

# CURSO: CC322 - 2017 -2

## Practica Lab 11

### 1. Introducción

Objetivo general:

-

### 2 . Recursos Informáticos

<https://www.opengl.org>

[https://www.khronos.org/opengl/wiki/Getting\\_Started](https://www.khronos.org/opengl/wiki/Getting_Started)

[https://www.khronos.org/opengl/wiki/Category:Core\\_API\\_Reference](https://www.khronos.org/opengl/wiki/Category:Core_API_Reference)

### 3. DESARROLLO

1. Verifique si tiene instalado un compilador de código en C (cpp, g++)  
Sino, proceda a instalarlo en su PC
2. Verifique si tiene un editor de texto adecuado para editar programas en C  
(p.ej. Geany, SublimeText, u otro con el que esté familiarizado) si no está – Proceda a Instalarlo.
3. Verifique si tiene instaladas las librerías OPENGGL Y GLUT. Si no están proceda a instalarlas.  
sudo apt-get install freeglut3-dev
4. Al final entregara un archivo con el nombre CC322\_Lab11\_<Nombre\_apellido>.zip con los archivos generados en la practica.

### 4 . Ejercicio

Este ejercicio demuestra el uso de la iluminación junto con las propiedades de reflectancia superficial.

Se requieren cuatro pasos en OpenGL para iluminar una escena:

- (1) Definir vectores normales para cada vértice
- (2) Configurar las luces (ubicación, dirección, tipo)
- (3) Seleccionar un modelo de iluminación
- (4) Habilitar la eliminación de las superficies ocultas.
- (5) Definir las propiedades materiales de los objetos.

Una vez hecho esto, OpenGL puede renderizar la escena iluminada en base a la apariencia de la superficie, bajo las condiciones dadas de iluminación.

( I ) Compilar y ejecutar el programa teapot.c

Esto muestra una tetera sombreada sin ninguna iluminación de la escena, que aparece como una silueta de color contra el fondo blanco.

(II ) Las luces OpenGL se especifican usando `glLight * ( )` que especifica el tipo de fuente de luz y las propiedades correspondientes como posición, dirección del color. El comando se especifica como sigue :

`glLight * (light, type, param );`

light - es el número de la luz `GL_LIGHT0-8`

type - es el tipo de iluminación ambient, diffuse, specular, position, spot, .....

param - conjunto de parámetros asociados para un tipo particular de luz

Añada una única luz de posición y active la *iluminación de la siguiente* manera en la función `initgl ( )` :

`glLightfv (GL_LIGHT0, GL_POSITION, light0_position);`

`glEnable (GL_LIGHTING);`

`glEnable (GL_LIGHT0);`

Y definir la posición de la luz como una variable estática global:

`Static GLfloat light0_position [] = {1.0,1.0,1.0,0.0};`

Compilar y ejecutar el programa muestra el efecto de la iluminación en la superficie sombreada mate del modelo de tetera. Sin embargo, el modelo no está sombreado directamente ya que el sombreado no tiene en cuenta el orden de las superficies en relación con el espectador . Por ejemplo, el borde de la La tetera estará sombreada incorrectamente ya que se ve la tapa que esta oculta de la vista.

(III) La eliminación de la superficie oculta hace que la escena sea tal que la superficie más cercana que aparece en la imagen, sea la más cercana al observador, es decir, la superficie visible. En OpenGL esto se consigue utilizando un "buffer de profundidad", para cada pixel procesado el buffer de profundidad almacena la distancia a la superficie. Si una nueva superficie está más cerca del espectador entonces el valor del pixel es reemplazado de otra manera, se descarta.

Para habilitar el "depth buffering" en OpenGL y GLUT debemos inicializar el buffer de profundidad con el Siguiente comando en `main ( )`:

`glutInitDisplayMode (... | ... | GLUT_DEPTH);`

Esto inicializa un buffer de profundidad junto con otros buffers es decir `GLUT_RGB`, `GLUT_SINGLE / DOUBLE`

El uso del buffer de profundidad debe habilitarse en OpenGL en `initgl ( )` mediante:

`glEnable (GL_DEPTH_TEST);`

Finalmente, cuando limpiamos la pantalla, el buffer de profundidad también debe ser borrado:

`glClear (GL_COLOUR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);`

Implementar buffer de profundidad, compilar y ejecutar el programa.

La tetera debe aparecer ahora correctamente sombreada con un acabado mate.

(IV) Para cambiar la apariencia de la superficie de la tetera podemos ajustar las propiedades superficiales tales como Como el brillo. Agregue las siguientes propiedades de superficie y vuelva a mostrar la tetera.

`static GLfloat mat_specular [] = {1.0,1.0,1.0,1.0};`

`static GLfloat mat_shininess [] = {50.0};`

`glMaterialfv (GL_FRONT, GL_SPECULAR, mat_specular );`

`glMaterialfv (GL_FRONT, GL_SHININESS, mat_shininess );`

Esto ahora debe producir una tetera mucho más realista.

Trate de girar la tetera alrededor del eje y. (utilice un par de teclas para ello)

(V) Cambiar el color de la superficie para ver el efecto combinado de iluminación, color y reflectancia. Para ello, no sólo debe configurar el color del objeto como antes, sino también habilitar el color del material

En `initgl ( )` habilitar el color del material:

```
GLenum (GL_COLOR_MATERIAL);
```

Ahora *especifique* el color de la tetera como antes usando `glColor3f ( )`

Mediante la función `glColorMaterial ( )` puede aplicar las propiedades del material.

Para aplicar el color sólo a los polígonos del frente iluminados para el componente de reflectancia ambiental especificar:

```
glColorMaterial (GL_FRONT, GL_DIFFUSE);
```

**Nota:** Esto especifica el modo para aplicar el color del material - sólo se puede especificar un modo a la vez. Se requieren llamadas repetidas para establecer diferentes colores para las caras delanteras y traseras si las caras traseras son visibles.

(VI) Ampliar el programa para controlar dinámicamente la posición de la luz, la dirección de la vista y las propiedades de la superficie utilizando la entrada del teclado. Note que la transformación de fuentes de luz se trata como con otras Primitivas geométricas utilizando la matriz `modelview` .

(VII) Experimentar con otras propiedades de los materiales de superficie ambientales, difusas. Especular , emisión capture la pantalla de cada variante.

Comprima los archivos resultantes y suba el archivo comprimido **<Su Nombre->Lab11.zip** al repositorio del curso.