.
- Puede redirigir al usuario de una página a otra, sin necesidad de un enlace estático.
- Puede realizar algunos cálculos con los valores ingresados en campos de formulario.
- Puede obtener la fecha del sistema operativo donde se está ejecutando la página web en el cliente.

Referencias

Adobe Animate. (9 Septiembre de 2019). Creación y publicación de un documento de WebGL. 22 de Abril de 2020, de Adobe Sitio web: <https://helpx.adobe.com/mx/animate/using/creating-publishing-webgl-document.html>

Crespo Amigo, J. (2012). Diseño, estudio y desarrollo en WebGL.

Parisi, T. (2012). *WebGL: up and running*. " O'Reilly Media, Inc."

Sanz Asenjo, A. (2016). Tutorial web para el aprendizaje de WebGL.

Cimas Bravo, Ó. (2017). Editor y prevvisualizador de WebGL.

ManualesdelaWeb. (s. f.). QUE ES JAVASCRIPT, PARA QUE SIRVE, Y QUE PUEDO HACER CON EL. Recuperado 23 de abril de 2020, de <http://manualesdelaweb.com/a/que-es-javascript-para-que-sirve-y-que-puedo-hacer-con-el-20110117>

Retromix, UBD. (2012, septiembre 20). ¿Qué se puede hacer con JavaScript? Recuperado 23 de abril de 2020, de <http://lenguajesscript.blogspot.com/2012/09/que-se-puede-hacer-con-javascript.html>

Yang, X. (s. f.). Tetris. Recuperado 23 de abril de 2020, de <https://www.uv.es/jac/guia/jscrip/juego/tetris.htm>

Juego del ahorcado. (s. f.). Recuperado 23 de abril de 2020, de <https://www.uv.es/jac/guia/jscrip/palabras.htm>

Calendario-agenda. Ejemplo de JavaScript. (s. f.). Recuperado 23 de abril de 2020, de <https://www.uv.es/jac/guia/jscrip/calendar.htm>

WebGL Samples. (s. f.). electroShock. Recuperado 23 de abril de 2020, de <https://webglsamples.org/electroShock/application.html>

WebGL Samples. (s. f.-a). Dynamic Cubemap. Recuperado 23 de abril de 2020, de <https://webglsamples.org/dynamic-cubemap/dynamic-cubemap.html>

Grados, J. G. (2019, octubre 15). ¿Qué es JavaScript? Recuperado 23 de abril de 2020, de <https://devcode.la/blog/que-es-javascript/>

Pérez, D. (2017, julio 3). ¿Qué es Javascript? Recuperado 23 de abril de 2020, de <http://www.maestrosdelweb.com/que-es-javascript/>

Capítulo 1. Introducción. (s. f.). Recuperado 23 de abril de 2020, de <https://uniwebsidad.com/libros/javascript/capitulo-1>

Pelhon, L. (2019, mayo 27). Qué es JavaScript: primeros pasos. Recuperado 23 de abril de 2020, de <https://www.hostgator.mx/blog/que-es-javascript/>

¿Qué es JavaScript? Introducción básica a JS para principiantes. (s. f.). Recuperado 23 de abril de 2020, de <https://www.hostinger.mx/tutoriales/que-es-javascript-introduccion-basica/>