

*24 DE ABRIL DEL 2020*

FORMAS BASICAS Y SHADING

webgl

MULTIMEDIA 4TO.

LA SALLE NOROESTE

TOPICOS DE FISICA

Integrantes: Jesús Alberto Hoyos Félix

Aracely Arizmendi León

Rubén Borbolla Castelo

Damián Eduardo Burboa Almada

***ABSTRACT*.**

Dentro del presente archivo, se presenta la documentación realizada durante la creación de figuras hechas a base de WebGL y AnimationFrame de JavaScript esta es una especificación estándar que define una API implementada en JavaScript para la renderización de gráficos en 3D dentro de cualquier navegador web.

WebGL nació en 2011 como una derivación de OpenGL, una API con funciones similares. La diferencia es que WebGL está diseñado para funcionar en páginas web; exactamente, a través del elemento <canvas> de HTML5. Esto implica que, a diferencia de lo que ocurre con Flash, los gráficos creados con WebGL se integren perfectamente con el Document Object Model (DOM), pudiendo interactuar con los demás elementos HTML de la página. WebGL utiliza aceleración por hardware para optimizar la presentación de los gráficos. Esto significa que recurre a la placa de video del usuario para mejorar el rendimiento. Así, el desempeño de WebGL depende en buena medida de las características de la tarjeta gráfica del usuario.

Si el usuario cuenta con una placa de video incompatible, la aceleración puede realizarse por software, aunque el resultado no será tan satisfactorio. La compatibilidad con navegadores modernos es muy buena, aunque en ciertos casos el usuario necesitará actualizar los drivers de video de su dispositivo.

Por su contacto directo con el hardware del usuario, WebGL es una API de bajo nivel. Esto implica que el programador debe preocuparse por controlar una serie de aspectos complejos sobre procesamiento computacional de gráficos, que en otras bibliotecas ya pueden estar resueltos automáticamente.

WebGL permite que el contenido web utilice una API basada en [OpenGL ES](http://www.khronos.org/opengles/) 2.0 para llevar a cabo la representación 2D y 3D en un elemento [canvas](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API) HTML en los navegadores que lo soporten sin el uso de plug-ins. WebGL consiste en código de control escrito en JavaScript y código de efectos especiales (código shader) que se ejecuta en la unidad de procesamiento gráfico de una computadora (GPU). Los elementos WebGL se pueden mezclar con otros elementos HTML y componerse con otras partes de la página o el fondo de esta.

<https://developer.mozilla.org/es/docs/Web/API/WebGL_API/Tutorial/Wtilizando_texturas_en_WebGL>

*Desarrollo:*

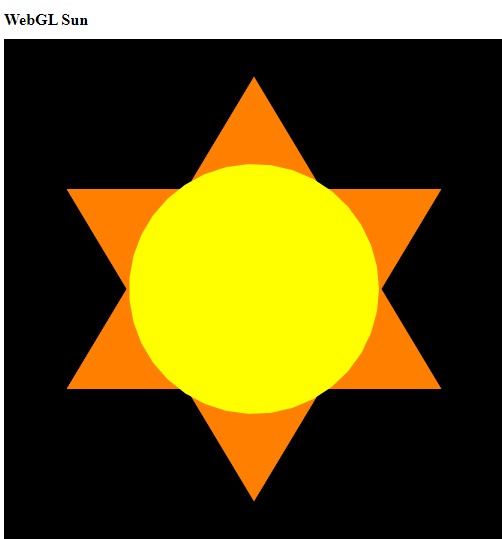
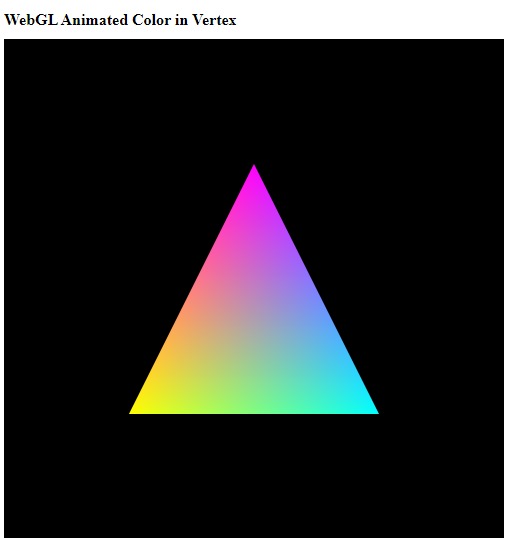
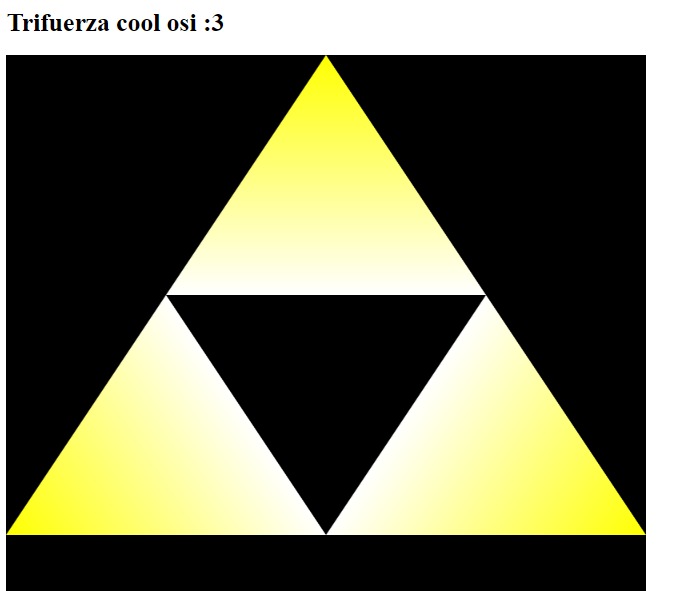
* ***Graphics 2D: Letra F:***

Como parte del proyecto parcial, se requirió utilizar un estándar que forma parte de JavaScript para la renderización en un gráfico 2D y que este puede ser visible en los navegadores, como HTML. Gracias a esta se logró realizar uno de los trabajos acordados, en el cual se necesitó la creación de 6 triángulos, equivalentes para formar cartas formas o figuras como son los quads para así poder llegar a un resultado final, que en este caso seria la realización de la letra F.

* ***Trifuerza (WebGL) y Sol(webGL):***

Al igual que en el trabajo mencionado anteriormente, para la realización de estos dos de igual forma se utilizo WebGL, ya que ambas figuras hacen presencia tanto de triángulos como de cirlces.

**Resultados:**



**Conclusión:**

Gracias a los conocimientos que obtuvimos ahora somos capaces de crear figuras y objetos en 2D mediante webGL, el cual es un conocimiento muy importante al momento de la creación de videojuegos, además cabe destacar que no todos los programadores se inclinan por este lado, por lo tanto, esto se convierte en una herramienta fundamental para nosotros como productores multimedia.

Sin duda alguna hubo varios retos en la realización del proyecto, pero logramos sacarlo adelante pese la contingencia ocurrida en estas fechas gracias a la colaboración y buena actitud que tenía el equipo de trabajo, es decir que hubo una buena sinergia.

**Referencias:**

* <https://developer.mozilla.org/es/docs/Web/API/WebGL_API/Tutorial/Adding_2D_content_to_a_WebGL_context>
* <https://webglfundamentals.org/webgl/lessons/webgl-text-texture.html>
* <https://webgl2fundamentals.org/webgl/lessons/webgl-text-texture.html>
* <https://www.youtube.com/watch?v=F7ISkQn9FZc&t=391s>
* <https://developer.mozilla.org/enUS/docs/Web/API/WebGL_API/Basic_2D_animation_example>
* <https://www.4rsoluciones.com/blog/webgl-graficos-complejos-en-nuestro-sitio-web/>
* <https://developer.mozilla.org/es/docs/Web/API/WebGL_API/Tutorial/Wtilizando_texturas_en_WebGL>
* <https://youtu.be/M4wgg5XS8js>
* <https://www.youtube.com/watch?v=YDCSKlFqpaU>
* <https://bl.ocks.org/robinhouston/4b736b08f369a94c419098a72e11c47b>
* <https://miketon.posthaven.com/webgl-triforce-prototype>
* <https://webgl2fundamentals.org/webgl/lessons/webgl-scene-graph.html>
* <https://developer.mozilla.org/es/docs/Web/API/WebGL_API/Tutorial/Adding_2D_content_to_a_WebGL_context>
* <https://blog.scottlogic.com/2019/05/28/drawing-2d-charts-with-webgl.html>
* <https://webglfundamentals.org/webgl/lessons/webgl-2d-scale.html>