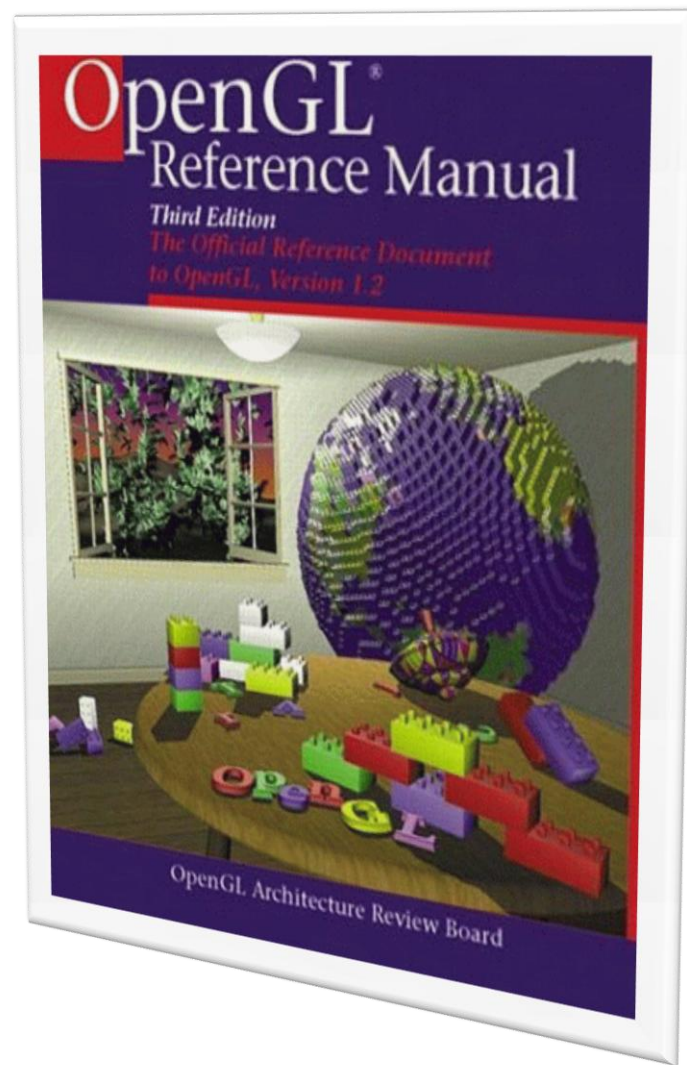
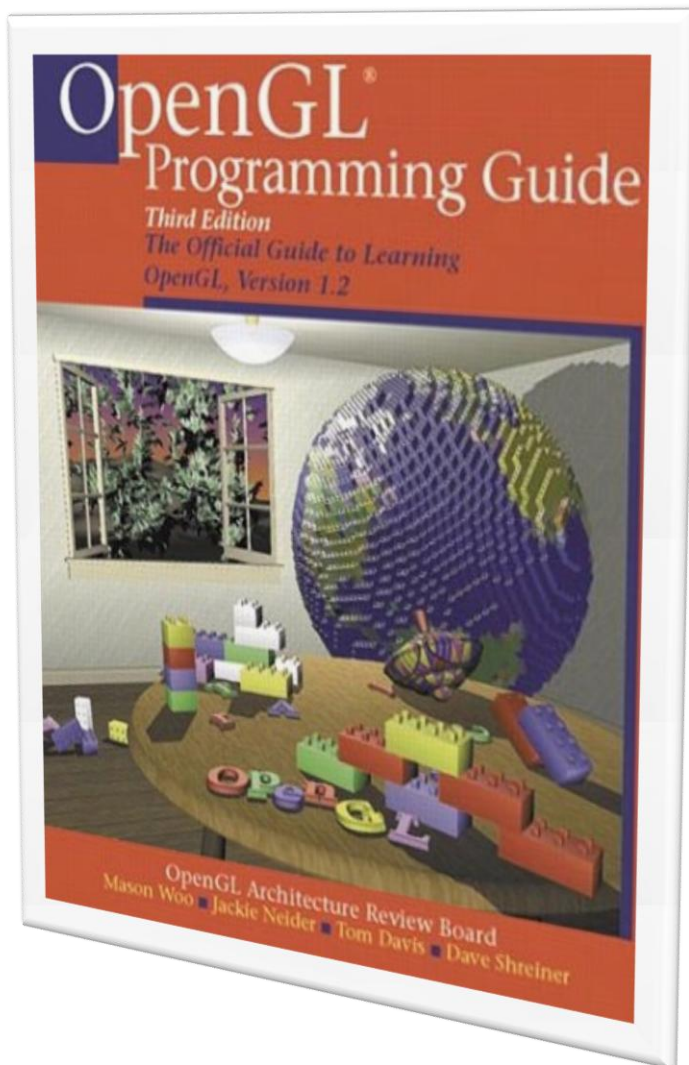


¿Qué es OpenGL?

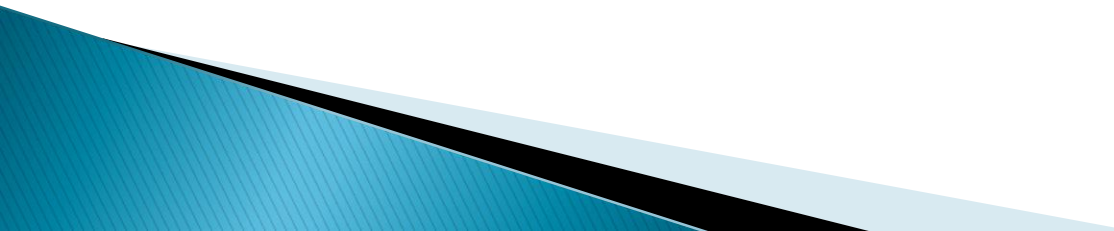


- ▶ OpenGL es una biblioteca para el desarrollo de gráficos por computadora en 2D y 3D multiplataforma (Linux, Windows, Unix, MacOS, PS3)
- ▶ Su nombre proviene de Open Graphics Library
- ▶ Es una especificación que define una API para la programación de gráficos en varios lenguajes

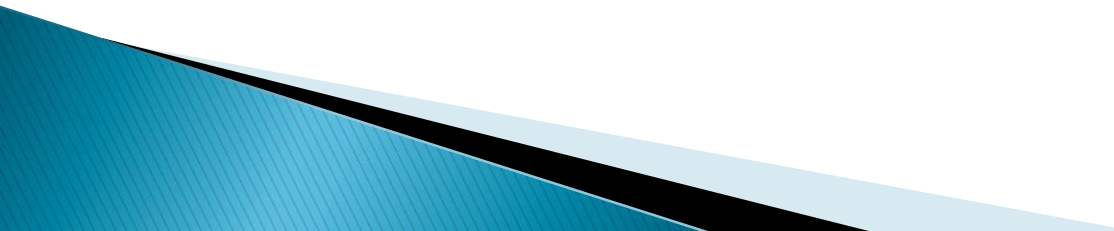




Características de OpenGL

- ▶ **Objetivos**
 1. Ocultar complejidad de Hardware
 2. Ocultar las capacidades del Hardware
 - ▶ **Propósito: Dibujado (puntos, líneas, polígonos)**
 - ▶ **OpenGL como una Máquina de estados (o switches)**
- 

OpenGL tiene...

- ▶ Funciones para dibujar en 2D y 3D
 - ▶ Transformaciones
 - ▶ Algoritmos de iluminación
 - ▶ Z-Buffering
 - ▶ Mapeo de Texturas
 - ▶ Soporte para Shaders
 - ▶ Blending, elementos de antialiasing , efectos de Fog
- 

OpenGL no tiene...

- ▶ Funciones para manejar ventanas, eventos de ratón o teclado
- ▶ Funciones o procedimientos para crear un objeto completo a partir de una descripción natural ejemplo: comando `drawCar()`;
- ▶ Soporte nativo para dibujado de curvas
- ▶ Soporte nativo para audio y/o texto
- ▶ Detección de colisiones
- ▶ Carga de imágenes

Qué pasa con lo que no tiene

- ▶ Ventanas, mouse, teclado????
 - GLUT, JOGL
 - API de Win32
 - IDE de desarrollo (forms, eventos, etc.)
- ▶ Imágenes
 - SDL Image
 - Devil
- ▶ Sonido
 - SDL Mixer
 - OpenAL
 - fmod

Sintaxis Básica

- ▶ OpenGL contiene 3 elementos distintos básicos:
 - Funciones
 - Constantes
 - Tipos de datos

Funciones

► Usos

- Dibujado de elementos
- Manipulación de proyecciones
- Transformaciones Geométricas
- Habilidad / Deshabilitación de “switches”
- Texturizado
- Aplicación de elementos para lograr iluminación
- Cuenta con 150 funciones básicas

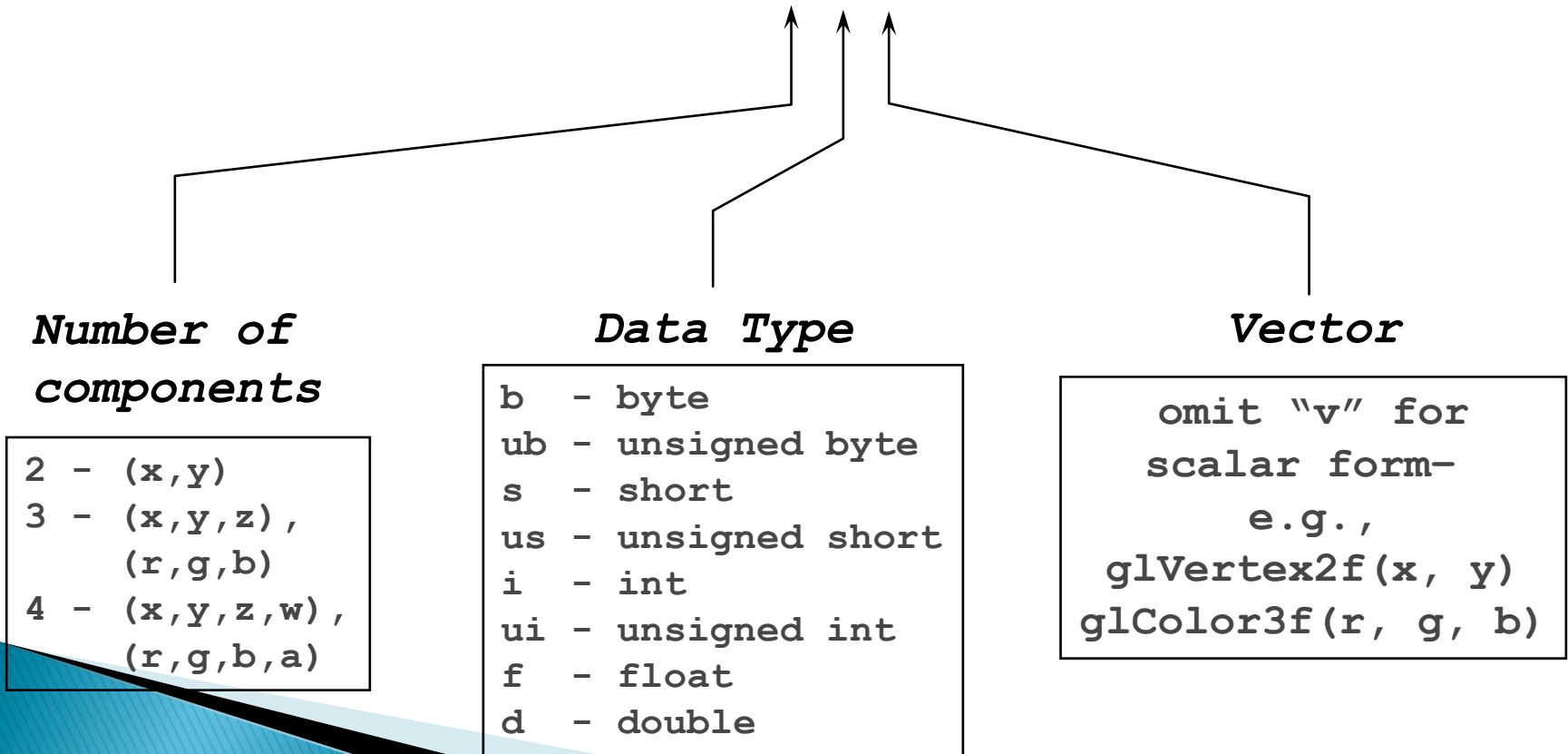
Sintaxis de funciones

- ▶ Se anteponen gl si pertenecen a la biblioteca OpenGL, glu o glut si son de éstas respectivamente
- ▶ Contienen información al final de cuántos parámetros reciben y cuál es su tipo
- ▶ Ejemplos:
 - glVertex3f
 - glVertex4fv
 - glVertex2i

OpenGL Vertex/Color Command Formats

`glVertex3fv(v)`

`glColor3fv(v)`



Constantes

▶ Usos

- Activar o desactivar ciertas propiedades de dibujo o switches
- Dar ciertos valores a parámetros de funciones que configuran algún elemento

- ▶ Mayúsculas, empezando con GL
- ▶ Nombres descriptivos

Constantes

- ▶ Ej.
 - GL_LIGHTING
 - GL_COLOR_BUFFER_BIT
 - GL_LINES
- ▶ También las encontramos en GLUT
 - Ej.
 - GLUT_LEFT_BUTTON

Máquina de estados

► Set:

- `glPointSize(size) ;`
- `glLineWidth(width) ;`
- `glLineStipple(repeat, pattern) ;`
- `glShadeModel(GL_SMOOTH) ;`

► Get:

- `glGet*() ;`

► Habilitaciones:

- `glEnable(GL_LIGHTING) ;`
- `glDisable(GL_TEXTURE_2D) ;`
- `glIsEnable(GL_DEPTH_TEST) ;`

Seteando estados

- ▶ OpenGL es una máquina de estados: los polígonos son afectados por el color, la transformación, el modo de render actuales.
- ▶ Habilitar y deshabilitar opciones como iluminación, uso de texturas y alpha blending.
 - `glEnable (GL_LIGHTING); // habilitar lighting (disabled por default)`
- ▶ Olvidarse de habilitar/deshabilitar algo es el origen más común de los bugs. Corroborar siempre el estado de las variables que necesitamos y sus valores por default. (lista de defaults en Apéndice B).

Mini “engine” (método draw)

- ▶ Borrar contenido del buffer actual
- ▶ Setear transformaciones (View Projection, Viewport transf.)
- ▶ Loop para cada primitiva:
 - Transformaciones
 - Tipo de primitiva, atributos, estados de render
- ▶ Flush o Swap buffers

Trabajo Práctico



Próximos pasos

- ▶ En la web: (próximamente)
 - Clases.
 - Links.
- ▶ Busquen (o nos piden):
 - Recursos (3ds, ase, obj)
- ▶ **En Diciembre:**
 - Trabajo de cursada (load, light, cámara).
- ▶ Consultas vía MAIL (claridad e info):
 - jpdamato@exa.unicen.edu.ar
 - cristianGB@gmail.com

- ▶ <https://jogl-demos.dev.java.net/>
- ▶ <http://pepijn.fab4.be/software/nehe-java-ports/>