

# Shaders y Strips

Juan E Gomez, Daniel P Pedroza, Bayron D Morales

## Resumen –

En este documento se podrá apreciar como caracteres esenciales en la computación gráfica, tales como Strip, Shaders, etc., entran en el campo de investigación de polígonos y su construcción en three.js y en diversas plataformas que leen y pueden inclusive programar en dicho idioma de programación.

En esta guía se podrá ver los inicios el principio y como funciona con sus debidos ejemplos, aplicado a la computación gráfica.

## I. PROCEDIMIENTO PARA EL ENVÍO DEL TRABAJO

### A. Comentarios, Revision, Sugerencias

-

### B. Etapa Final

Enlace GitHub:

### C. Figuras.

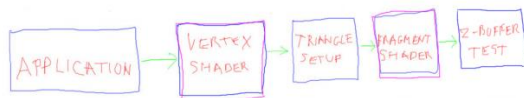


Fig. 1: Concepto básico de Shader.

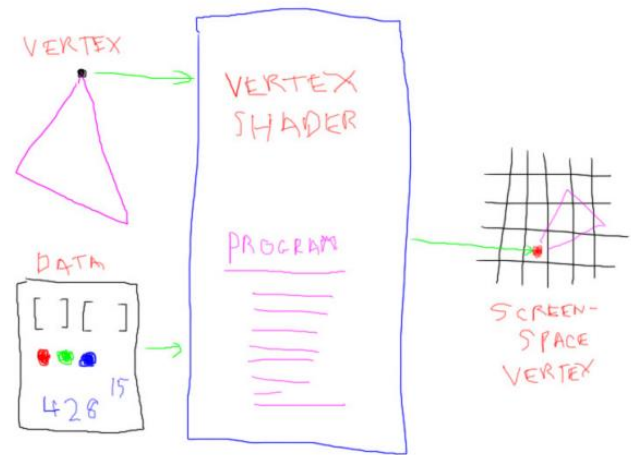


Fig. 2: Fragmentos de Shaders.

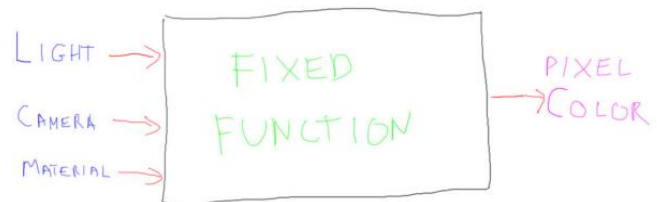


Fig. 3: Arquitectura.



Fig 4: Diferencia de Graficos.

Documento recibido el 23 de septiembre de 2020. Escriba los Nombres completos de los autores en el campo autor, pero no es necesario. Ponga un espacio entre los autores.

J. E Gomez is with the Militar University New Granada, Boulder, CO 80305 USA (corresponding author to provide phone: 303-555-5555; fax: 303-555-5555; e-mail: author@boulder.nist.gov).

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).



Fig. 5: Grafica comportamiento GPU.

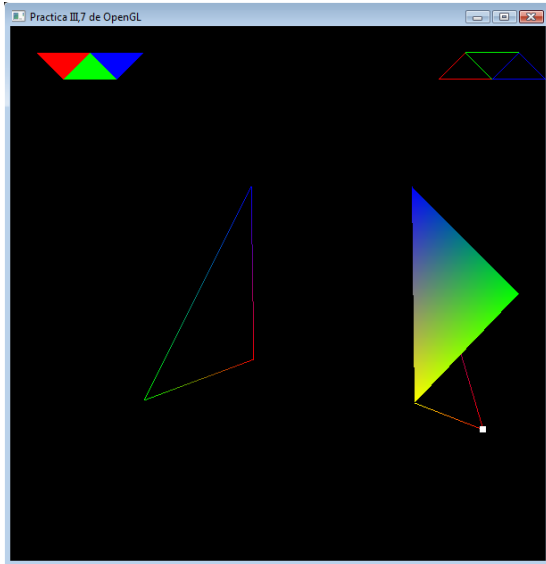


Fig. 6: Diferencia entre Strip y NoStrip.

```
vertexShader: [
  "uniform vec3 uMaterialColor;",
  "uniform vec3 uDirLight;",
  "varying vec3 vColor;",
  "void main() {",
    // Transform the vertex from model space to clip coordinates
    "gl_Position = projectionMatrix * modelViewMatrix * vec4( position, 1.0 );",
    "vec3 light = normalize( uDirLight );",
    // Compute a diffuse color using Lambertian reflection, N * L
    "float diffuse = max( dot( normal, light ), 0.0 );",
    "vColor = uMaterialColor * diffuse;",
  "}"
].join("\n"),
```

Fig. 7:Codigo Shader básico.

## II. SHADERS

Este es un concepto nuevo que se ha venido desarrollando hasta hace poco a un ritmo acelerado, y al igual que muchas tecnologías, o la potencia que se ha venido multiplicando exponencialmente hasta la fecha, esto nos da herramientas como programadores que nos permite hacer uso de distintos lenguajes o software que a pesar de manejarse de manera diferente, siempre cumplen con el mismo propósito y hacen uso de este lenguaje universal llamado programación.

Los Shader como su nombre lo dice ( sombra ), es un algoritmo que puede mejorar la estética y el rendimiento de un

programa o ponerlo lento ya que cuando se aplica un shader se puede estar abriendo una puerta “trasera” dicho de manera coloquial, lo que significa que puede estar usando dos o mas programas para procesar una figura creada por medio de código, o un modelo en 3d inclusive, por lo que a la hora de hacer estos, aunque aumenta la calidad visualmente hablando, hay un precio en términos de procesamiento como se puede ver en la Figura 3, que la arquitectura de un shader, está constituida por mínimo 3 funciones básicas, desde iluminación hasta el mismo material.

Como se pudo evidenciar en la figura la materia es un aspecto del shader, que, al juntarlo con otras piezas como el color o la luz, se genera este “sombreado” que es de donde sale su nombre, estos shader en su principio y cuando fueron aplicándose a los videojuegos por ejemplo lo que hacen es consumir un poco más de GPU, y CPU para adquirir calidad.

Citando el contenido anexo de las clases virtuales de Udacity, Leccionnumero 9:

*“En 2002 comenzaron a aparecer GPU que incluían sombreadores de vértices y fragmentos. Un sombreador es realmente un  
Pieza programable de la tubería. En otras palabras, envía poco o, a veces, mucho,  
programas de sombreado para cada una de estas unidades. Estos programas están escritos en un lenguaje Clike, si  
saber cómo es C o C++. Hasta este momento, hemos tenido tres.js encargados de la programación  
para nosotros. Cuando creamos un material, three.js crea dos pequeños programas, uno para cada uno  
los sombreadores de vértices y fragmentos. Cuando se va a mostrar el objeto con ese material, estos  
los sombreadores se cargan con sus programas. Luego, los triángulos se envían por la tubería y el  
se ejecutan los programas.”*

Como se pudo apreciar el GPU dieron un avance significativo a los shader y a su uso haciendo que sea mas común además de mejorando la calidad de los productos entregados, y como se estaba diciendo antes la evolución de los shader aunque un poco temprana los pasos agigantados que ha dado en tan pocos años se han evidenciado en piezas únicas tales como Grand Thief Auto, en este caso haciendo referencia a todas sus entregas, desde Vice City hasta la quinta entrega, se ha ido viendo una notable mejora de gráficos y obviamente de Shader, que aunque consumen mas computador y requieren de una potencia impresionante, la calidad es algo que no esta en discusión.

Ahora como se puede ver en la Fig 4, la diferencia del GPU puede influenciar en la calidad de los modelos, ya que es el mismo escenario y aun asi la calidad de lo shader es totalmente diferente.

Las imágenes son del Far Cry 5, un juego actual que salió el año 2018 que es relativamente actual y a pesar de tener dos

años de antiguo es un ejemplo perfecto de la aplicación de shader y la calidad.

Para ver ejemplo de un Shader (código), remítase a la Figura 7.

### III. STRIP Y FAN

Una de las funciones del Strip, al menos en Python, es la de quitar los espacios en código Sting (Letras y símbolos), lo que hace que quede todo junto, no obstante, una de las otras aplicaciones tales como three.js, sirve para darle doble cara a los sólidos como se puede evidenciar en la figura 6, donde se puede ver una diferencia entre usar el Strip y no usarlo, con exactamente la misma construcción y los mismos colores, esta aplicación del Strip permite usar los mismos lados de las construcciones, para darle mas figura e incluso por ciencia del color lo hace ver más grande.

Y citando la paina web, opengl-esp.superforo.net, julio 31 de 2007:

*“Los triángulos STRIP funcionan como las líneas STRIP, primero damos los vértices de un triángulo, y a partir de ahí, cada nuevo vértice, formará con los dos anteriores otro triángulo, creando así triángulos encadenados. Así con 5 vértices creamos 3 triángulos como en el ejemplo, observa los puntos y mira. En este caso va en el sentido contrario de las agujas del reloj (ya veremos para que digo esto).*

*Los triángulos FAN son parecidos a los STRIP, el ejemplo común que se dice siempre es que genera abanicos. La explicación es que, los 3 primeros vértices forman un triángulo, y partir de ahí con cada nuevo vértice, se genera un nuevo triángulo con el nuevo vértice, el anterior y el primero. A compartir todos los triángulos un punto, parece una abanico, pero en general se usa para hacer circunferencias. En el ejemplo va en el sentido de las agujas del reloj.”*

Fan es un tipo de construcción con características similares pero únicas que se puede usar para diferentes tipos de construcción por lo que lo lleva a que los resultados puedan variar dependiendo de los componentes y su vez Strip también puede usarse para construir otros tipos de objetos con conceptos diferentes.

A continuación, el código ejemplo del Fan y Strip.

Código Strip:

```
glShadeModel(GL_FLAT);
glBegin(GL_TRIANGLE_STRIP);
glColor3f(1,0,0);
glVertex3i(-9,9,0);
glVertex3i(-8,8,0);
glVertex3i(-7,9,0);
glColor3f(0,1,0);
glVertex3i(-6,8,0);
```

```
glColor3f(0,0,1);
glVertex3i(-5,9,0);
glEnd();
```

```
glBegin(GL_TRIANGLE_FAN);
glColor3f(1,0,0);
glVertex3i(8,8,0);
glVertex3i(6,8,0);
glVertex3i(7,9,0);
glColor3f(0,1,0);
glVertex3i(9,9,0);
glColor3f(0,0,1);
glVertex3i(10,8,0);
glEnd();
```

```
glShadeModel(GL_SMOOTH);
```

E

### IX. CONCLUSIÓN

#### REFERENCES

- [1] Gabriel Dalporto / <https://www.udacity.com/> junio 2011
- [2] Cnadyland / [https://www.youtube.com/watch?time\\_continue=3&v=Wi3soaTLcQM&feature=emb\\_logo](https://www.youtube.com/watch?time_continue=3&v=Wi3soaTLcQM&feature=emb_logo) / marzo 27 de 2018
- [3] Harze / <https://opengl-esp.superforo.net/t33-tutorial-iii-7-las-dos-caras-del-poligono-triangulos-strip-y-fan#:~:text=Los%20tri%C3%A1ngulos%20FAN%20son%20parecidos,el%20anterior%20y%20el%20primero.> / julio 31 de 2009.
- [4] G. S. Jackson / [https://techlandia.com/fucion-strip-python-info\\_254266/](https://techlandia.com/fucion-strip-python-info_254266/) / 2020.
- [5]