

# **Sistemas de Desarrollo de Programas OpenGL en Linux**

Dr. Luis Gerardo de la Fraga

E-mail: [fraga@cs.cinvestav.mx](mailto:fraga@cs.cinvestav.mx)

Sección de Computación

Departamento de Ingeniería Eléctrica

CINVESTAV-IPN

5 de noviembre de 2001

## **Motivación**

Dar a conocer programas de uso público, gratuitos para instituciones públicas y universidades, para la enseñanza y desarrollo de la computacion bajo el sistema operativo Linux.

## **Objetivos**

1. Introducción a OpenGL (Mesa). Cuerpos 3D.
2. El sistema de desarrollo de interfaces gráficas con Qt.
3. Ejemplo de realización de un programa con Qt y OpenGL

## OpenGL y Mesa

OpenGL es una interface de software (API, en inglés) para el hardware gráfico. La interface consiste de un conjunto de varios cientos de procedimientos y funciones que permiten al programador producir imágenes en color de alta calidad especialmente de objetos 3D.

OpenGL .- **O**pen **G**raphics **L**ibrary. OpenGL(R) es una marca registrada de Silicon Graphics, Inc. (SGI)

**Mesa** es una librería gráfica 3D con un API que es muy similar a la de OpenGL. Mesa utiliza la misma sintaxis de los comandos de OpenGL (autorizado por SGI).

OpenGL es independiente del Sistema Operativo.

## Documentación

El sitio primario de *Mesa* es [www.mesa3d.org](http://www.mesa3d.org)

Ya que Mesa usa la misma API de OpenGL y mimetiza su semántica, la misma documentación de OpenGL sirve como documentación de las funciones del núcleo de Mesa. Checar [www.opengl.org](http://www.opengl.org)

## Funcionalidad de OpenGL

OpenGL solo dibuja un pequeño grupo de primitivas: puntos, líneas, triángulos y cuadriláteros. Provee las funciones para el

1. Modelado geométrico
2. Transformaciones geométricas
3. Color
4. Iluminación
5. Sombreado

OpenGL permite también características avanzadas como

- Mapeo de textura
- Niebla
- Emborronado por movimiento.

## **Programa “Hola Mundo” en OpenGL**

1. Un procedimiento para inicializar OpenGL (initializeGL)
2. Un procedimiento que maneje el evento de cambio de tamaño de la ventana (resizeGL).
3. Un procedimiento con las funciones para pintar la escena (paintGL).



## Qt (1)

Qt es una caja de herramientas para el desarrollo de Interfaces Gráficas de Usuario (GUI, del inglés Graphical User Interface) multiplataforma. Qt es un producto de Troll Tech ([www.trolltech.com](http://www.trolltech.com)). Qt es soportado por todas las variantes de Microsoft Windows y Unix/X Windows. Es distribuido bajo una licencia GPL, lo que significa que para el desarrollo de software público no es necesario comprar una licencia para su uso. Si se quiere tener software propietario es necesario comprar una licencia.

## Qt (2)

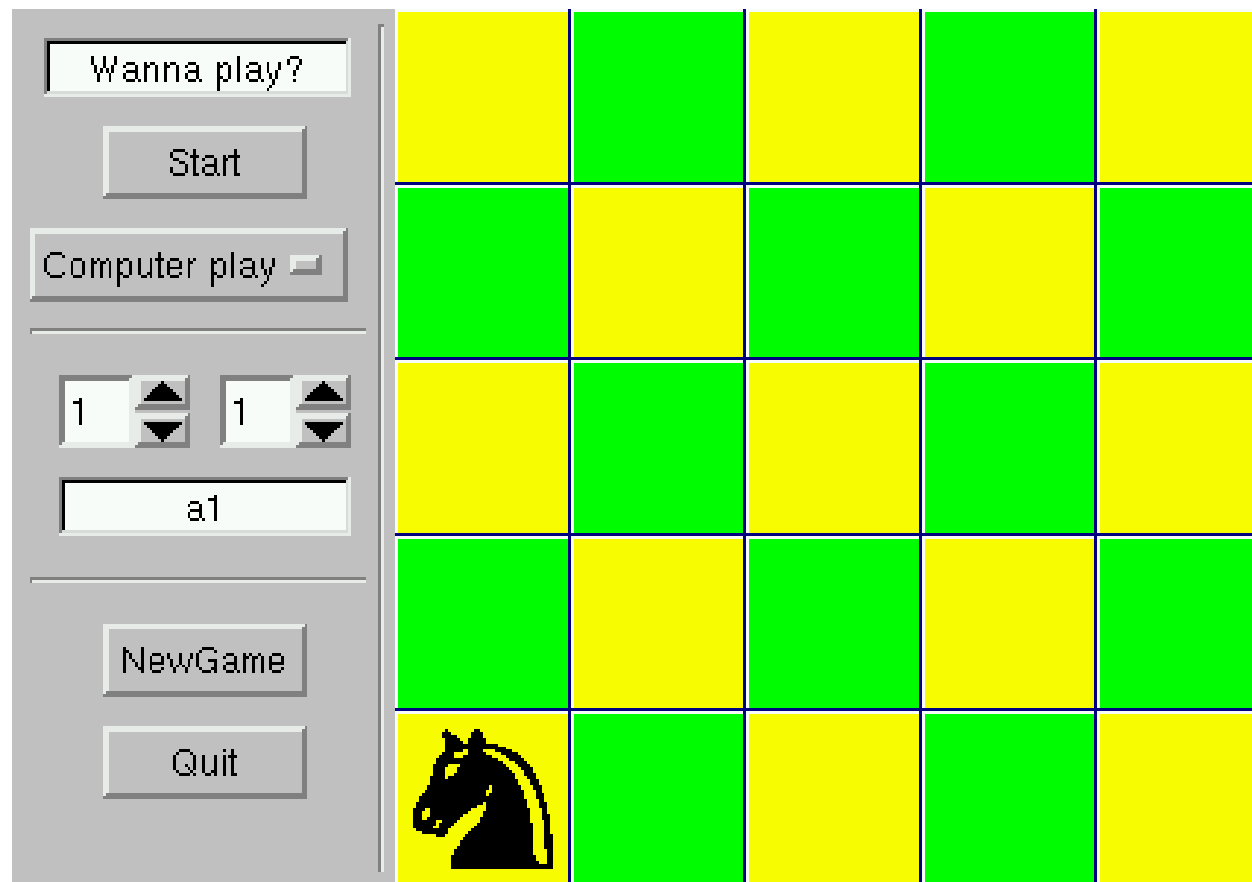
Qt permite el desarrollo de prototipos rápidos y es gratuito bajo Linux, aparte de que tiene una documentación excelente.

Qt introduce el mecanismo de señales–ranuras para el intercambio de mensajes entre objetos que es bastante intuitivo, en comparación al mecanismo de retrollamadas del desarrollo en **X Windows** y **Motif**. Las señales son enviadas por un objeto a la ranura de otro. Este mecanismo provoca un sobreuso del procesador porque hay que usar un metacompilador (llamado MOC, Meta Object Compiler, en Qt) para traducir el código de C++ con señales y ranuras a código C++ simple.

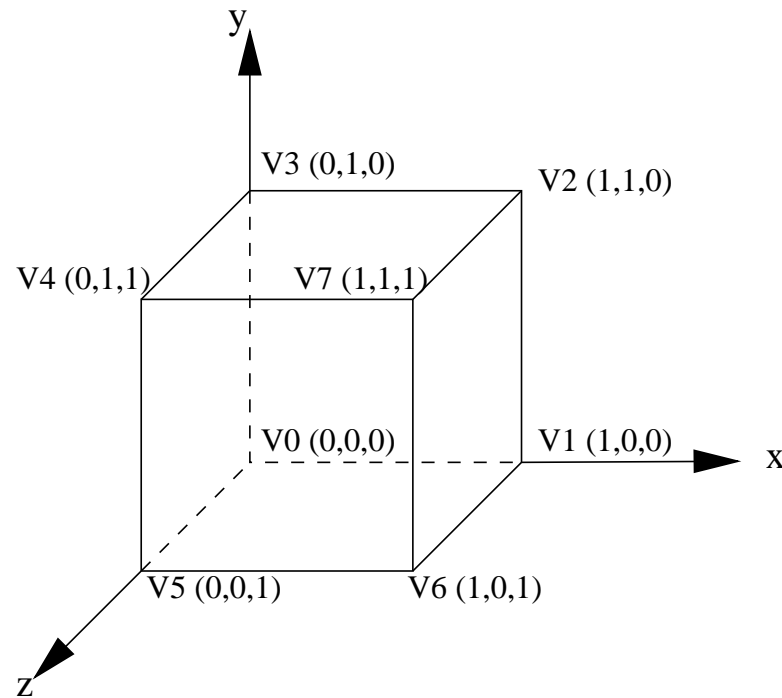
## Interfaz gráfica

Los componentes para diseñar una interfaz gráfica (GUI, en inglés) se llaman *widgets*. Widgets comunes son: botones, barras de scroll, cuadros de diálogo, menús, botones con imágenes (o iconos), ventanas, etc.

La librería de Qt provee toda la funcionalidad para el dibujo de widgets, sus estilos de dibujo, y para controlar los eventos de teclado y ratón, además de poder diseñar nuevos widgets inexistentes, como podría ser una manija circular (como las que controlan el volumen en un radio). El programador se enfoca a la funcionalidad y la comunicación entre los diferentes widgets.



## Ejemplo 1. Cubo Unitario



6 vértices, 6 caras:

$$V = (V_0, V_1, V_2, V_3, V_4, V_5)$$

$$V = ((x_0, y_0, z_0), (x_1, y_1, z_1), \dots, (x_5, y_5, z_5))$$

$$C_1 = (0, 1, 2, 3)$$

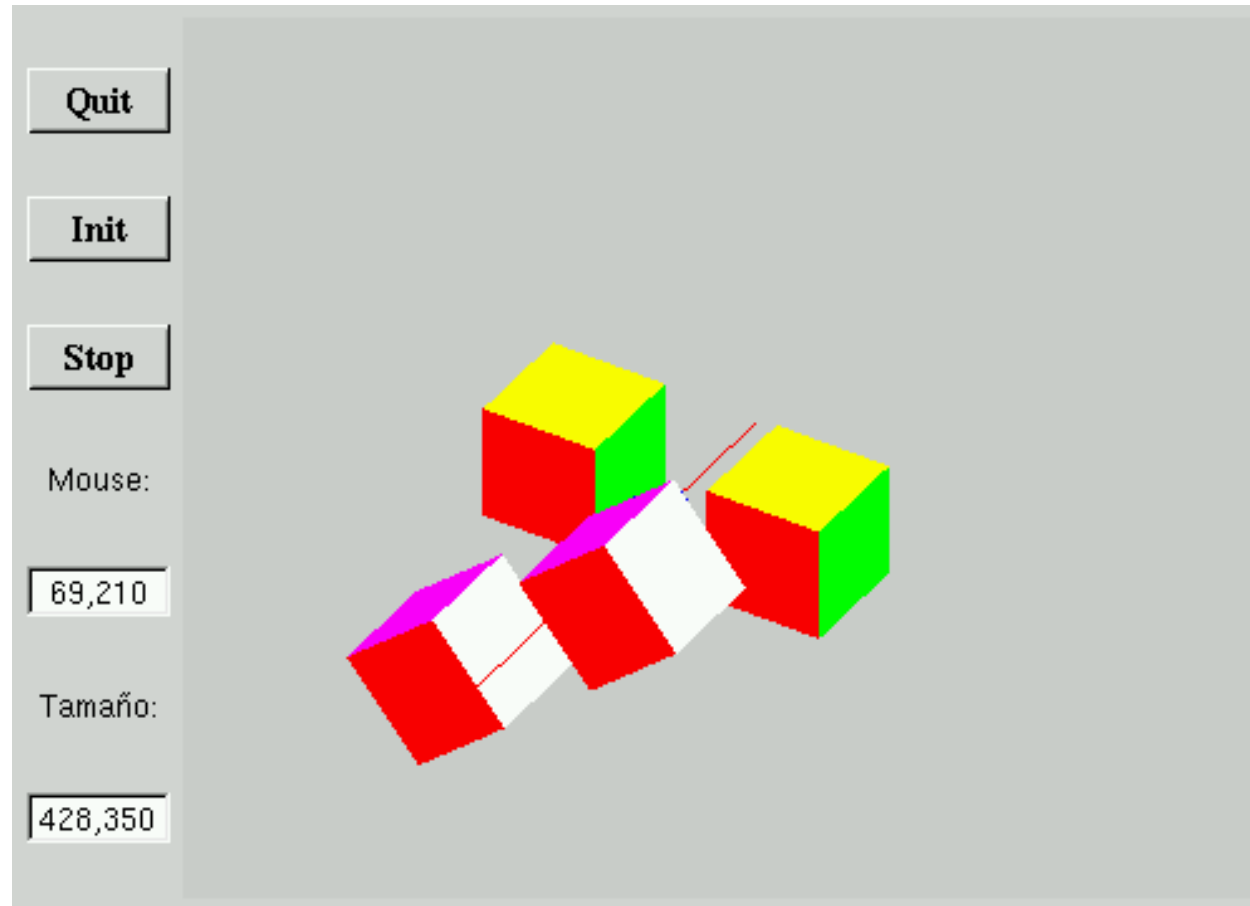
$$C_2 = (5, 4, 7, 6)$$

$$C_3 = (1, 6, 7, 2)$$

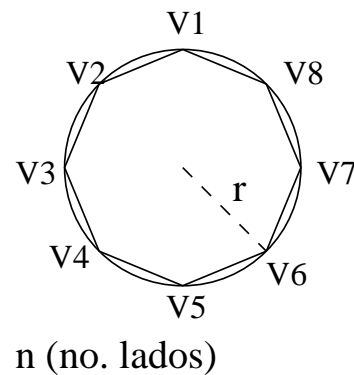
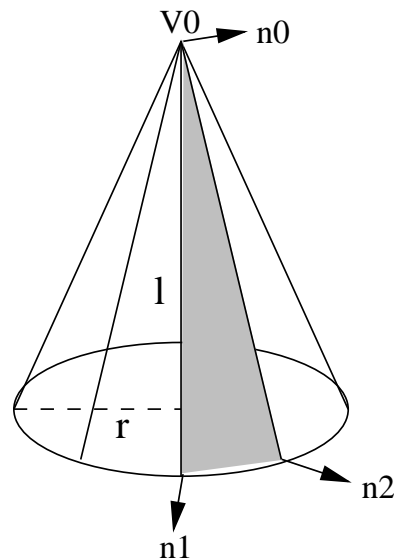
$$C_4 = (0, 3, 4, 5)$$

$$C_5 = (3, 2, 7, 4)$$

$$C_6 = (0, 5, 6, 1)$$



## Ejemplo 2. Conos



$n + 1$  vértices,  $n$  caras:

$$V = (V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8)$$

$$V = ((x_0, y_0, z_0), (x_1, y_1, z_1), \dots, (x_8, y_8, z_8))$$

$$C_1 = (1, 0, 2)$$

$$C_2 = (2, 0, 3)$$

$$C_3 = (3, 0, 4)$$

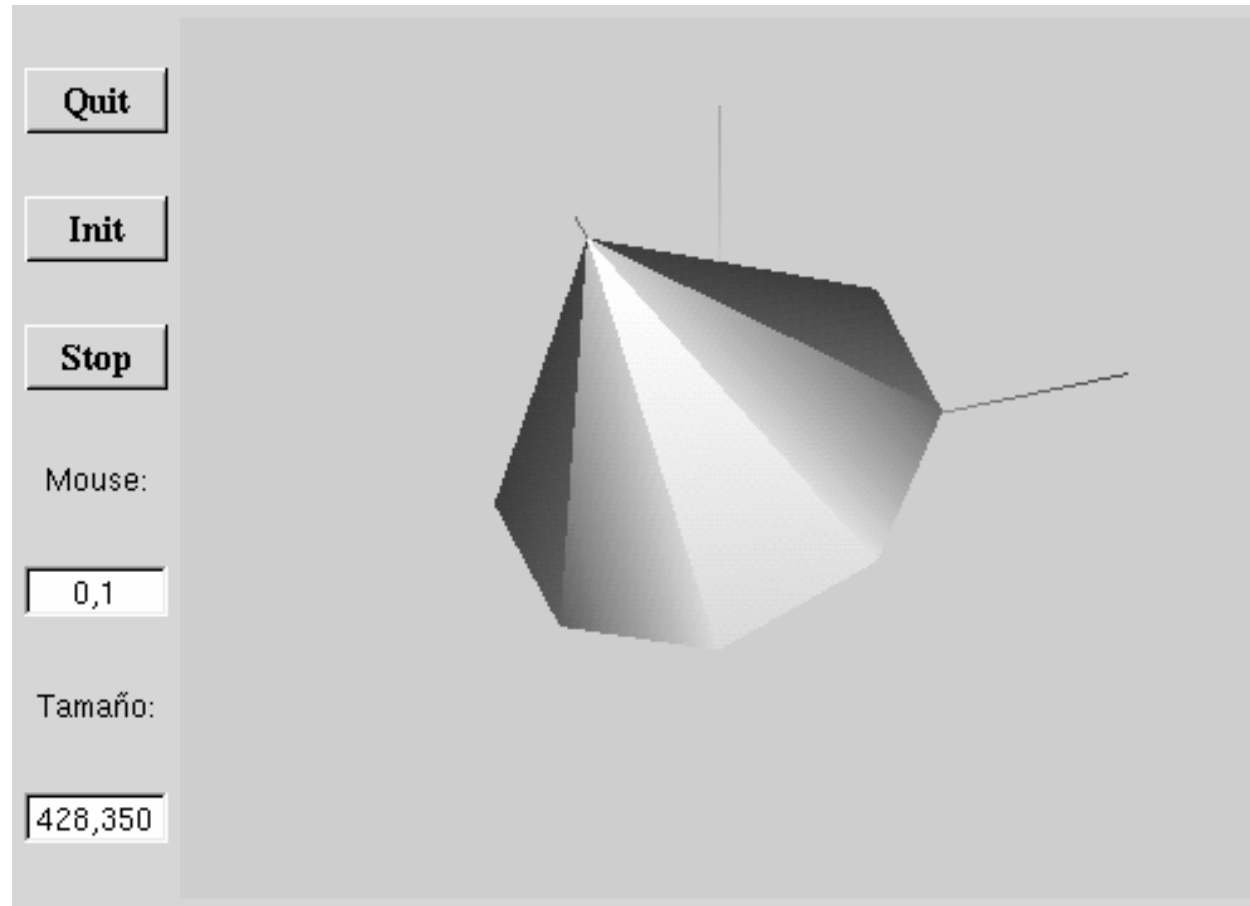
$$C_4 = (4, 0, 5)$$

$$C_5 = (5, 0, 6)$$

$$C_6 = (6, 0, 7)$$

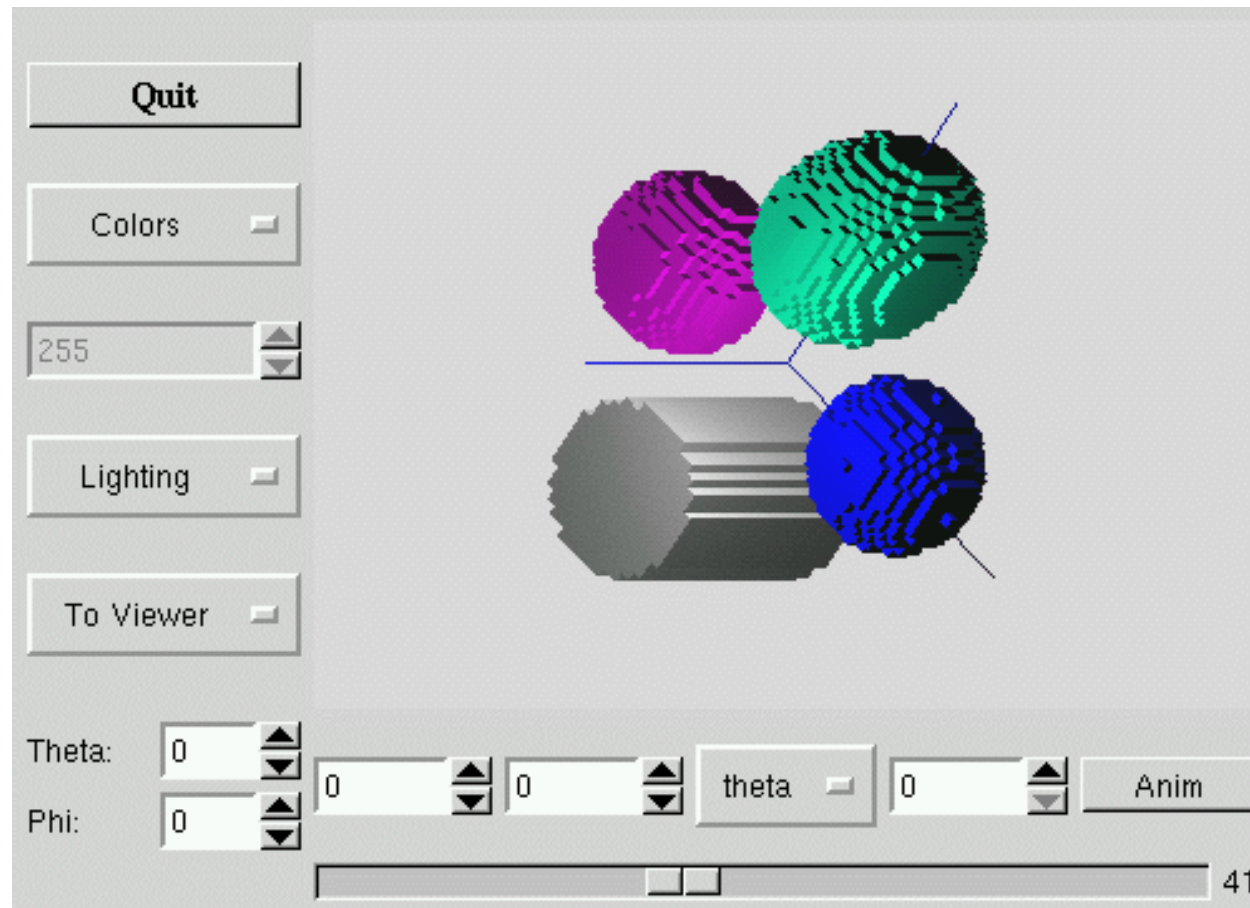
$$C_7 = (7, 0, 8)$$

$$C_8 = (8, 0, 1)$$





<http://www.cs.cinvestav.mx/Scubes>



## Conclusiones

Linux es una plataforma donde pueden desarrollarse y probarse software de desarrollo, no sólo de visualización, cuyo costo es nulo para universidades públicas.

Mesa y Qt representan una excelente combinación para el desarrollo de programas de visualización 3D profesionales. Mesa pone el API OpenGL y Qt pone las facilidades de la interfaz gráfica.

Las transparencias de esta charla estarán disponibles en

Internet en la URL:

<http://delta.cs.cinvestav.mx/~fraga>