



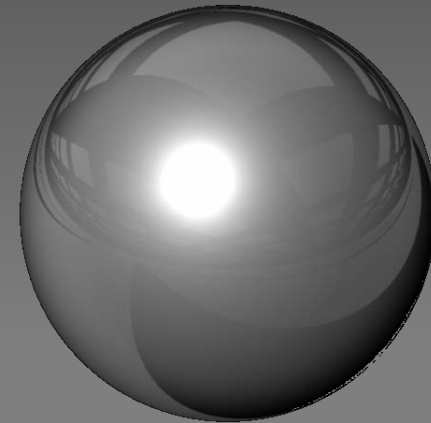
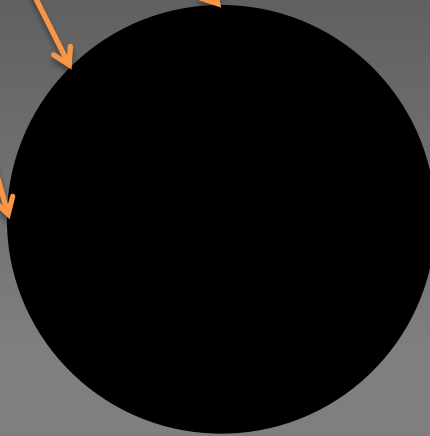
Introducción a los Shader (Materiales)

Presentado por Eduardo Roa

Definición



Un shader es un programa o algoritmo que se encarga de dar la apariencia de un objeto, a un nivel mas teórico se puede ver como el comportamiento de una superficie cuando un rayo de luz incide sobre el.



Sin embargo, en video juegos y producción, un shader puede utilizarse para hacer efectos visuales y efectos en imágenes 2D o postproducción

Historia



Silicon Graphics - Indy



1992 – Librería OpenGL

Pipeline Gráfico

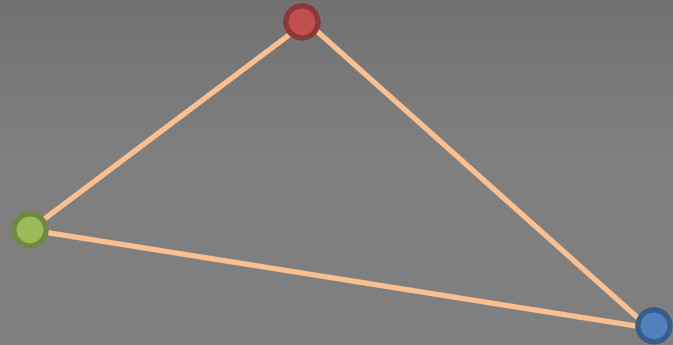
Aplicación 3D



```
glTranslatef(-1.5f,0.0f,-6.0f);
glBegin(GL_TRIANGLES);
    glVertex3f( 0.0f, 1.0f, 0.0f);
    glVertex3f(-1.0f,-1.0f, 0.0f);
    glVertex3f( 1.0f,-1.0f, 0.0f);
glEnd();

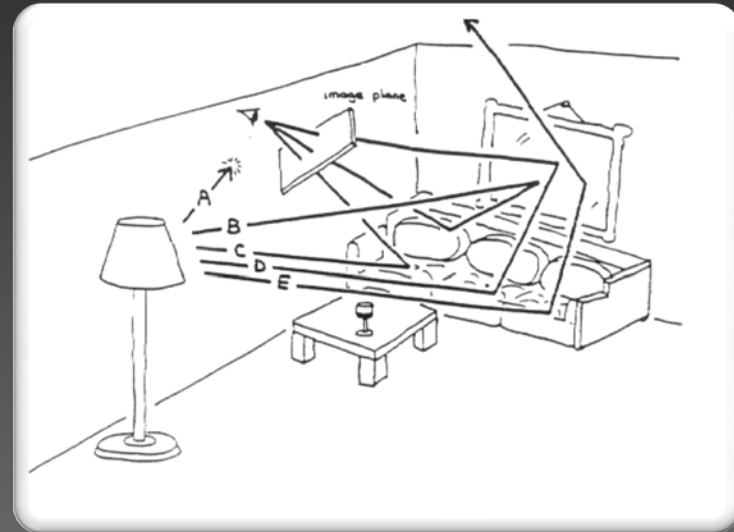
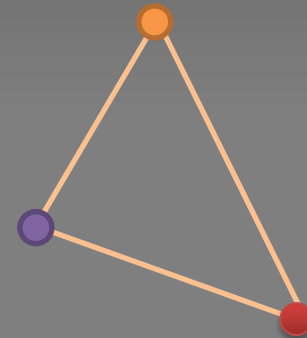
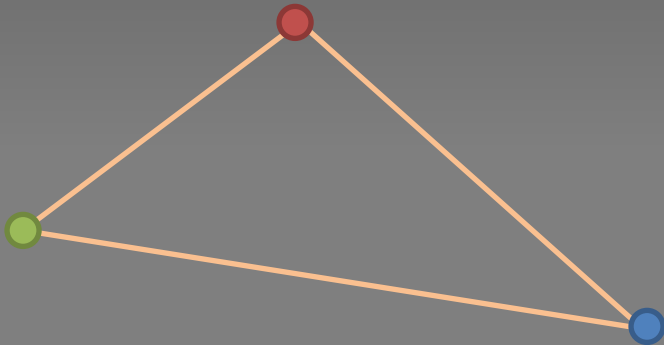
glTranslatef(3.0f,0.0f,0.0f);
glBegin(GL_QUADS);
    glVertex3f(-1.0f, 1.0f, 0.0f);
    glVertex3f( 1.0f, 1.0f, 0.0f);
    glVertex3f( 1.0f,-1.0f, 0.0f);
    glVertex3f(-1.0f,-1.0f, 0.0f);
glEnd();
```

En esta fase se definen la coordenadas de los vértices, normales, color del vértice de las primitivas que usara el programa.



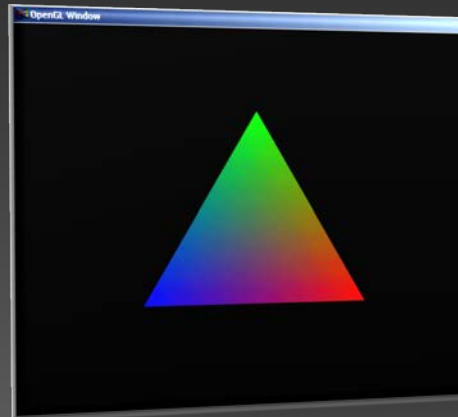
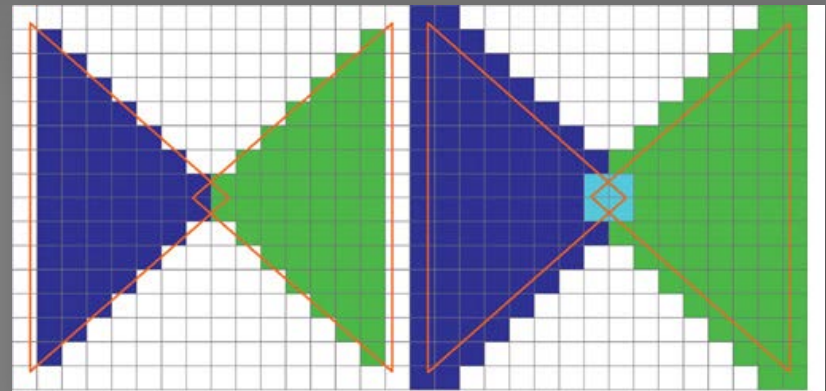
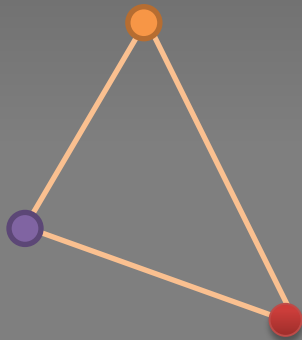
Pipeline Gráfico

Procesamiento de
Vértices

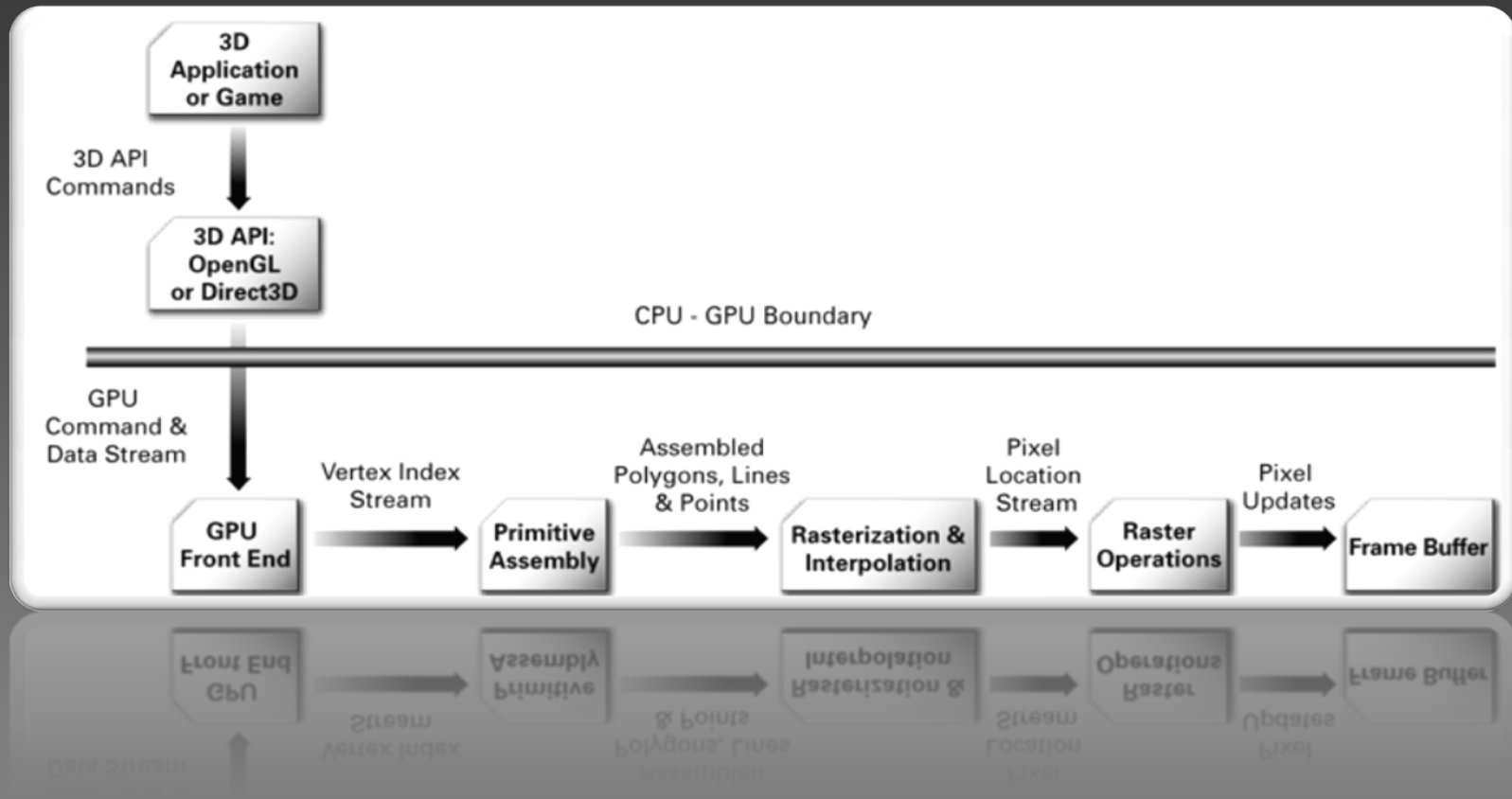


Pipeline Gráfico

Procesamiento de
Píxeles



Pipeline Gráfico



En los comienzos, las tarjetas gráficas implementaban este pipeline gráfico de manera fija o estática, es decir, no era posible la alteración o modificación de algunos de estos procesos.

Pipeline Gráfico



En 1996, se estrenan dos juegos que fueron iconos en la computación gráfica, Tomb Raider (lado izquierdo), QUAKE 1 (inferior)

La calidad gráfica estaba limitada tanto por la velocidad del CPU y las limitaciones del GPU

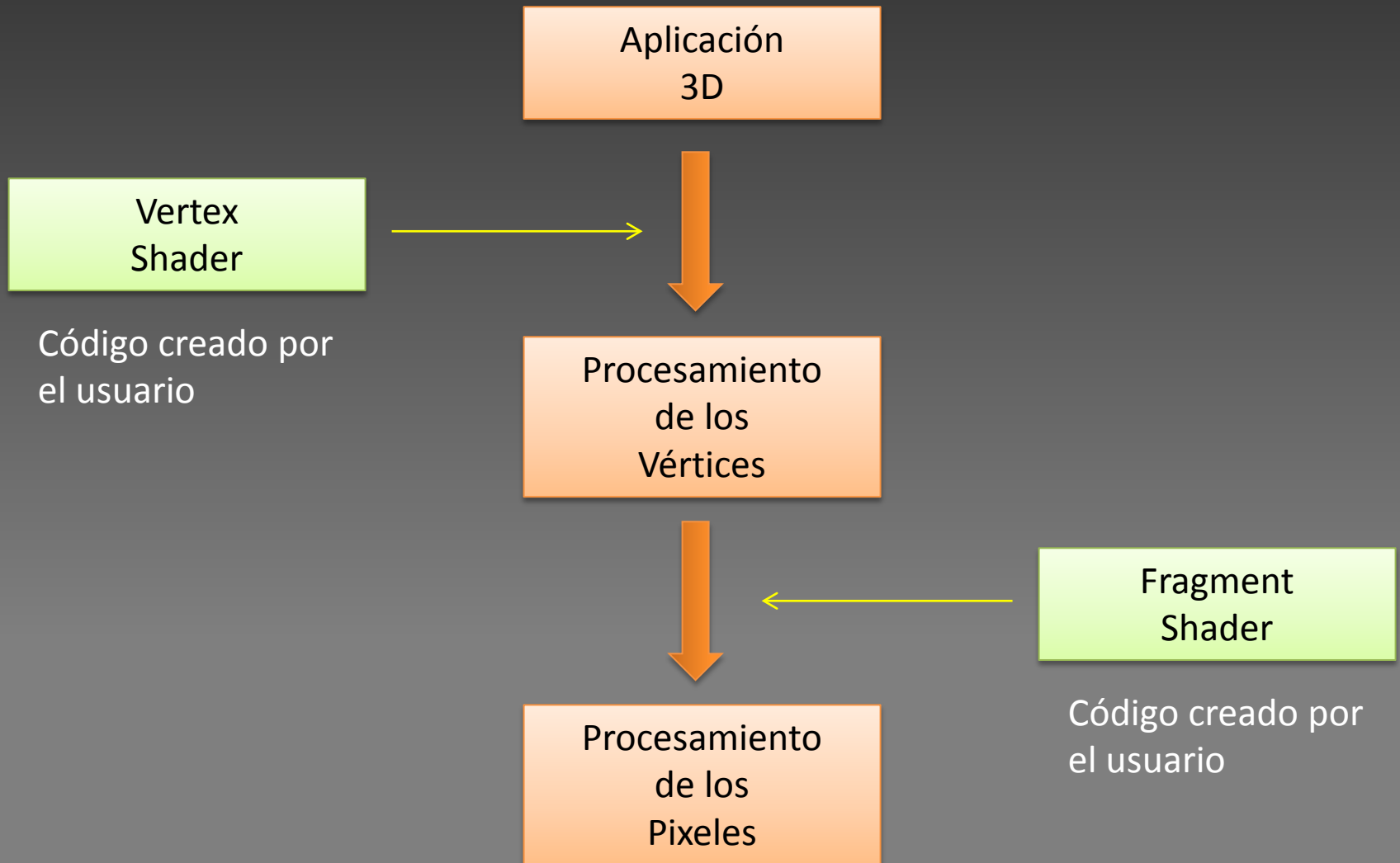


Renderman (1988-89) – Pixar Animation Studios

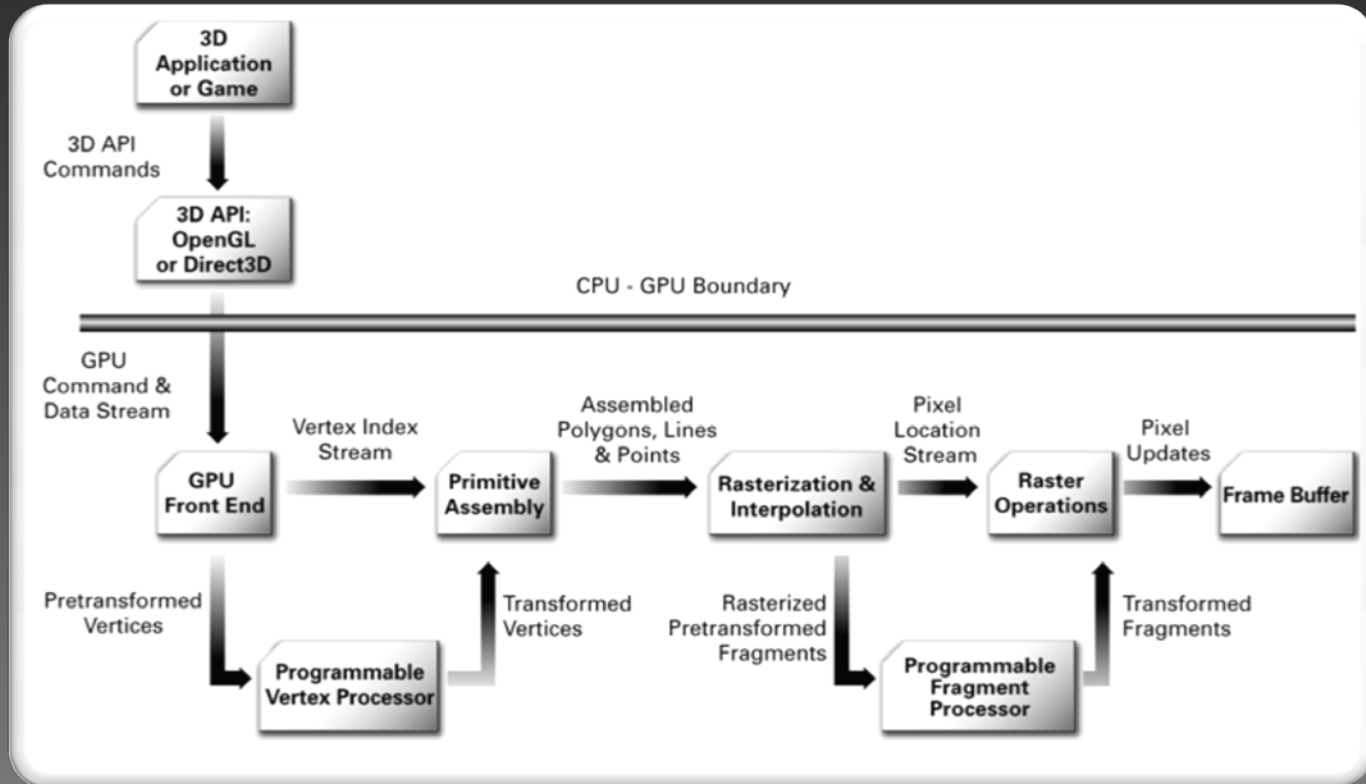


Renderman Language Shader, lenguaje basado en la sintaxis de C, que permitía al usuario programar shader de mayor complejidad

Pipeline Gráfico



Pipeline Gráfico



A partir del año 2000, empieza un cambio radical en el pipeline gráfico, OpenGL y DirectX le dan mas poder al programador para el uso del GPU. NVIDIA con la serie Geforce 3 y ATI con la RADEON 9700, presentan compatibilidad con OpenGL 1.2 y DirectX 9.0, donde se introduce la primera versión del lenguaje de shader (Shaders Language)

Efectos Visuales



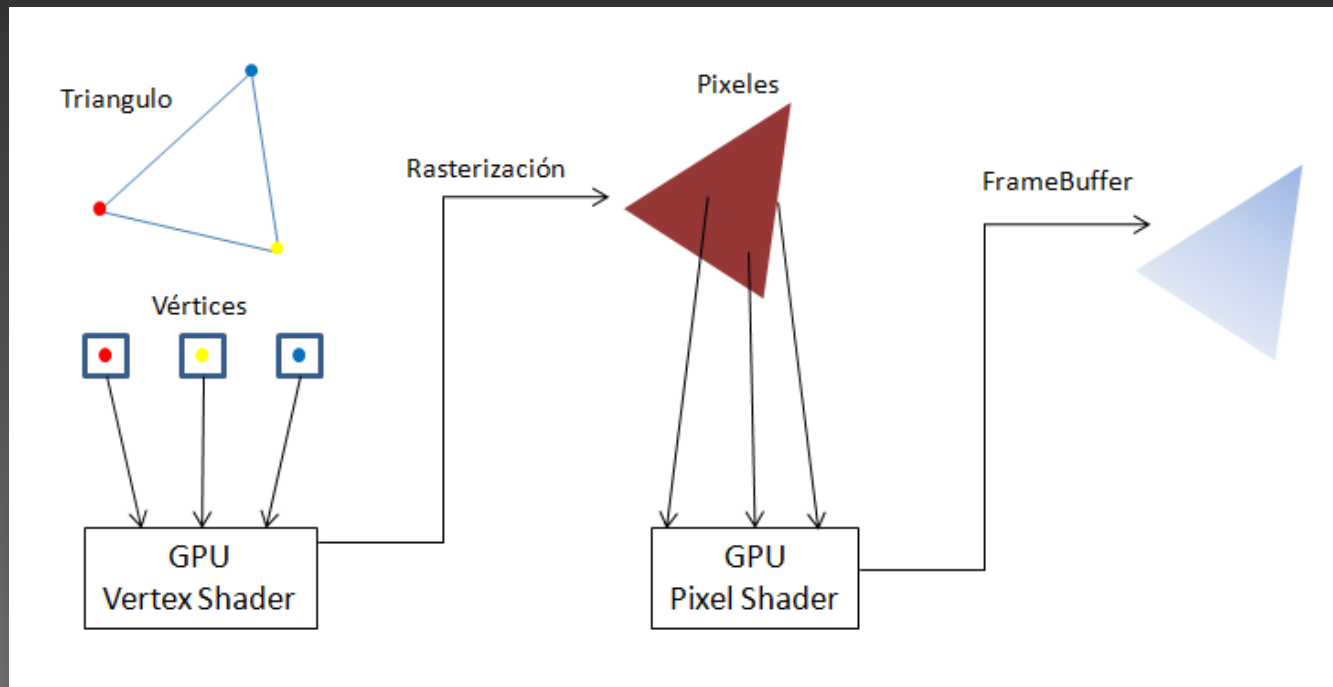
Bump Mapping (Superior)

Depth of Field (Derecha)

Efectos visuales que solamente se podían aplicar para producciones cinematográficas, se empiezan aplicar al mundo de los videos juegos

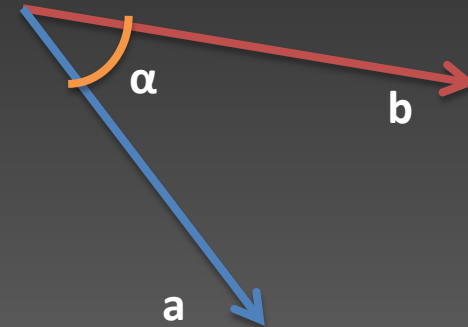
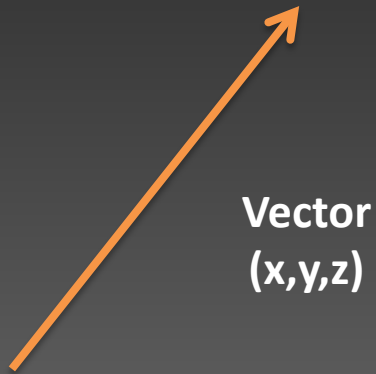


Pipeline Gráfico

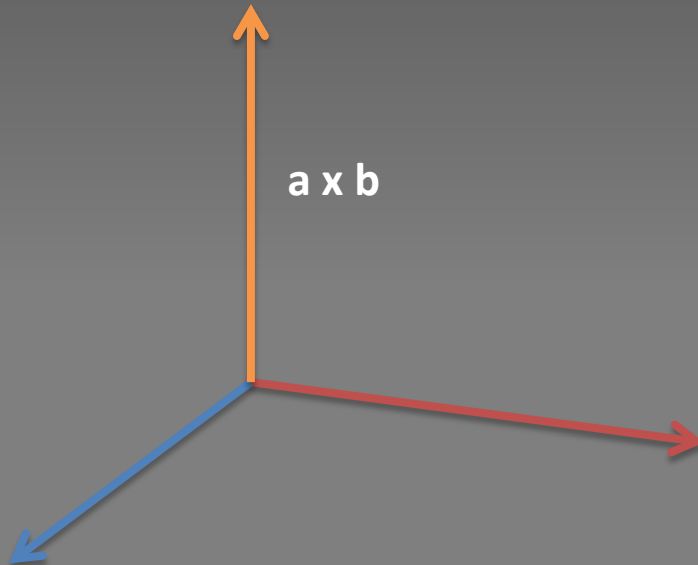


- El **Vertex Shader** es un código que suministra el programador, este shader tiene información del vértice (coordenada del vértice, normales, tangente, binormal, etc), el programador captura esta información y procede hacer algún tipo de cálculo con él.
- El **Pixel Shader**, es un código que tiene como entrada la información de un píxel (coordenada U y V, información de la textura, etc), este píxel es el resultado del proceso de rasterización de la tarjeta gráfica, y el programador en su código tiene que dar el color al del píxel.

Aspectos Básicos



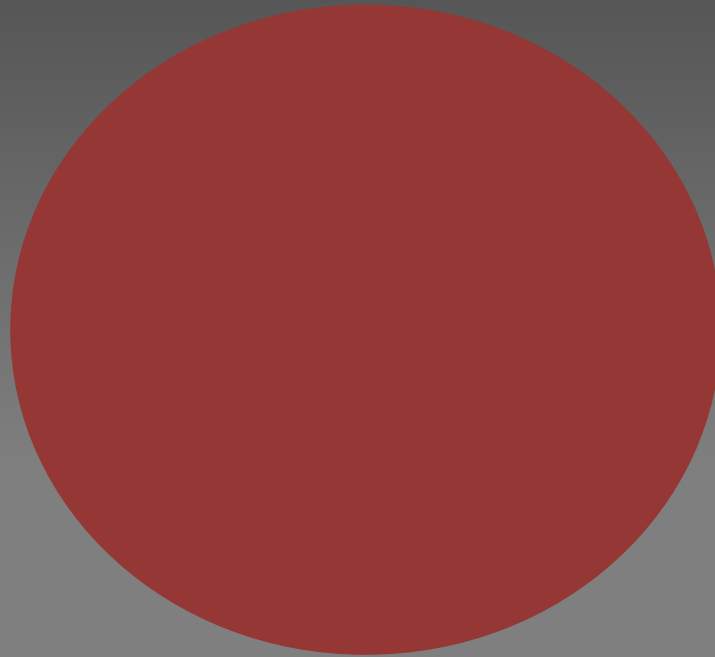
$$a \cdot b = ||a|| ||b|| \cos(\alpha)$$



Conocimientos básicos de Geometría y Algebra Lineal son necesario para entender ciertos efectos visuales

Color Ambiental

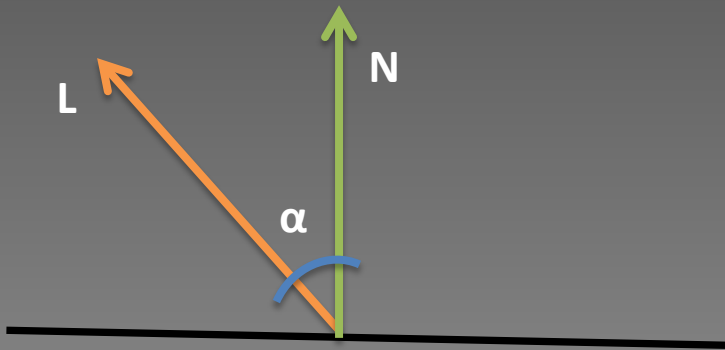
$$\text{Color_pixel} = k_a * c_a;$$



Color Difuso

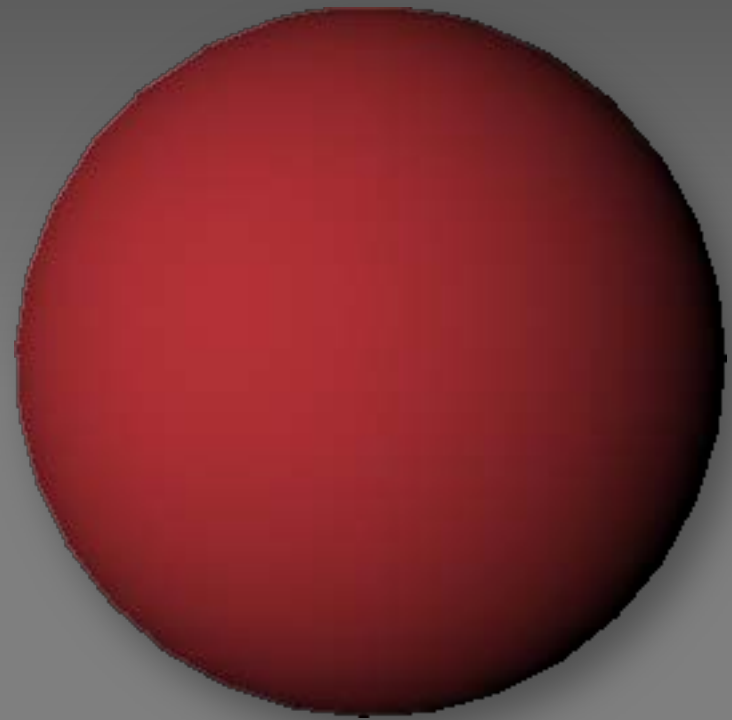
$$\text{Color_pixel} = k_d * \text{Cos}(\alpha)$$

Producto Interno $\text{Color_pixel} = k_d * (L \cdot N) * c_d$



L = dirección de la luz

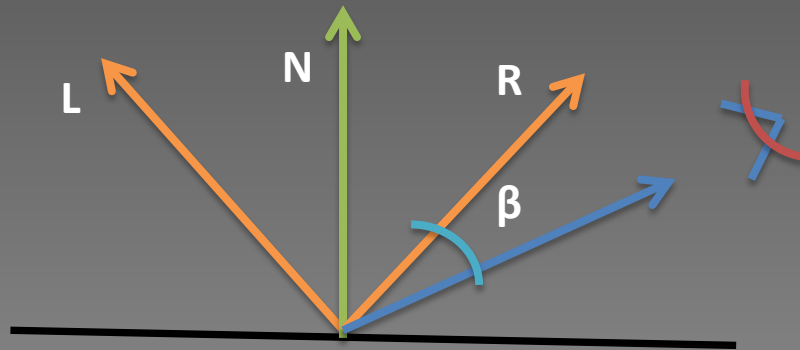
N = Vector normal del punto



Color Especular (Phong)

$$\text{Color_pixel} = k_s * \text{Cos}(\beta)$$

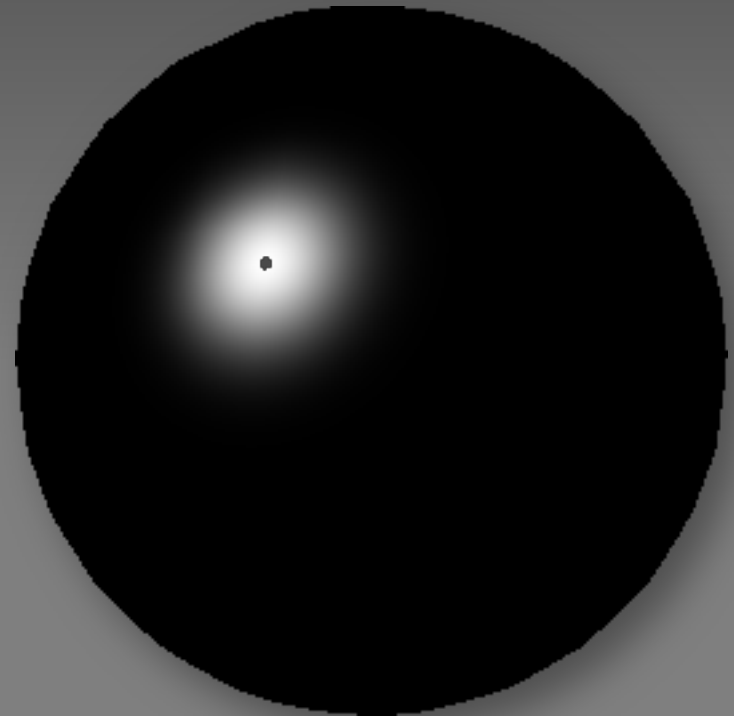
Producto Interno $\text{Color_pixel} = k_s * (V \cdot R)^n * c_s$



V = dirección de la cámara

R = vector reflexión de la luz

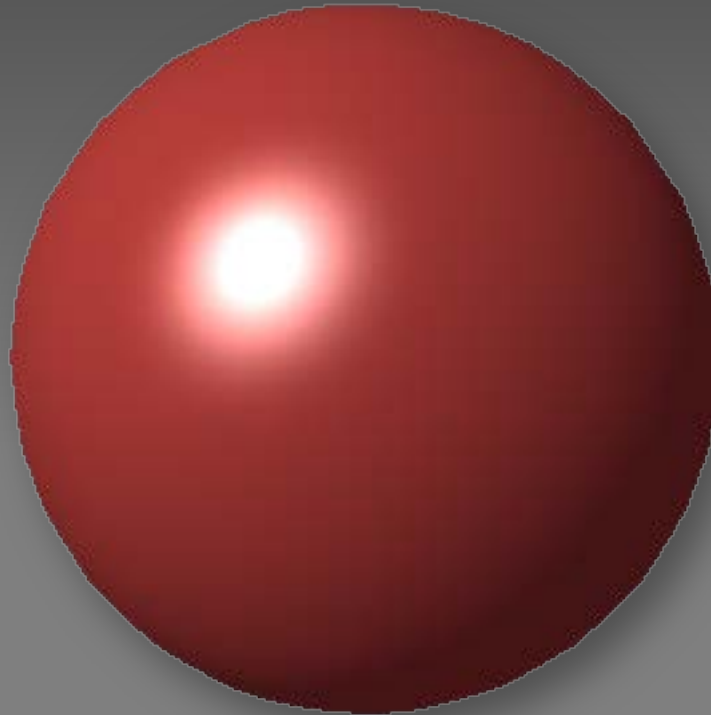
R = Ecentricidad del brillo especular



Modelos de Sobreados

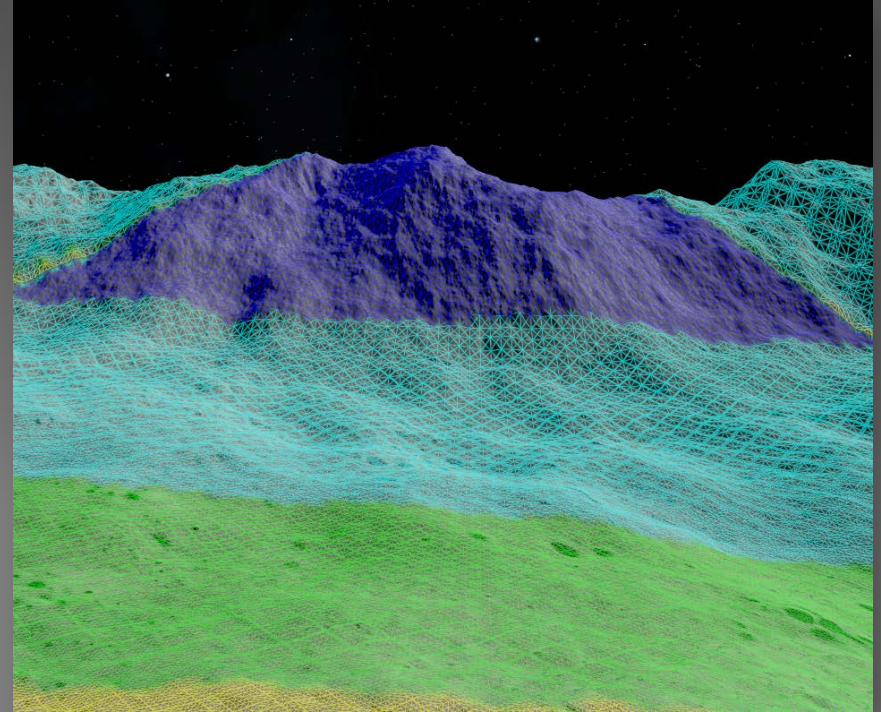
Color Ambiental + Difuso + Especular

$$\text{Color_pixel} = k_a * c_a + k_d * (L.N) * c_d + k_s * (V . R)^n * c_s$$



Actualidad

A partir de la versión Directx 10 y Opengl 3.2 se introduce el **Geometry Shader**, que permite la creación de nuevos vértices y por ende de primitivas en el GPU. Este **Geometry Shader** se ejecuta después del Vertex Shader.



GRACIAS...