

Gráficos Avanzados

Repaso de Gráficos I (2)



Temas a tratar

- Arquitectura Cliente-Servidor OpenGL.
- Interactividad gráfica.
- Primitivas 3D. Matrices.
- Visión 3D
- Iluminación



Arquitectura Cliente-Servidor OpenGL.

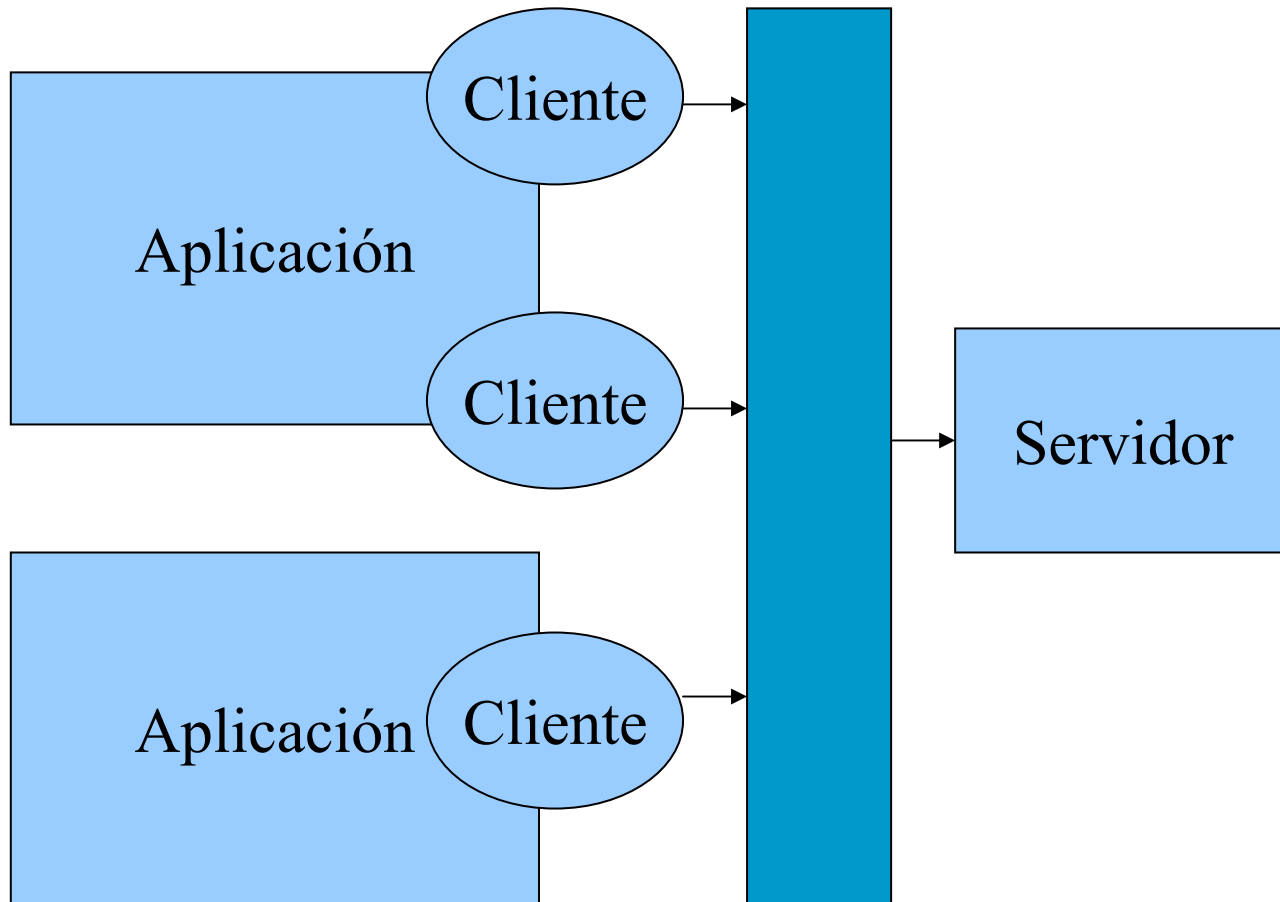
■ Cliente/Servidor:

- El cliente genera envía comandos al servidor.
- El servidor ejecuta/interpreta los comandos.
- El servidor no tiene que residir necesariamente en el máquina cliente.
- El cliente se identifica por el contexto.

■ Efectos de ser Cliente/Servidor:

- OpenGL no tiene eventos, debemos comunicar acciones manualmente.
- OpenGL sólo conoce la estructura del framebuffer de la ventana.

Arquitectura Cliente-Servidor OpenGL.





Interactividad

- Un interfaz gráfico ha de ser muy interactivo.
- Métodos
 - Menús (Contextuales y desplegable)
 - Botones
 - Picking
 - Manipuladores



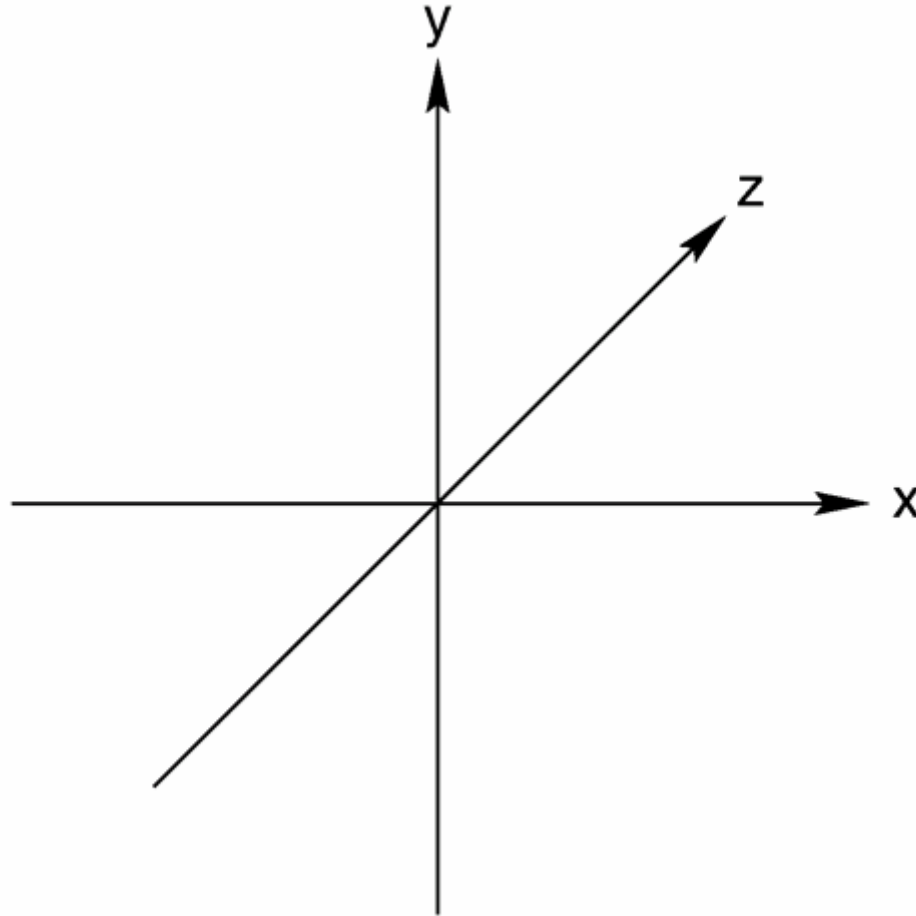
Interactividad

■ Ideal:

- El usuario debe ver pocos botones.
- Los botones han de ser claros y preferiblemente cuadrados.
- Interacciones
 - Son movimientos de ratón, pulsación de botones, etc.
 - Debe haber un máximo de 3 interacciones para los operaciones menos comunes y 1 para las más comunes.
- El espacio de trabajo ha de ser grande, se ha de ver el modelo.

Primitivas 3D. Matrices.

■ Ejes





Primitivas 3D. Matrices.

- **Coordenadas estándar 3D**
 - Son de la forma $\langle x, y, z \rangle$
 - Representan un lugar en el espacio 3D
 - Ejemplo $\langle 0, 0, 0 \rangle$ (origen de coordenadas)

Primitivas 3D. Matrices.

■ Coordenadas homogéneas

- Son de la forma $\langle x, y, z, w \rangle$
- Representan un lugar en el espacio o un vector.
 - Ejemplos:
 - $\langle 0, 0, 0, 1 \rangle$ (origen de coordenadas)
 - $\langle 0, 0, 1, 0 \rangle$ (Eje Z)

■ Coordenadas homogéneas -> estándar

Coordenadas homogéneas $\langle x, y, z, w \rangle$

Coordenadas estándar $\langle \frac{x}{w}, \frac{y}{w}, \frac{z}{w} \rangle$

Primitivas 3D. Matrices.

- Matrices de transformación en coordenadas estándar:
 - Rotación
 - Escalado
- Matrices de transformación en coordenadas homogéneas:
 - Rotación
 - Escalado
 - Traslación
 - Perspectiva
 - ... (Qualquier transformación lineal)

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

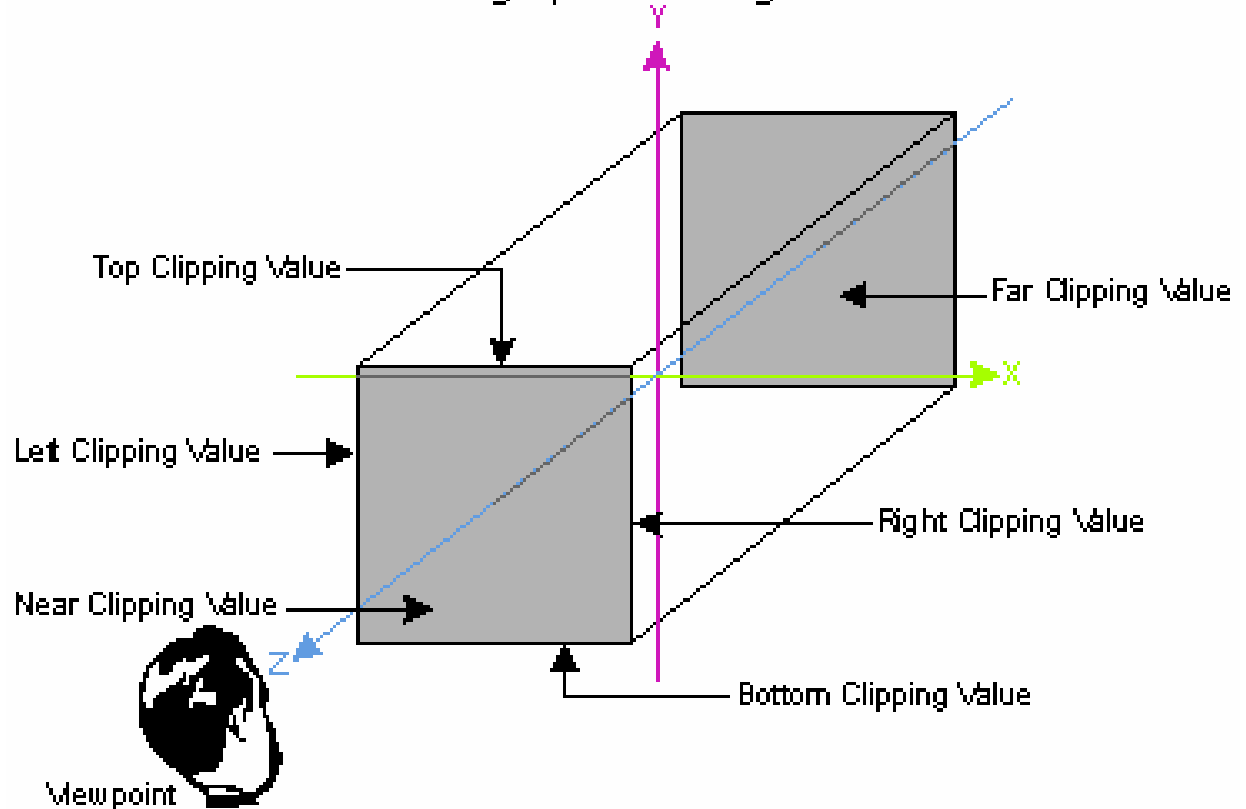
Primitivas 3D. Matrices.

- Las matrices en coordenadas homogéneas se pueden concatenar multiplicando.
 - Ejemplo:
 - Rotar 30° = M1
 - Escalar $2 \times$ = M2
 - Trasladar -5 en z = M3
 - $Y = X * M1 * M2 * M3$;

Visión 3D. Proyecciones.

■ Ortográfica (glOrtho)

An Orthographic Viewing Volume



Visión 3D. Proyecciones.

■ Ortográfica: glOrtho:

```
void glOrtho(  
GLdouble left,  
GLdouble right,  
GLdouble bottom,  
GLdouble top,  
GLdouble near,  
GLdouble far);
```

$$t_x = -\frac{right+left}{right-left}$$

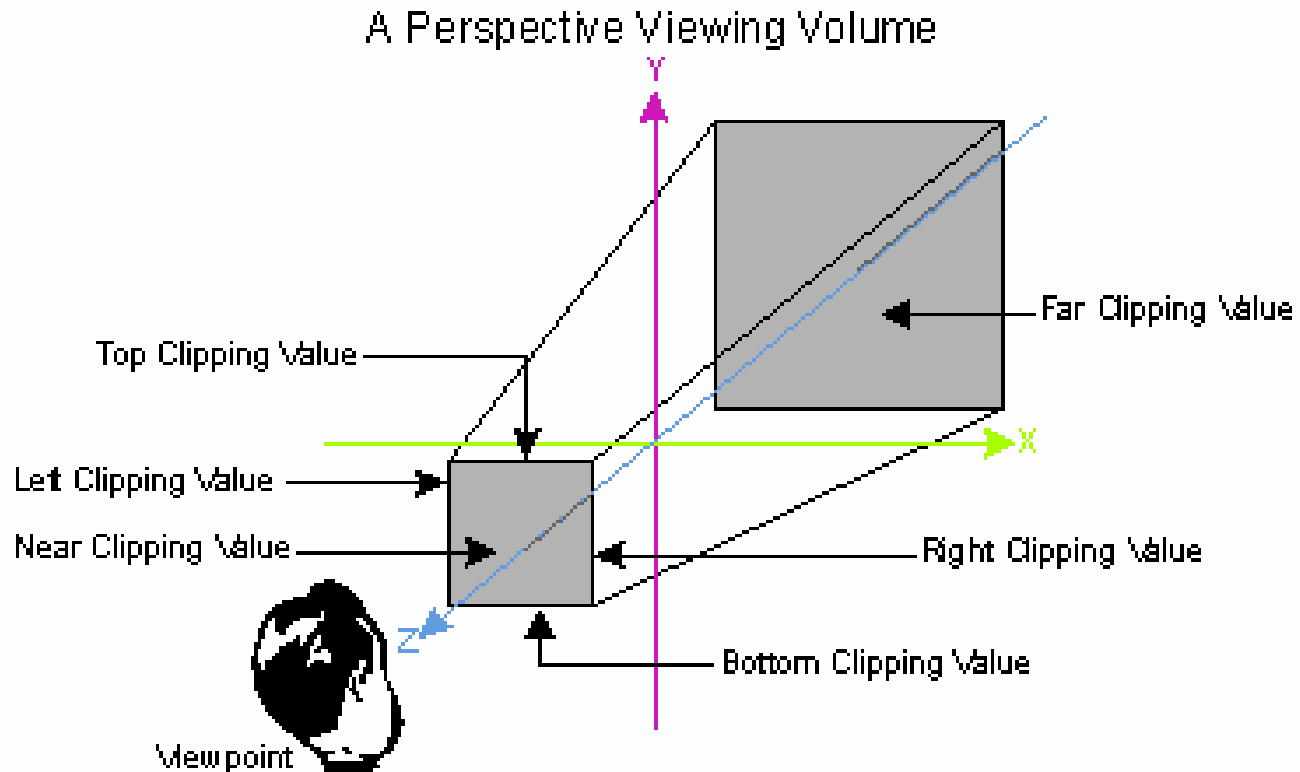
$$t_y = -\frac{top+bottom}{top-bottom}$$

$$t_z = -\frac{far+near}{far-near}$$

$$\begin{pmatrix} \frac{2}{right-left} & 0 & 0 & t_x \\ 0 & \frac{2}{top-bottom} & 0 & t_y \\ 0 & 0 & \frac{-2}{far-near} & t_z \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Visión 3D. Proyecciones.

■ Perspectiva (glFrustum, gluPerspective)



Visión 3D. Proyecciones.

■ Perspectiva: glFrustum.

- Define la pirámide truncada de perspectiva.

```
void glFrustum(  
GLdouble left,  
GLdouble right,  
GLdouble bottom,  
GLdouble top,  
GLdouble znear,  
GLdouble zfar);
```

$$\begin{pmatrix} \frac{2 \text{ near}}{\text{right-left}} & 0 & A & 0 \\ 0 & \frac{2 \text{ near}}{\text{top-bottom}} & B & 0 \\ 0 & 0 & C & D \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

$$C = -\frac{\text{far} + \text{near}}{\text{far} - \text{near}}$$

$$D = -\frac{2 \text{ far near}}{\text{far} - \text{near}}$$

$$A = \frac{\text{right} + \text{left}}{\text{right} - \text{left}}$$

$$B = \frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}}$$

Visión 3D. Proyecciones.

■ El atajo

```
void gluLookAt(
```

```
    GLdouble eyex, GLdouble eyey, GLdouble eyez, // donde está  
    mi ojo
```

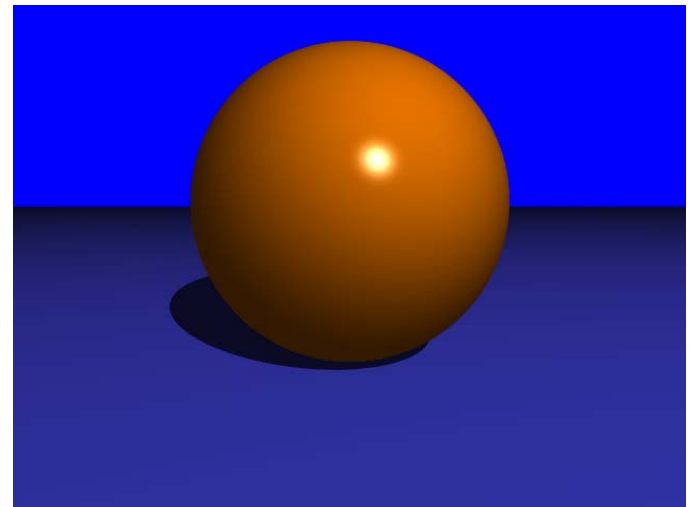
```
    GLdouble centerx, GLdouble centery, GLdouble centerz, //  
    donde está lo que quiero ver
```

```
    GLdouble upx, GLdouble upy, GLdouble upz // hacia donde  
    está el techo
```

```
);
```

Iluminación

- Consiste en sombrear un objeto alterando su color en función de un punto de luz o un foco.
- Tipos de luces OpenGL.
 - Omni
 - Spotlight



Ver

■ www.opengl.org

- Obtener los manuales de
 - GLUI
- Obtener los fuentes de
 - GLUI, en especial (algebra3.h)

■ www.msdn.com

- Clase base OpenGL
 - “OpenGL Without the Pain: Creating a Reusable 3D View Class for MFC”