

Daniel Calderón S.

**CG 3D,
OPENGL Y
MODELACIÓN**

Modelo - Visualización

Computación Gráfica

Síntesis

Datos

Modelos
Matemáticos

Visualización
Imagen 2D

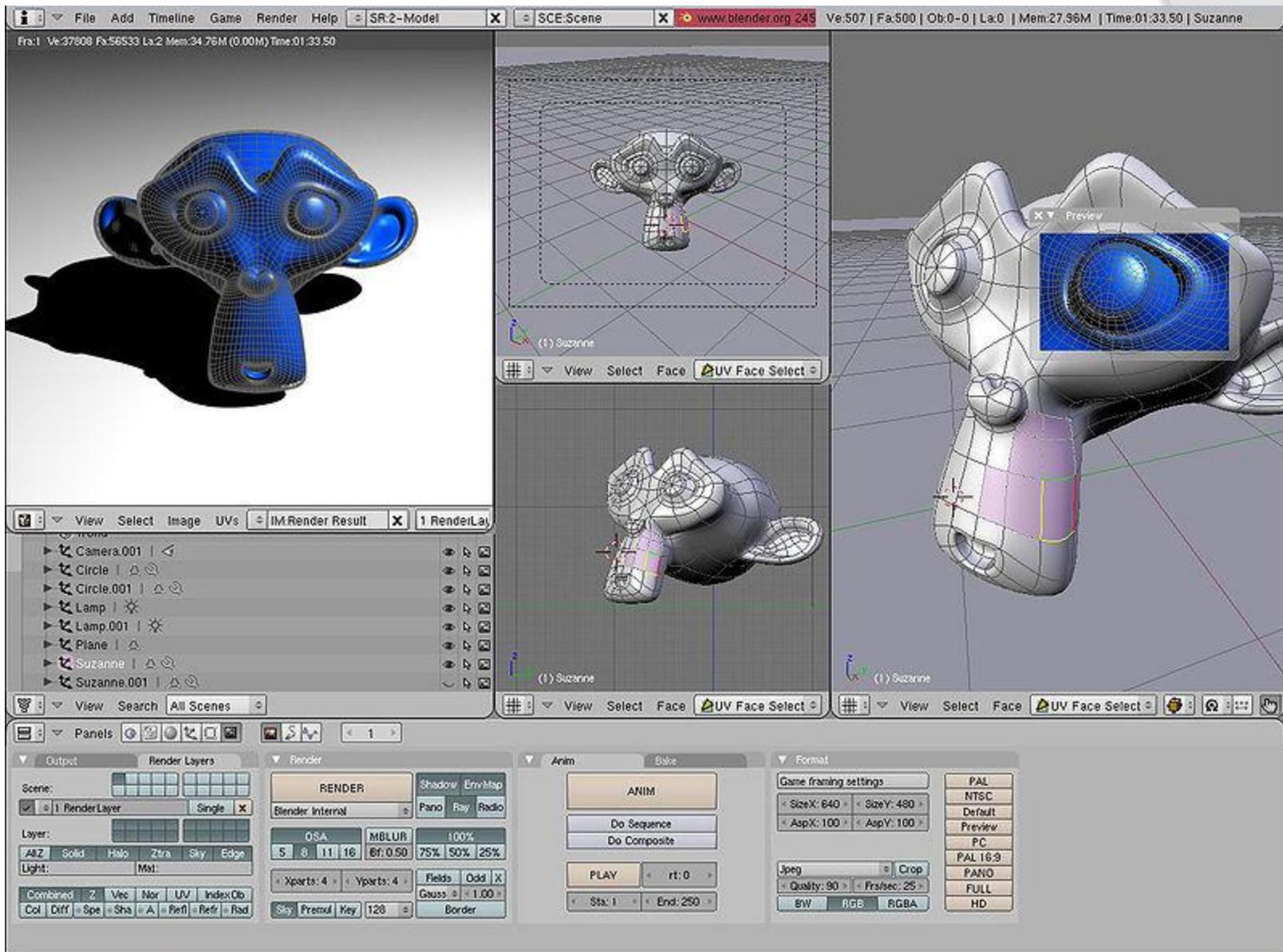
Análisis

Procesamiento de
Imágenes









CG en el cine...

Tron (1982)

- Intervienen imágenes generadas por computadora.

The Abyss (1989),
Terminator 2 (1991)

- Las imágenes generadas por computadora adquieren un rol importante.

Jurassic Park
(1993)

- Mezcla perfecta

Toy Story (1995)

- Primer largometraje

Final Fantasy: The
Spirits Within (2001)

- Primera película realista

⦿ Computación gráfica

- Tarea pesada
- Procesar vértices
- Procesar píxeles

⦿ Solución:

- Hardware específico
- La GPU
 - Muchos procesadores en paralelo!



Evolución de las GPU



Pong, 1972



Pac-Man, 1983



Wolfenstein 3D, 1992



Doom, 1993

Duke Nukem 3D, 1996



Quake 3 Arena, 1999



Half-Life 2, 2004



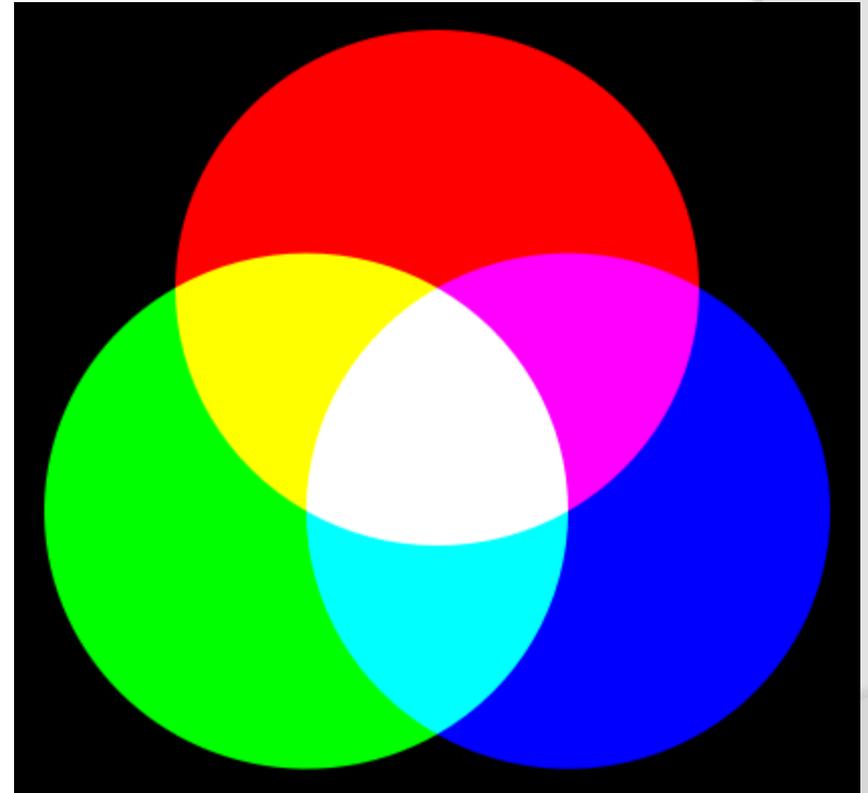
Quake 4, 2005



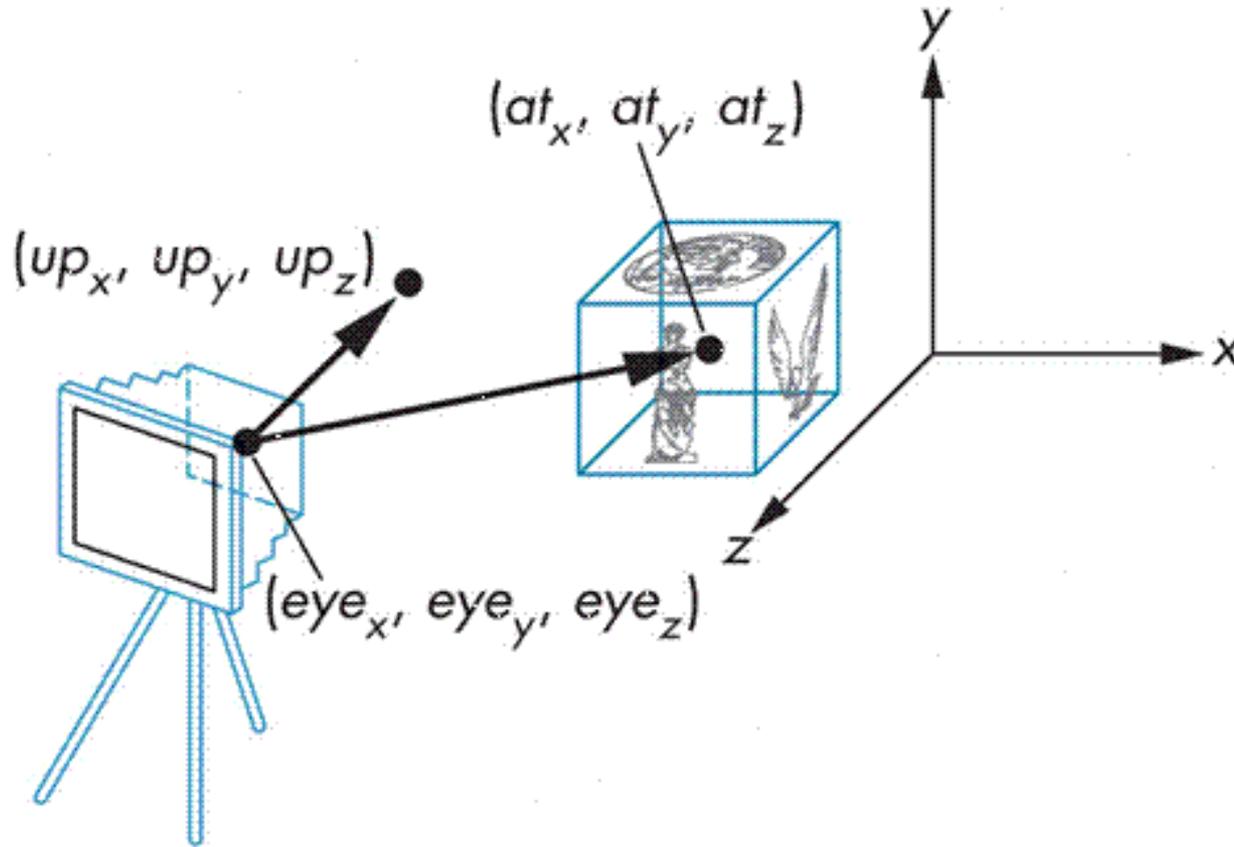
Modelo de color

- ⦿ 4 componentes:
 - Rojo
 - Verde
 - Azul
 - Transparencia (alpha)

- ⦿ $[0, 1]$: rango de mínima a máxima intensidad.

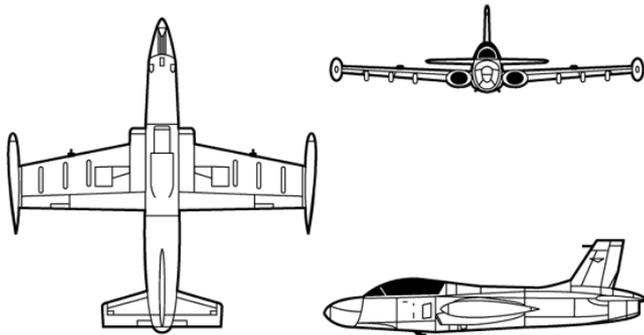
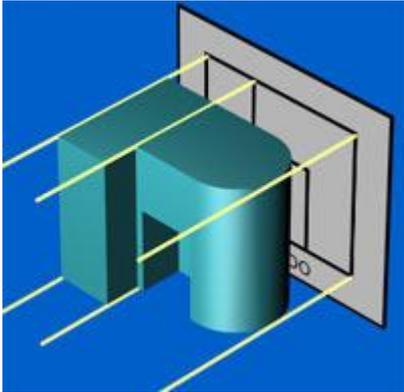


El espacio 3D y la cámara



`gluLookAt(eyeX, eyeY, eyeZ, atX, atY, atZ, upX, upY, upZ)`

Proyecciones

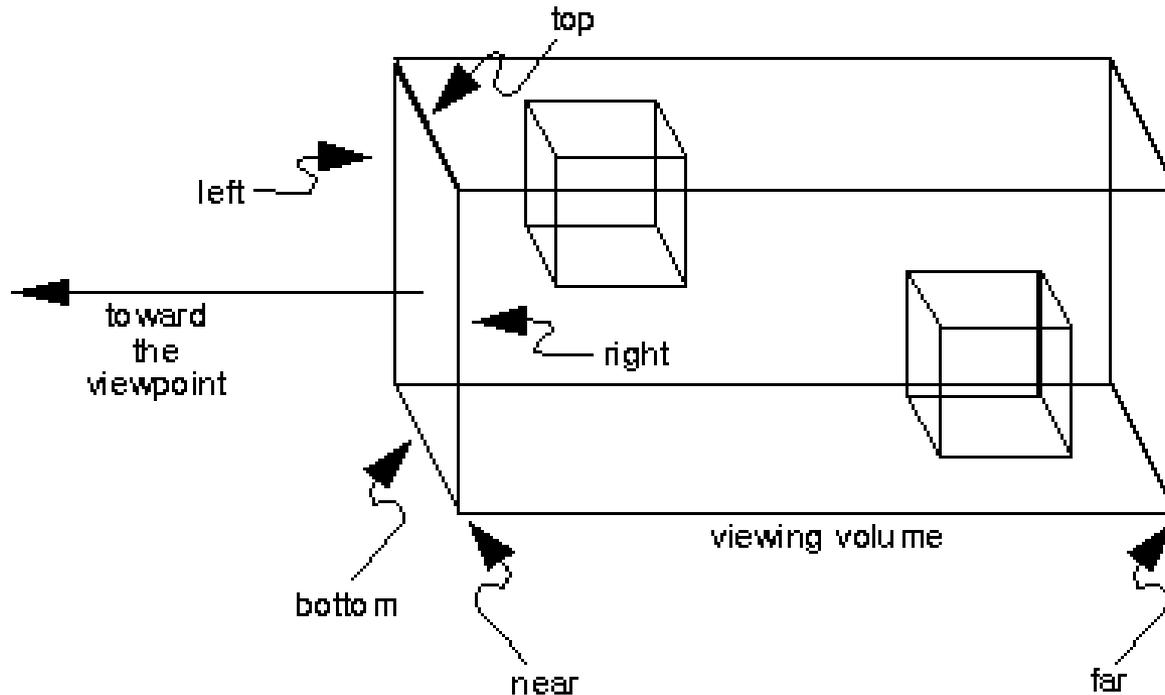


Proyección ortogonal

Proyección perspectiva

Proyecciones

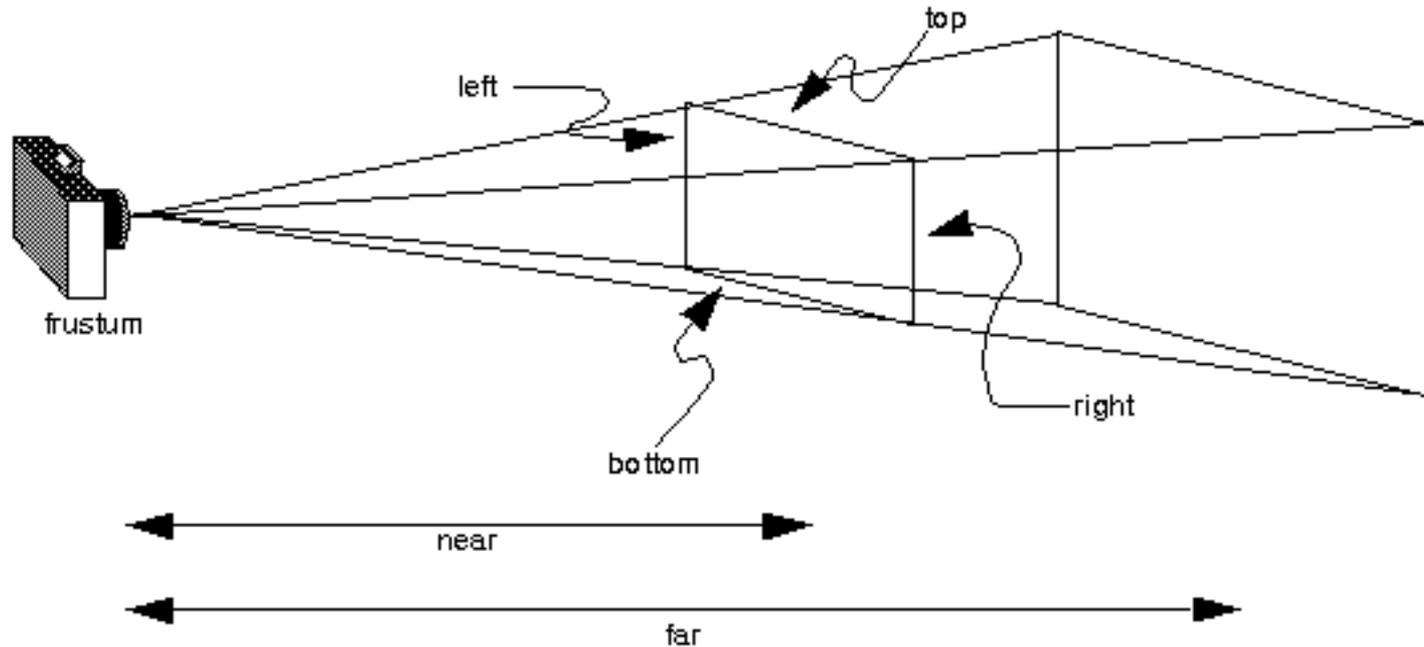
Proyección ortogonal



```
void glOrtho(izquierda, derecha, abajo, arriba, cerca, lejos);
```

Proyecciones

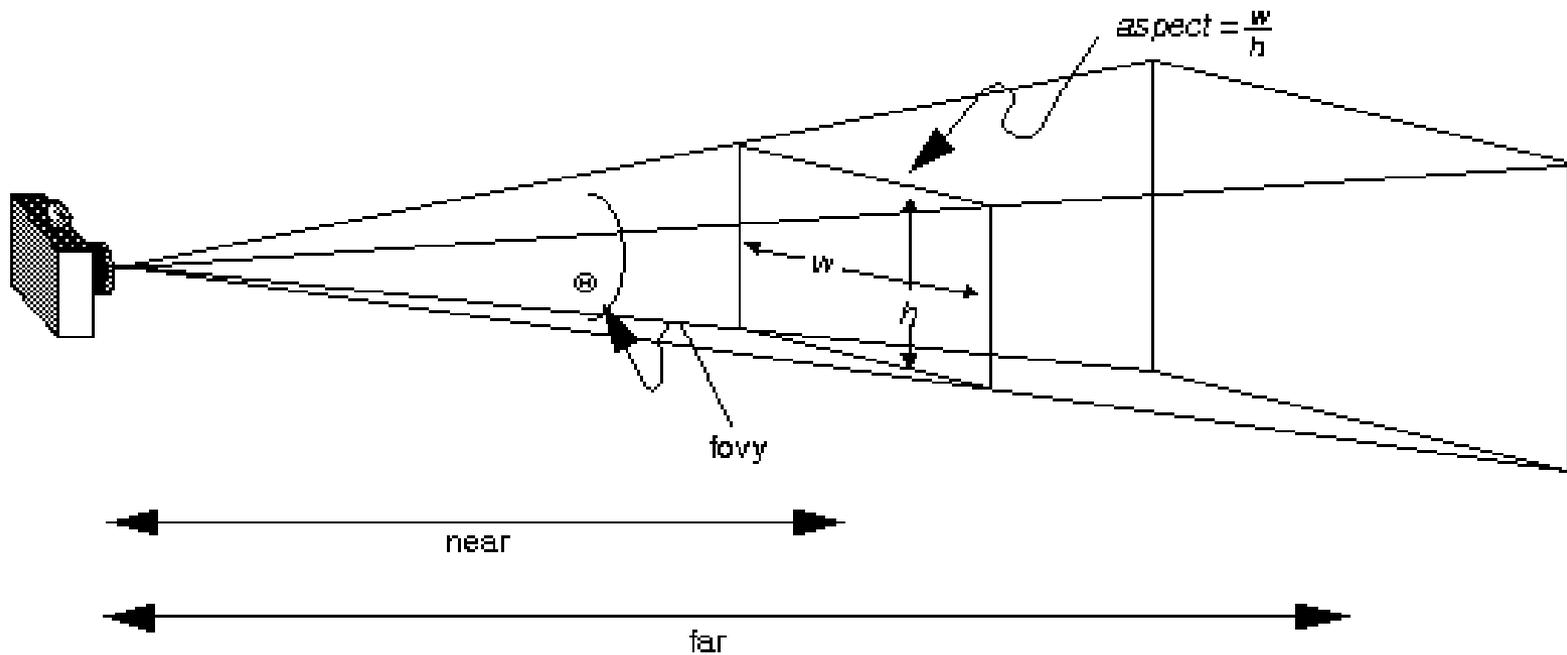
Proyección perspectiva 1



```
void glFrustum(izquierda, derecha, abajo, arriba, cerca, lejos);
```

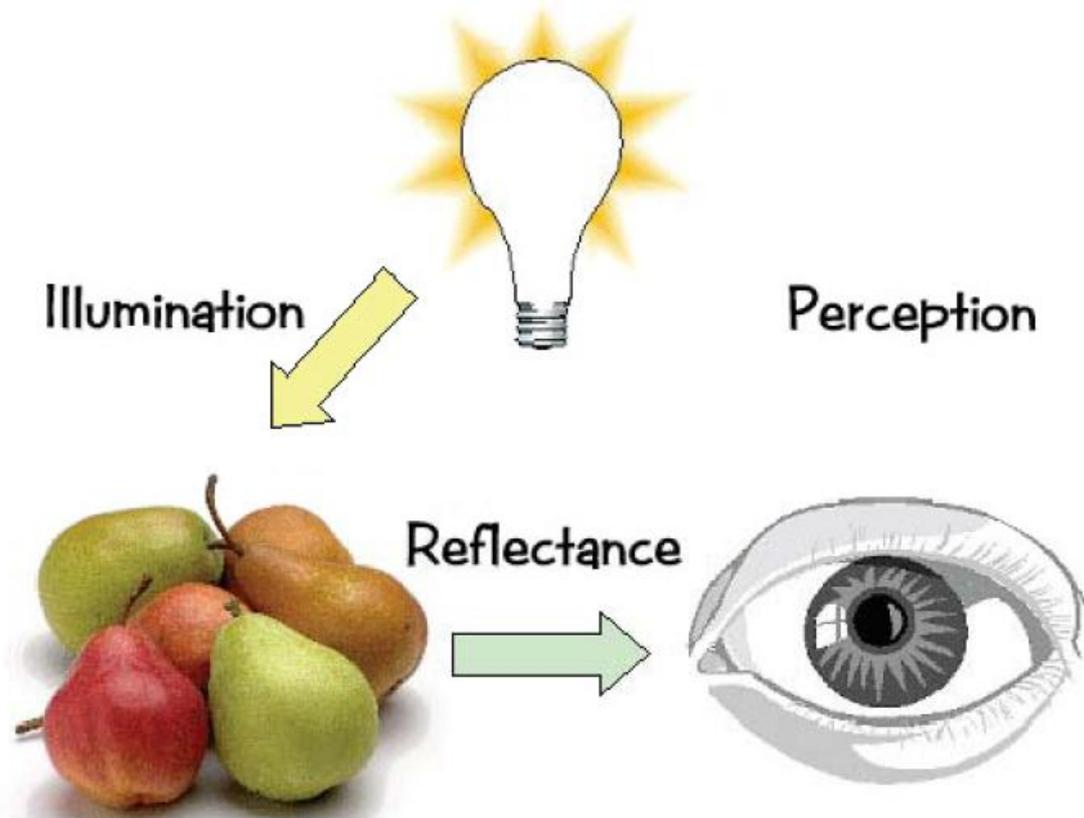
Proyecciones

Proyección perspectiva 2

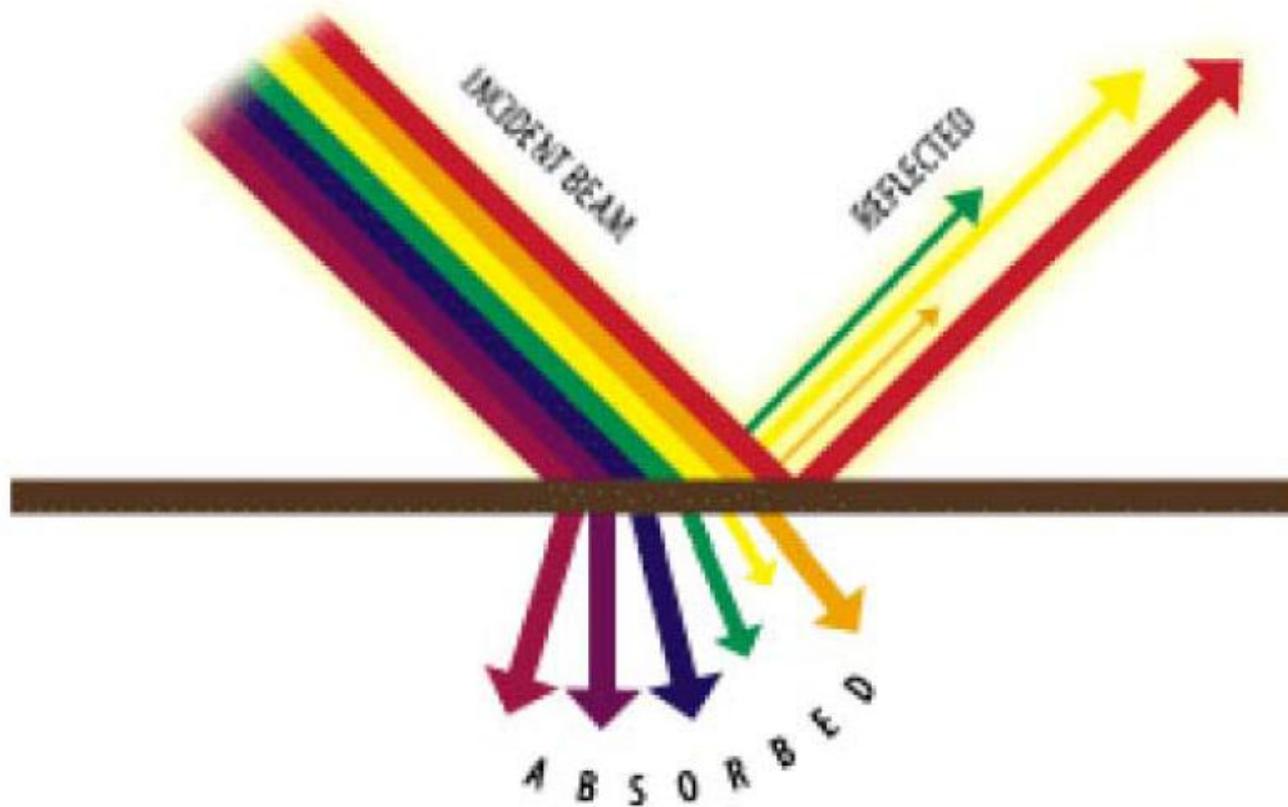


```
void gluPerspective(fovy, aspecto, cerca, lejos);
```

Modelo de iluminación



Modelo de iluminación

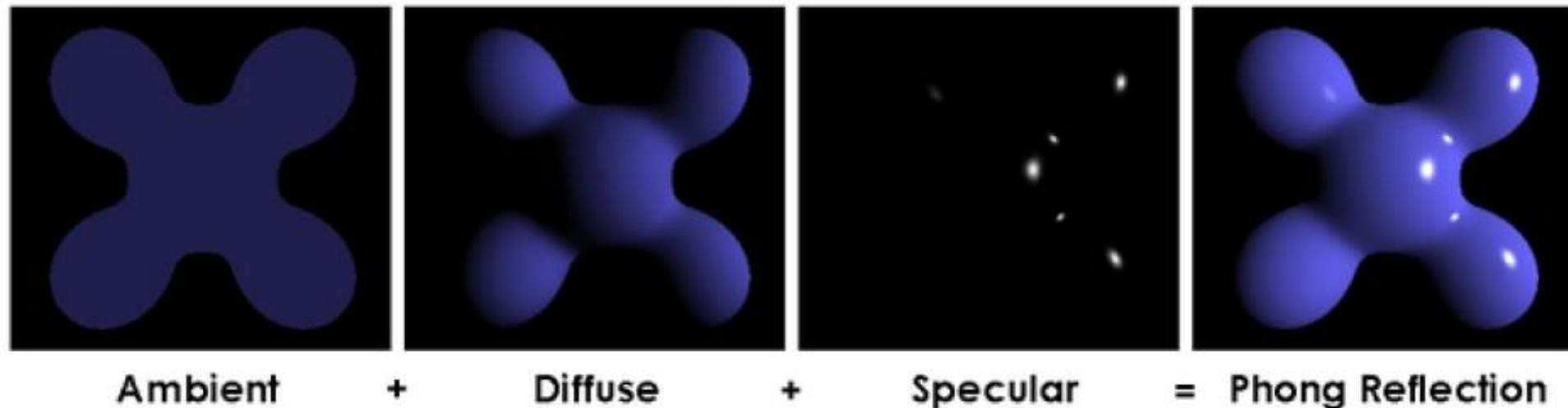


Modelo de iluminación

- ⦿ Fuente de luz
- ⦿ Características del material

Modelo de iluminación

Fuente de luz



```
glEnable(GL_LIGHTING)
glEnable(GL_LIGHT1)
glLightfv(GL_LIGHT1, GL_POSITION, [ 700, 700.0, 700.0, 1.0])
glLightfv(GL_LIGHT1, GL_AMBIENT, [ 0.2, 0.2, 0.2])
glLightfv(GL_LIGHT1, GL_SPECULAR, [ 10, 10, 10])
glLightfv(GL_LIGHT1, GL_DIFFUSE, [ 200, 200.0, 200.0, 1.0])
```

Modelo de iluminación

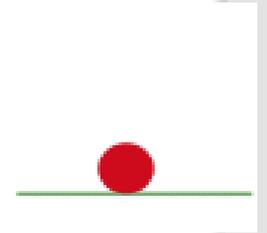
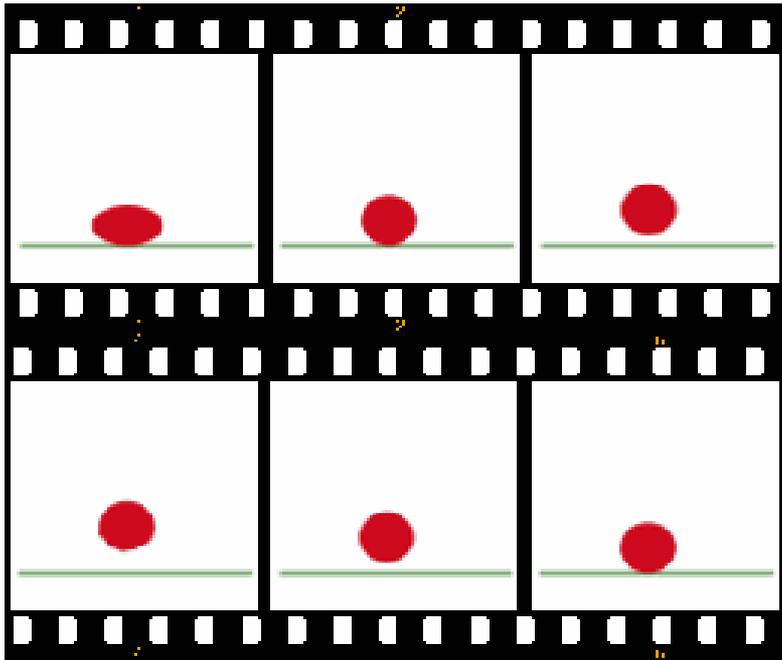
Configuración del Material



```
glMaterialfv(GL_FRONT, GL_AMBIENT, [ 0.0, 0.0, 0.0, 1.0 ] )  
glMaterialfv(GL_FRONT, GL_DIFFUSE, [0.8, 0.8, 0.8, 1.0])  
glMaterialfv(GL_FRONT, GL_SPECULAR, [ 0.0, 0.0, 0.0, 1.0 ] )  
glMaterialiv(GL_FRONT, GL_SHININESS, [0])  
glMaterialfv(GL_FRONT, GL_EMISSION, [ 0.0, 0.0, 0.0, 1.0 ] )
```

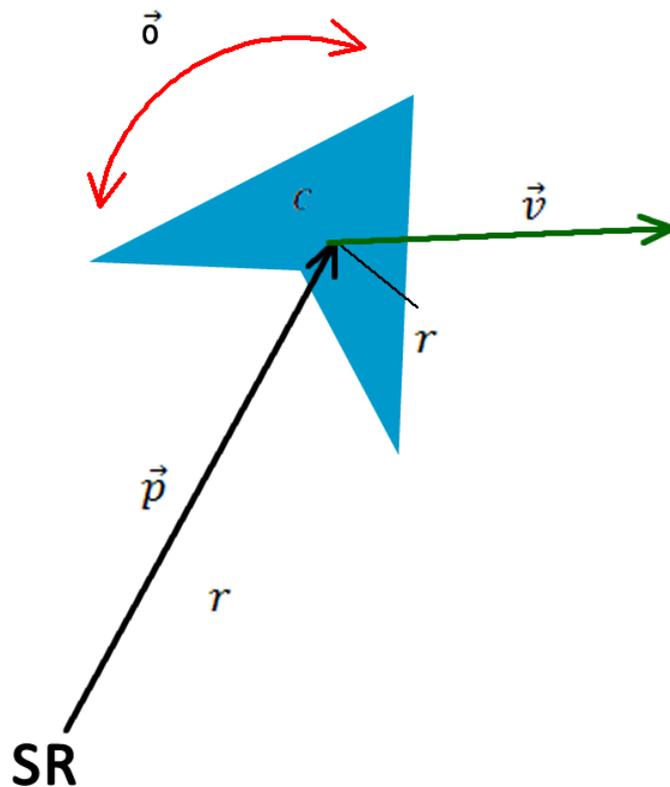
```
glEnable(GL_COLOR_MATERIAL)  
glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE)
```

Animación



Animación

Definiendo actores...



- Se considera una clase por tipo de objeto con las siguientes características:
 - Variables de instancia: posición, orientación y velocidad instantánea.
 - Métodos: mover(), dibujar().

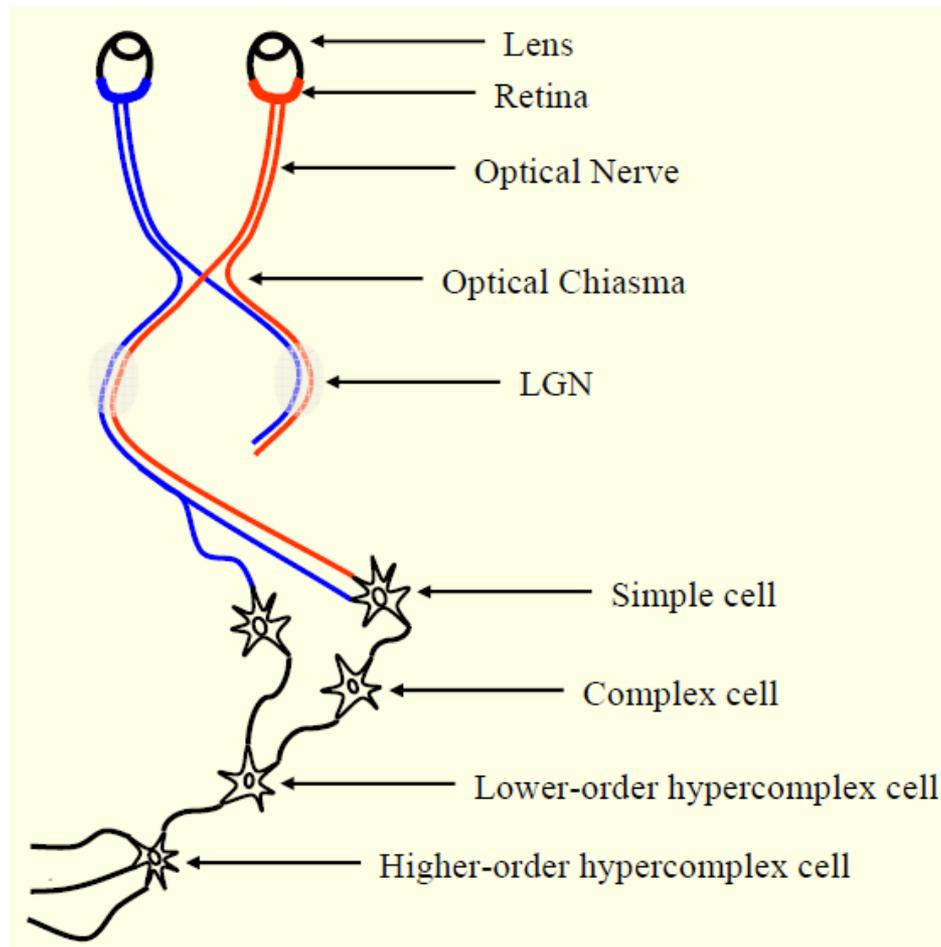
Animación

Construcción de cada frame...



- La animación se implementa con un loop, en cada una de las iteraciones se debe:
 - Detectar eventos del teclado.
 - Cada objeto debe actuar, esto es:
 - Mover cada objeto.
 - Detectar interacciones entre ellos.
 - Dibujar cada objeto.

Sistema de visión



Sistema de visión

- ⦿ El ojo tarda en percibir la imagen.
 - Se visualiza movimiento si la imagen cambia.
 - La televisión trabaja con 30 fps.
 - No procesa mas de 70 fps.
- ⦿ 2 ojos producen visión estereoscópica (o binocular)
 - Sensación de profundidad.
 - Permite estimación de distancias, elemental para muchos seres vivos.

Visualización estereoscópica



- ⦿ Se necesita una imagen para cada ojo.
- ⦿ Se necesita reproducir al menos 140 fps.

Cine 3D

- Procesador
- Gafas
 - Se sincronizan con el sistema de proyección tapando uno u otro ojo.
- Camara especial.

