UNIVERSIDAD MILITAR
NUEVA GRANADA

Shaders

Octubre 2019

JHOAN LEONARDO SANTANA 6000229 CARLOS LEONARDO PÉREZ TUNJANO 6000252

I. Introducción

En presente trabajo tiene como objetivo aprender y entender qué son los Shaders, para qué sirven, cómo funcionan y en cómo este conjunto de instrucciones son ejecutadas al mismo tiempo por cada uno de los píxeles de la pantalla, permitiendo una mejor visualización. También se anexará al presente documento un ejemplo en OpenGl de Shaders.

II. Desarrollo

¿Qué es un Shader?

Un Shader es simplemente un programa que se ejecuta en la tubería de gráficos y le dice a la computadora cómo representar cada píxel. Estos programas se llaman Shaders porque a menudo se usan para controlar los efectos de iluminación y sombreado.

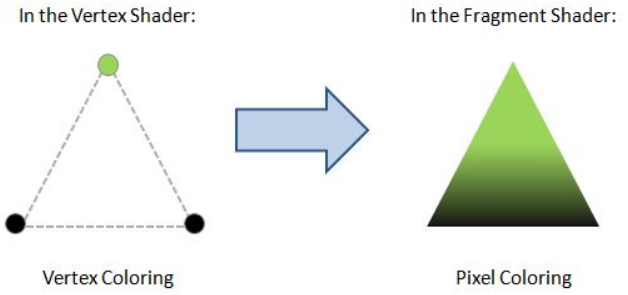
¿Qué es un vertex shader?

El vertex shader se llama en cada vértice. Este shader manipula por vértice datos como coordenadas de vértice, normales, colores y coordenadas de textura. Estos datos se representa mediante atributos dentro del vertex shader.
cada atributo apunta a una VBO desde donde se lee los datos de vértice.

¿Qué es un fragment shader?

Cada conjunto de tres vértices define un triángulo y cada elemento en la superficie de ese triángulo necesita ser asignado un color. De lo contrario nuestras superficies serían transparentes. Cada elemento de superficie se denomina fragmento. Dado que se trata de superficies que se van a visualizar en la pantalla, estos elementos son más comúnmente conocido como píxeles. El objetivo principal de la fragment shader es calcular el color de píxeles individuales.

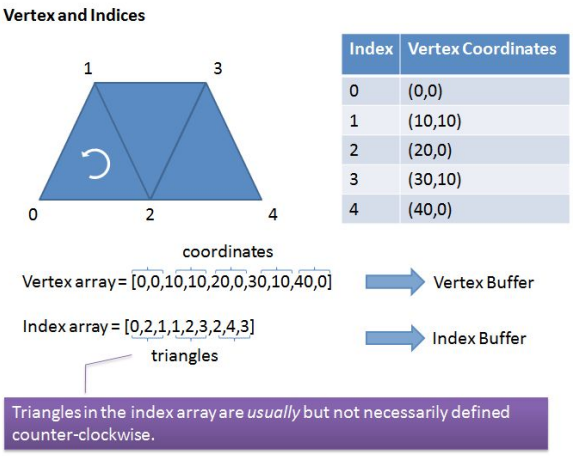
En el siguiente diagrama se explica esta idea:



¿Qué es un framebuffer?

Es un búfer bidimensional que contiene los fragmentos que han sido procesados por el sombreador de fragmentos. Una vez que se han procesado todos los fragmentos, se forma una imagen 2D o 3D y se muestra en la pantalla. El framebuffer es el destino final de la canalización de renderizado.

Para este ejemplo veremos como se recorren los Vértices e índices en el array para unos triángulos que forman un trapecio:



Como se puede ver en la captura de pantalla anterior, hemos colocado la secuencialmente coordenadas en la matriz de vértice y entonces hemos indicado en la matriz de índice de cómo se utilizan estas coordenadas para dibujar el trapecio. Por lo tanto, el primer triángulo se forma con los vértices que tienen índices 0, 1, y 2; el segundo con los vértices que tienen índices 1, 2, y 3; y finalmente, la tercera, con vértices que tienen los índices 2, 3 y 4. Se seguirá el mismo procedimiento para todas las geometrías posibles.

Introducción al Fragment Shader

El único propósito de un shader es devolver cuatro números: r, g, b, y a.

Eso es todo lo que hace o puede hacer. La función que ve frente a usted se ejecuta para cada píxel en pantalla. Devuelve esos cuatro valores de color, y ese se convierte en el color del píxel. Esto es lo que se llama Pixel Shader (a veces denominado Fragment Shader).

Con eso en mente, intentemos convertir nuestra pantalla en un rojo sólido. Los valores de rgba (rojo, verde, azul y "alfa", que define la transparencia) van de 0a 1, por lo que todo lo que tenemos que hacer es regresar r,g,b,a = 1,0,0,1. ShaderToy espera que se almacene el color de píxel final fragColor.

```
1  void mainImage( out vec4 fragColor, in vec2 fragCoord )
2  {
3      fragColor = vec4(1.0,0.0,0.0,1.0);
4 4 }
```

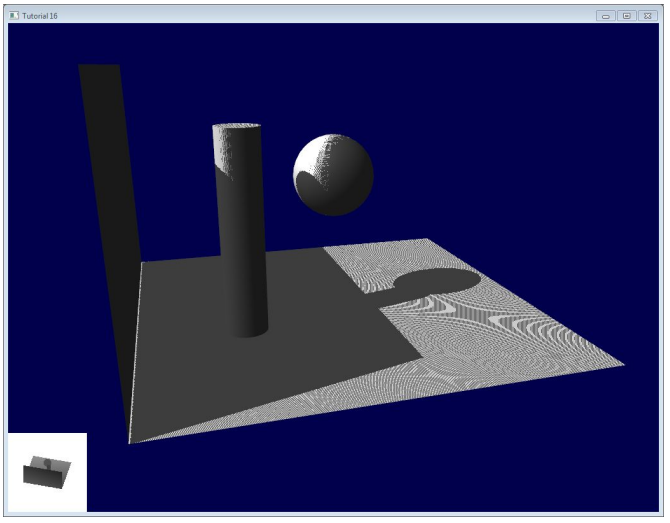
Cada sombreador de vértices inicia un sombreador de vértices por glDrawArrays ()

Un sombreador de vértices debe generar una posición en coordenadas de clip al rasterizador.

Usos básicos de sombreadores de vértices

- Transformaciones
- Iluminación
- Posiciones de vértices móviles

A Continuación se muestra imagen con ejemplo de OenGL:



III. Conclusiones

- Los Shaders son de gran importancia para mejorar la parte visual de cualquier página web, videojuego, entre otras.
- Estos Shaders toca saberlos usar debido a que no siempre van a funcionar como se desea.

IV. Bibliografía

- http://recursos.normalpopayan.edu.co:8983/wikipedia_es_all_2017-08/A/Shaders.html
- https://thebookofshaders.com/01/?lan=es
- http://recursos.normalpopayan.edu.co:8983/wikipedia_es_all_2017-08/A/Shaders.html
- https://www.ecured.cu/Shaders
- https://es.wikipedia.org/wiki/Sombreador
- https://www.cs.utexas.edu/~fussell/courses/cs384g-fall2013/lectures/lectureXX-OpenGL.pdf
- https://gamedevelopment.tutsplus.com/tutorials/a-beginners-guide-to-coding-graphics-shaders--cms-23313
- Cantor D. WebGL Beginner's Guide, (Junio 2012) Published by Packt Publishing Ltd. Copyright © 2012, Production Reference: 1070612