

Laboratorio 2 – ILUMINACIÓN Y SOMBREADO OPENGL

Windows Presentation Foundation

OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por Silicon Graphics Inc. (SGI) en 1992 y se usa ampliamente en CAD, realidad virtual, representación científica, visualización de información y simulación de vuelo. También se usa en desarrollo de videojuegos, donde compete con Direct3D en plataformas Microsoft Windows.

Este laboratorio se realizará bajo tecnología WPF Windows Presentation Foundation, que es una tecnología de Microsoft, presentada como parte de Windows Vista. Permite el desarrollo de interfaces de interacción en Windows tomando características de aplicaciones Windows y de aplicaciones web.

WPF ofrece una amplia infraestructura y potencia gráfica con la que es posible desarrollar aplicaciones visualmente atractivas, con facilidades de interacción que incluyen animación, vídeo, audio, documentos, navegación o gráficos 3D. Separa, con el lenguaje declarativo XAML y los lenguajes de programación de .NET, la interfaz de interacción de la lógica del negocio, propiciando una arquitectura Modelo Vista Controlador para el desarrollo de las aplicaciones.

Microsoft Silverlight es un subconjunto de WPF que permite crear aplicaciones similares a Flash para web y móviles, con el mismo modelo de programación que las aplicaciones .NET. No admite funciones 3D, pero soporta XPS y planos basados en vectores.

Para unir OpenGL con WPF, utilizaremos la librería SharpGL que encapsula a OpenGL. Para agregar esta librería a nuestro proyecto utilizaremos el gestor de paquetes NuGet.

¿Cómo Funciona OpenGL?

OpenGL es una librería que trabaja como una máquina de estados. ¿pero qué significa? La utilización de OpenGL consiste en activar y desactivar opciones, y realizar ciertas acciones, que tendrán como fruto una representación en pantalla (o no) de una serie de datos, dependiendo en el estado en que nos encontremos. Así, no será lo mismo dibujar un triángulo y activar una textura, que activar una textura y dibujar un triángulo en OpenGL, el orden de las acciones resulta crítico en la mayoría de las ocasiones de igual manera.

De este modo, de manera muy abstracta, la manera de dibujar algo en OpenGL suele ser la siguiente:

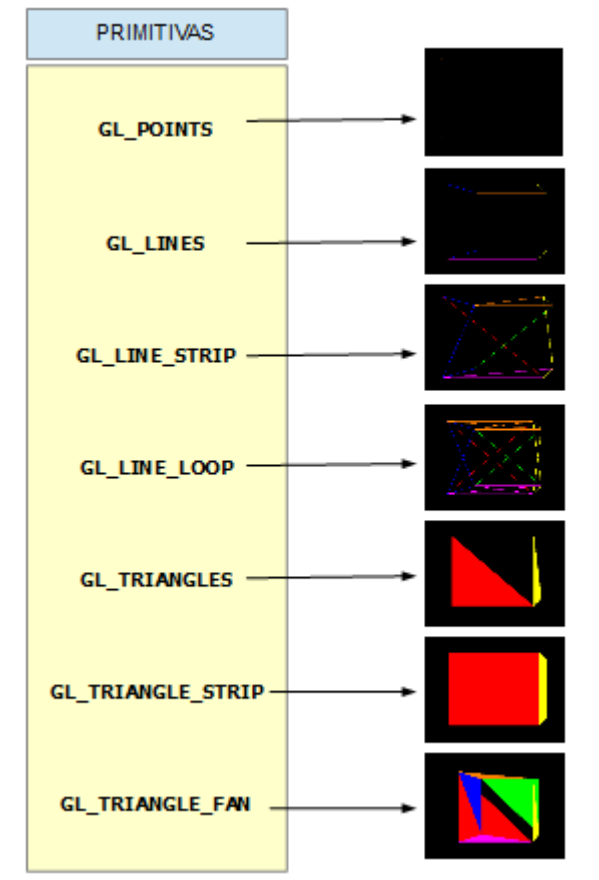
1. Activar todas las opciones que van a ser persistentes a la escena (ponemos la cámara, activamos la iluminación global).

2. Activar las opciones que establecen el estado de un objeto específico (su posición en el espacio, su textura).
3. Dibujar el objeto.
4. Desactivar las opciones propias de ese objeto (volver a la posición original, desactivar su textura)
5. Volver al punto 2 hasta haber dibujado todos los objetos.

Tipos de Primitivas

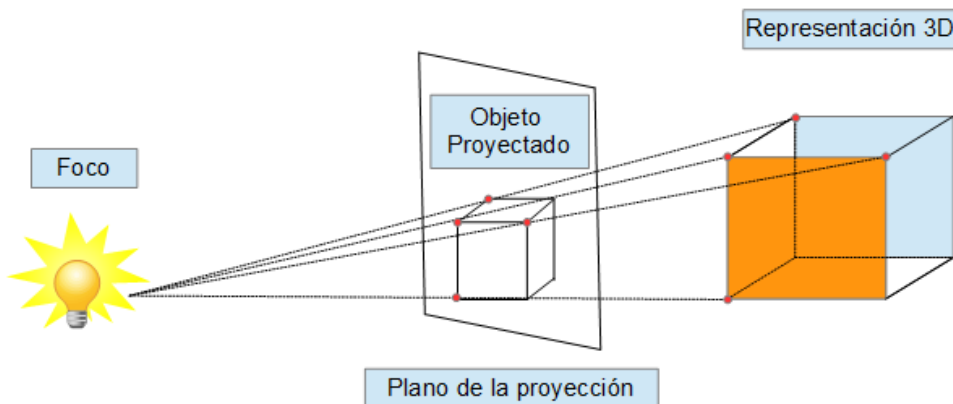
En OpenGL podemos encontrar las siguientes primitivas, que nos ayudaran a dibujar nuestros polígonos.

Primitiva	Descripción
GL_POINTS	Dibuja los vértices como puntos individuales.
GL_LINES	Cada par de vértices formaría una línea individual.
GL_LINE_STRIP	Cada vértice está unido con el anterior formando segmentos conectados.
GL_LINE_LOOP	Al igual que GL_LINE_STRIP, pero implementando un segmento más entre el primer y último vértice.
GL_TRIANGLES	Tres vértices son interpretados como tres triángulos
GL_TRIANGLE_STRIP	Cada vértice se une con los dos anteriores para formar un triángulo.
GL_TRIANGLE_FAN	Cada vértice se une con los dos primeros para formar un triángulo.



La proyección: Objetos 3D en superficie 2D.

La proyección de una escena, es aquella que permite representar objetos definidos en 3D en una superficie 2D. Esta proyección, se consigue trazando líneas imaginarias, desde un punto fijo llamado foco, a cada uno de los puntos del objeto 3D que se representa.



Tipos de proyección:

- Paralela: El foco se sitúa en el infinito, siendo las líneas de proyección trazadas paralelas. Si el plano de proyección se sitúa perpendicular a las líneas de proyección se denomina proyección ortogonal.
- Perspectiva: El foco se sitúa en un punto concreto, por lo que las líneas de proyección trazadas no serán paralelas. Si el objeto está cercano al foco se representará más grande, dando una sensación de profundidad. Esta opción es una de las más usadas en gráficos 3D.

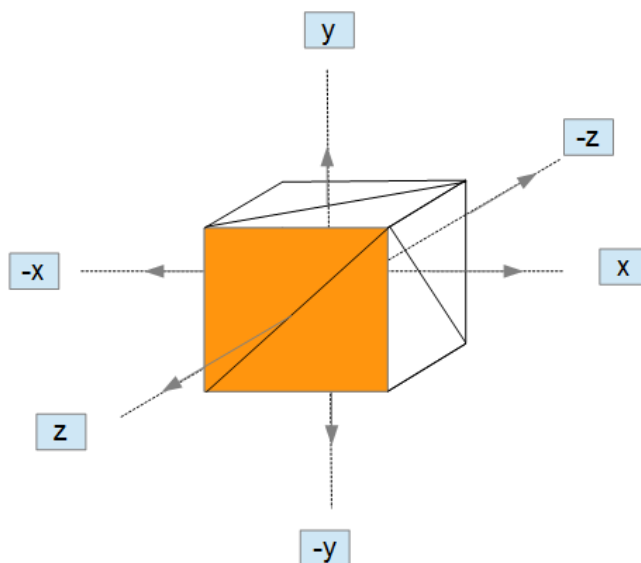
OpenGL se encarga de realizar todos los cálculos necesarios para la realización de dicha proyección, con lo que nos evita realizarla manualmente.

Iluminación en objetos 3D

En el siguiente punto, nos centraremos en entender como iluminar una escena, estableciendo un foco, para reflejar luz sobre un objeto 3D, mostrando los diferentes efectos de sombra.

En cualquier objeto 3D representado, lo habitual es que alguna de las caras que forman el objeto, queden a la sombra, y por lo tanto se vean más oscuras. Para ello OpenGL divide el proceso de iluminación en tres partes bien diferenciadas, para dar mucho más realismo a las formas construidas:

- Definir los vectores normales de cada cara: En este punto será necesario calcular la luz reflejada con respecto al ángulo que se forme con el foco de luz. Para ello es necesario definir un vector normal (perpendicular a la superficie apuntando hacia fuera de la parte visible) por cada uno de los vértices de nuestra representación. Por lo tanto, es necesarios definir 6 vectores normales por cada una de las caras de un cubo.



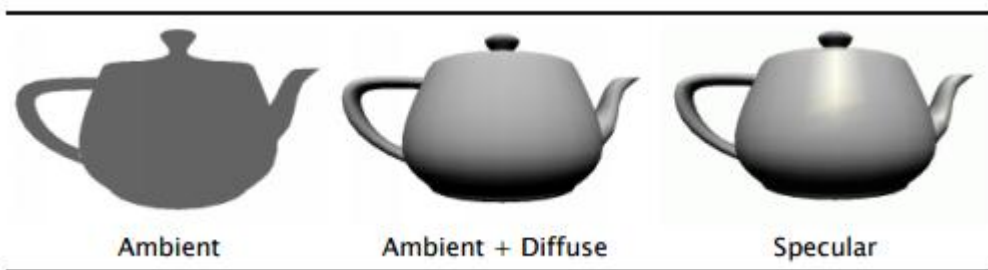
- Situar las luces: Para iluminar una escena será necesario situar las luces. OpenGL maneja dos tipos de iluminación:
 - Luz ambiental: ilumina toda la escena por igual, ya que esta no proviene de una dirección determinada.
 - Luz difusa: Viene de una dirección específica, y depende de su ángulo de incidencia para iluminar una superficie en mayor o menor medida.
- Definiendo materiales: OpenGL permite controlar la forma en que la luz se refleja sobre nuestros objetos, que es lo que se conoce como definición de materiales.

Modelo de Iluminación Phong

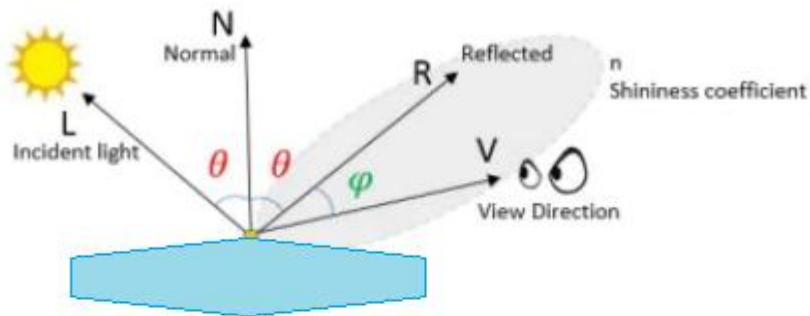
En la programación de gráficos 3D existen distintos modelos de iluminación que intentan simular cómo se comporta la luz al iluminar la superficie de un objeto tridimensional, uno de esto es el modelo de iluminación Phong, lo podemos aplicar por vértice (Gouraud shading) o por pixel (Phong shading).

El modelo de iluminación phong utiliza tres componentes básicos:

- Ambiente: La luz que llega rebotada de las paredes, los muebles, etc., se refleja en todas las direcciones simultáneamente.
- Difusa: La luz que llega directamente desde la fuente de luz pero rebota en todas direcciones, combinada con la luz ambiental definen el color del objeto.
- Especular: La luz que llega directamente de la fuente de luz y rebota en una dirección, según la normal de la superficie. Es la que afecta al "brillo" de la superficie.



Ecuación para el cálculo de la iluminación en un punto:



$$I_p = \text{ambient} + \text{diffuse} + \text{specular}$$

$$\text{ambient} = k_a * \text{ambientColor}$$

$$\begin{aligned} \text{diffuse} &= k_d * \text{lightColor} * \cos(\theta) \\ &= k_d * \text{lightColor} * \max(0, N \cdot L) \end{aligned}$$

$$\begin{aligned} \text{specular} &= k_s * \text{lightColor} * \cos(\varphi)^n \\ &= k_s * \text{lightColor} * \max(0, R \cdot V)^n \end{aligned}$$

Desarrollo de Laboratorio

Se implementará un modelo 3D, en el cual, analizaremos sus factores de luminosidad y sombras.

1. Paso 1 (20 puntos)

Se debe Crear una aplicación WCF, el cual debe tener un 6 checkbox como aparece en la siguiente imagen. Y un control OpenGL.



2. Paso 2 (20 puntos)

Se debe dibujar una figura 3D, esta figura debe girar en su eje Y.



3. Paso 2 (20 puntos)

Cada uno de los CheckBox, deben manejar los patrones de luminosidad, de Luz Ambiental Simple y Global y Posición de Luz, Además las que corresponde al Modelo de Reflexión Phong como son Luz Ambiental, Luz Difusa y Luz Especular.

