



¿Qué es la Informática Gráfica?

- La informática gráfica se preocupa de todos los aspectos relacionados con la generación de imágenes a través de un computador. En particular **estudia las herramientas utilizadas para producir dichas imágenes.**
- Las herramientas pueden ser hardware (tarjetas gráficas, monitores, impresoras, etc) y software (librerías gráficas).
- Nosotros nos centraremos en el software, estudiando técnicas y algoritmos para el modelado de gráficos. La interacción con el HW la haremos a través de la librería gráfica **Open GL** (Open Graphics Library).

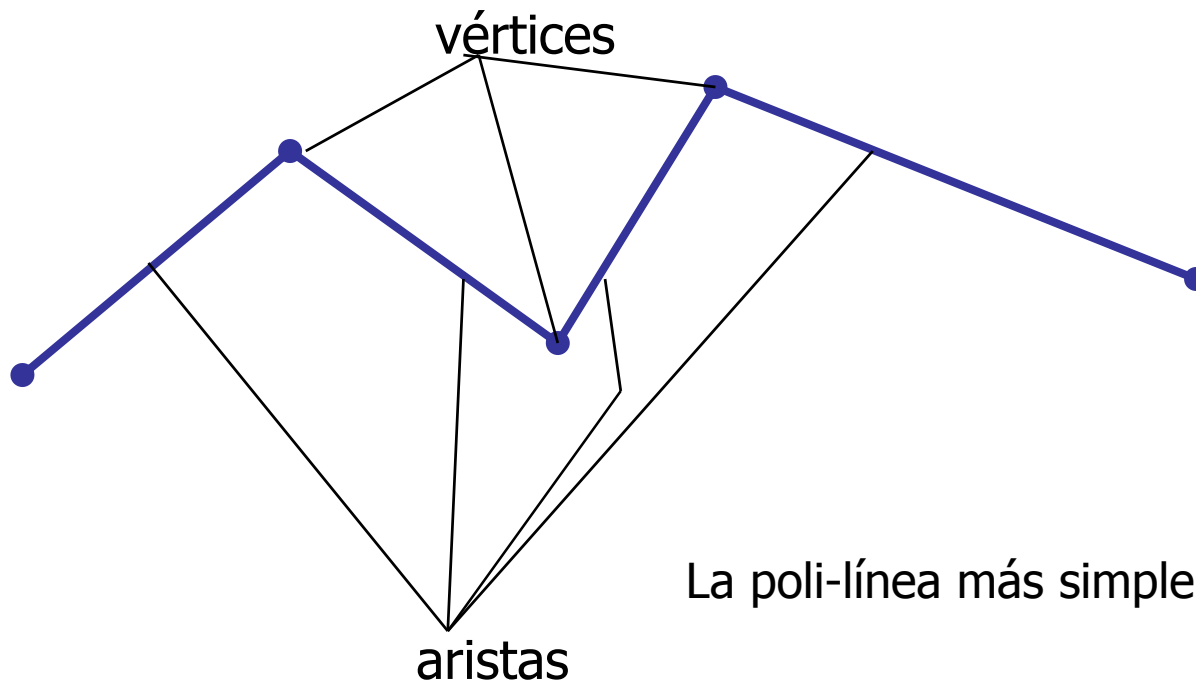


Elementos básicos de una gráfico generado por ordenador

- Un gráfico se compone de una serie de objetos básicos, denominados **primitivas de salida**.
- Una clasificación de éstas es:
 - ✓ Poli-líneas.
 - ✓ Texto.
 - ✓ Regiones rellenas.
 - ✓ Imágenes rasterizadas.

Poli-líneas

- Una poli-línea **es una secuencia contigua de líneas rectas** (aristas) unidas mediante **vértices**.

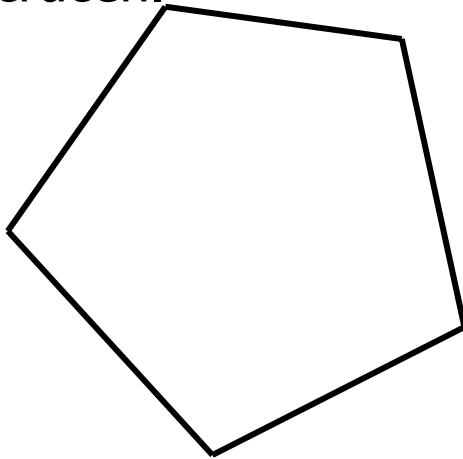


La poli-línea más simple es una línea recta

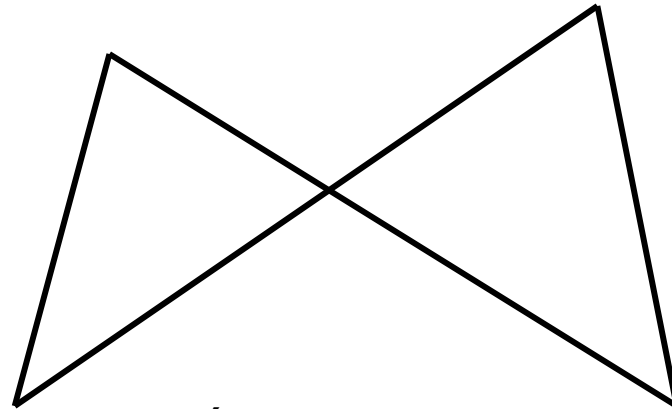


Poli-líneas

- Un gráfico construido a base de poli-líneas se denomina **dibujo de líneas**.
- Un **polígono** es una poli-línea en la que el primer y último vértice están conectados mediante una arista. Es **simple** si no existen dos aristas que se crucen.



Polígono simple



Polígono no simple

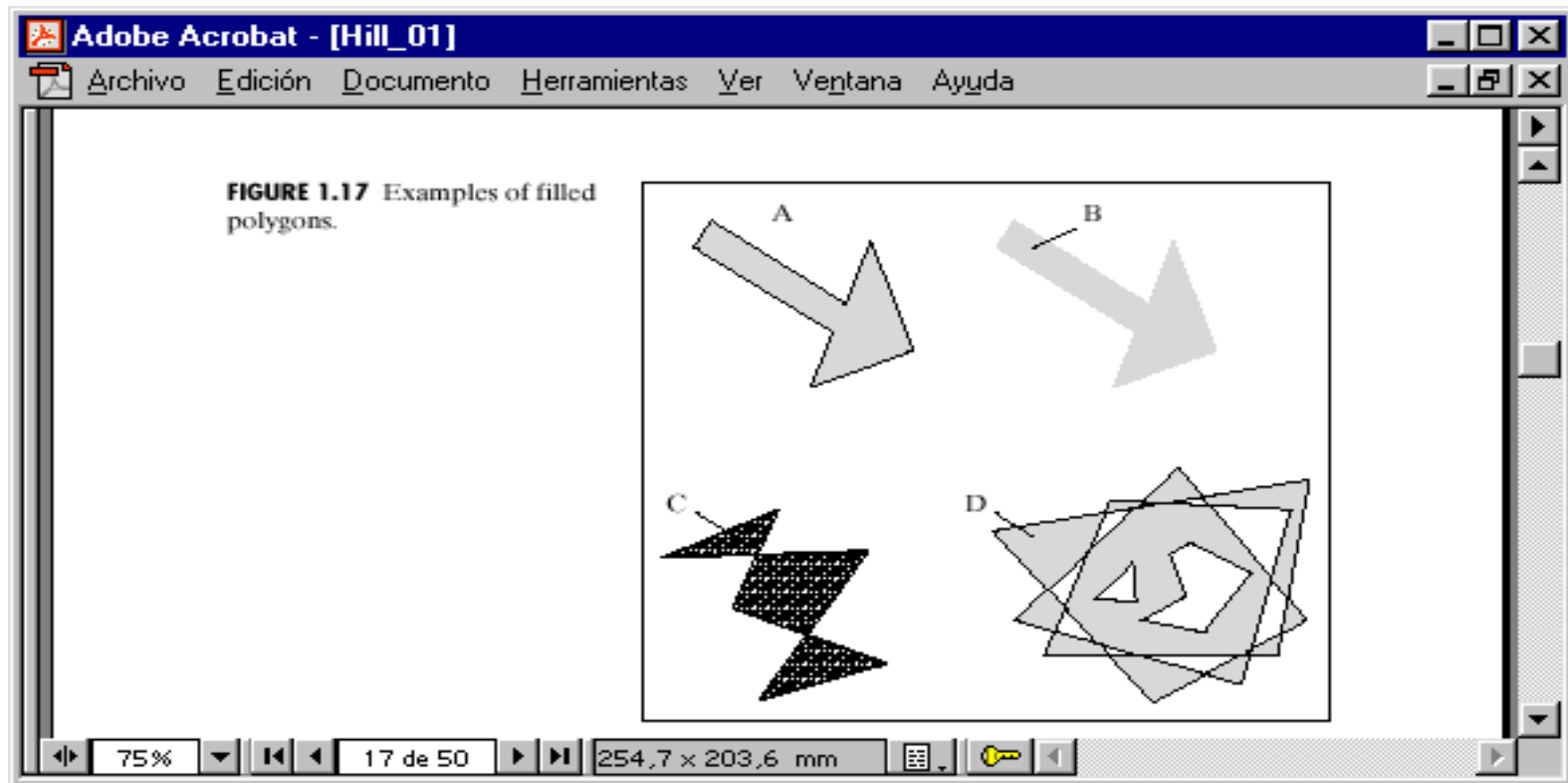


Poli-líneas

- Los polígonos son fundamentales en informática gráfica debido a que son fáciles de definir y a que muchos de los algoritmos de “rendering” (proceso por el cual un ordenador muestra una imagen a partir de un modelo) se han definido para que funcionen de forma óptima en presencia de polígonos.
- Los **atributos** asociados a una poli-línea son: color, grosor, estilo de línea

Regiones rellenas

- Son partes del gráfico que están **rellenas con algún color o patrón**. En general están delimitadas por polígonos.

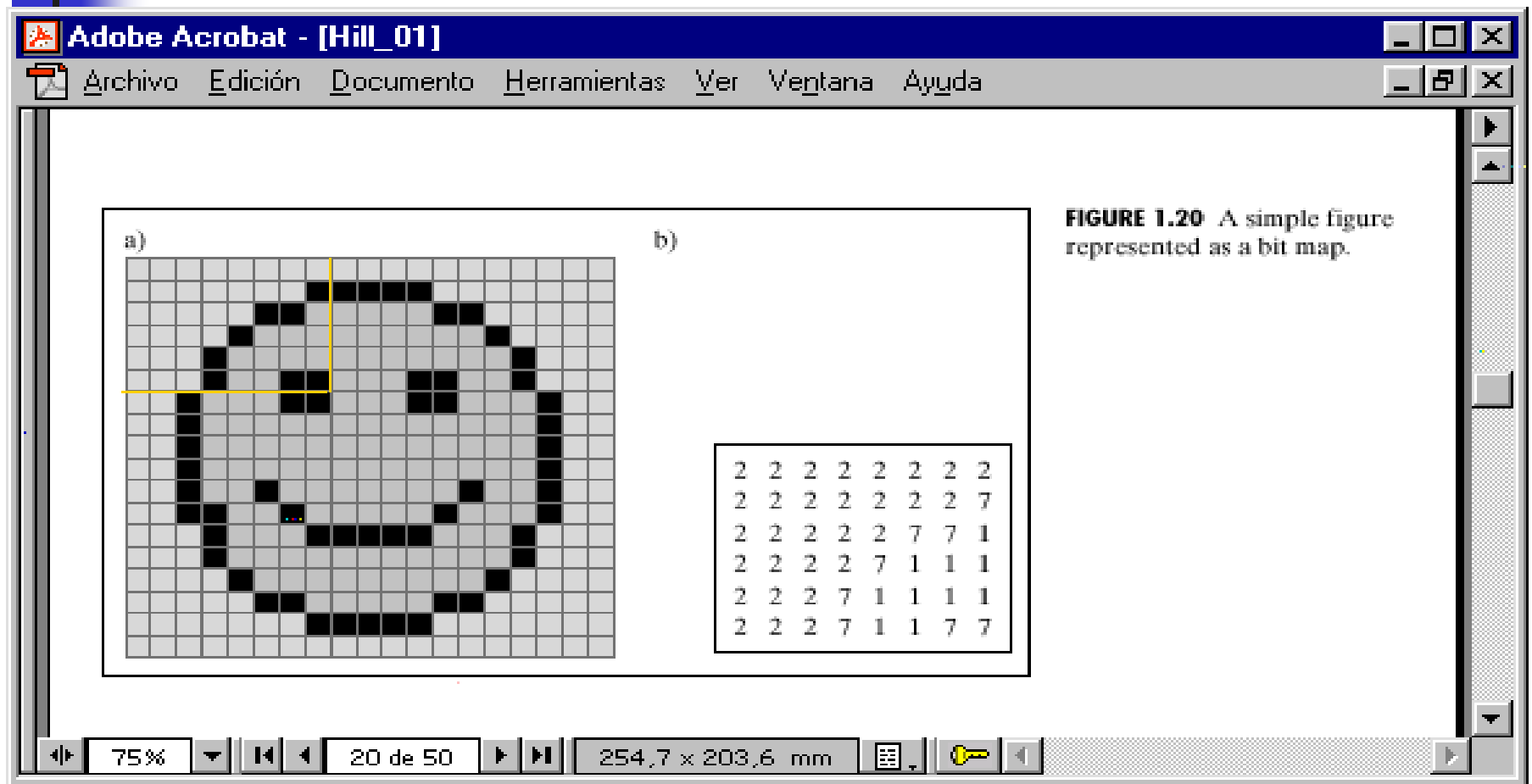




Imágenes rasterizadas

- **Son imágenes que aparecen en el gráfico** y que han sido cargadas como una entidad independiente. Por ejemplo, podemos incorporar un gráfico almacenado en un fichero **"bmp"** dentro de nuestra escena.
- Una imagen rasterizada se almacena en el ordenador como una matriz de valores numéricos (**mapa de píxeles**). Cada valor numérico representa el valor de un píxel y más concretamente almacena información sobre el color asociado a un píxel.
- Suele utilizarse también el término **bitmap** para hacer referencia a un mapa de píxeles, aunque en algunos ambientes bitmap se reserva para aquellos mapas de píxeles en los que un píxel se representa por un simple bit.

Imágenes rasterizadas





Imágenes rasterizadas: Colores

- Cada píxel tiene un **"valor de color"**, que es un valor numérico que representa un color.
- La forma más común de asociar un número a un color es describir un color como una combinación de rojo, verde y azul. Esto se conoce como **modelo de color RGB**.
- Cada valor de píxel es un triple ordenado **(r,g,b)**, donde r,g,b representan las intensidades de rojo, verde y azul respectivamente. r, g y b son cadenas de bits.
- La **profundidad de color** se define como la suma de los bits asociados a las componentes r, g y b. Por ejemplo, si utilizamos 2 bits para el rojo, 2 para el verde y 2 para el azul, tendremos una profundidad de color de 6 bits por píxel, y una paleta de colores de $2^6 = 64$ posibles colores.



Imágenes rasterizadas: Colores

- La profundidad de color ideal es la de 24 bits por píxel, conocida como *color verdadero*. Por encima de esta profundidad el ojo humano es incapaz de notar la diferencia.
- A mayor profundidad de color y mayor resolución, más memoria se necesita para almacenar un gráfico. El color verdadero con una cuadrícula de 1280x1024, necesita

$$1280 \times 1024 \times 24 = 31457280 \text{ bits}$$

(casi 4 Mb)



Imágenes rasterizadas: Colores

- Algunas correspondencias de colores en el modelo RGB son:

✓ Negro:	(0,0,0)
✓ Azul:	(0,0,1)
✓ Verde:	(0,1,0)
✓ Cian:	(0,1,1)
✓ Rojo:	(1,0,0)
✓ Magenta:	(1,0,1)
✓ Amarillo:	(1,1,0)
✓ Blanco:	(1,1,1)

glColor3f(x,y,z);



Color indexado

- En lugar del modelo RGB, algunos sistemas utilizan una **tabla de colores** que ofrece una asociación “programable” entre un valor de píxel y el color final que se muestra.
- Supongamos que cada píxel se guardan 6 bits.
- Estos bits se usan como un índice de una tabla de $2^6=64$ entradas.
- Cada celda de la tabla contiene información sobre un color. Su tamaño es independiente del tamaño reservado para cada píxel.
- Por ejemplo, si cada celda tiene una capacidad de almacenamiento de 15 bits, entonces podemos tener 2^{15} colores diferentes (**paleta del sistema**).



Color indexado

- La paleta es de 2^{15} colores, pero la tabla sólo tiene 64 entradas. Por lo tanto cuando se dibuja una imagen sólo podrán aparecer en un “repintado” 64 colores diferentes.
- Durante el proceso de dibujo de la imagen, la tabla de colores no se modifica. Se podrá modificar una parte de ésta durante el periodo comprendido entre un repintado y otro.
- En general, para un tamaño de **b** bits por píxel y **w** bits de capacidad para cada celda de la tabla, la paleta constará de 2^w colores, pero en cada repintado sólo podrán mostrarse 2^b colores distintos.
- **w y b no tienen relación alguna**, aunque w se suele elegir como múltiplo de 3, de forma que cada porción $w/3$ se corresponda con R, G y B respectivamente.



Color indexado

- Por ejemplo, si $w=15$:
- rojo

01010

verde

11001

azul

00001
- b debe ser menor que w , esto es, la tabla nunca debe tener más entradas que colores tenga la paleta.
- El color indexado con respecto al RGB tradicional tiene las ventajas:
- ✓ Ofrece una gran flexibilidad de colores.
 - ✓ Requiere menos memoria.
- Como contrapartida, el color indexado requiere que el programador defina la paleta de colores y además sólo pueda mostrar 2^b colores en cada "repintado".



Color indexado: Ejemplo

- ❑ Para una profundidad de color de 24 bits por píxel y una cuadrícula de 1280 x 1024, el modelo RGB necesita 1280 x 1024 x 24 bits de memoria, y podemos mostrar en un repintado 2^{24} posibles colores.
- ❑ Color indexado: para un tamaño de 8 bits por píxel y una capacidad en la tabla de 24 bits por celda, tendríamos una paleta de 2^{24} posibles colores (igual que en RGB), aunque sólo podríamos mostrar en un repintado $2^8 = 256$ colores (menos que en RGB). Una imagen rasterizada necesitaría 1280 x 1024 x 8 bits de memoria (mejor que en RGB).

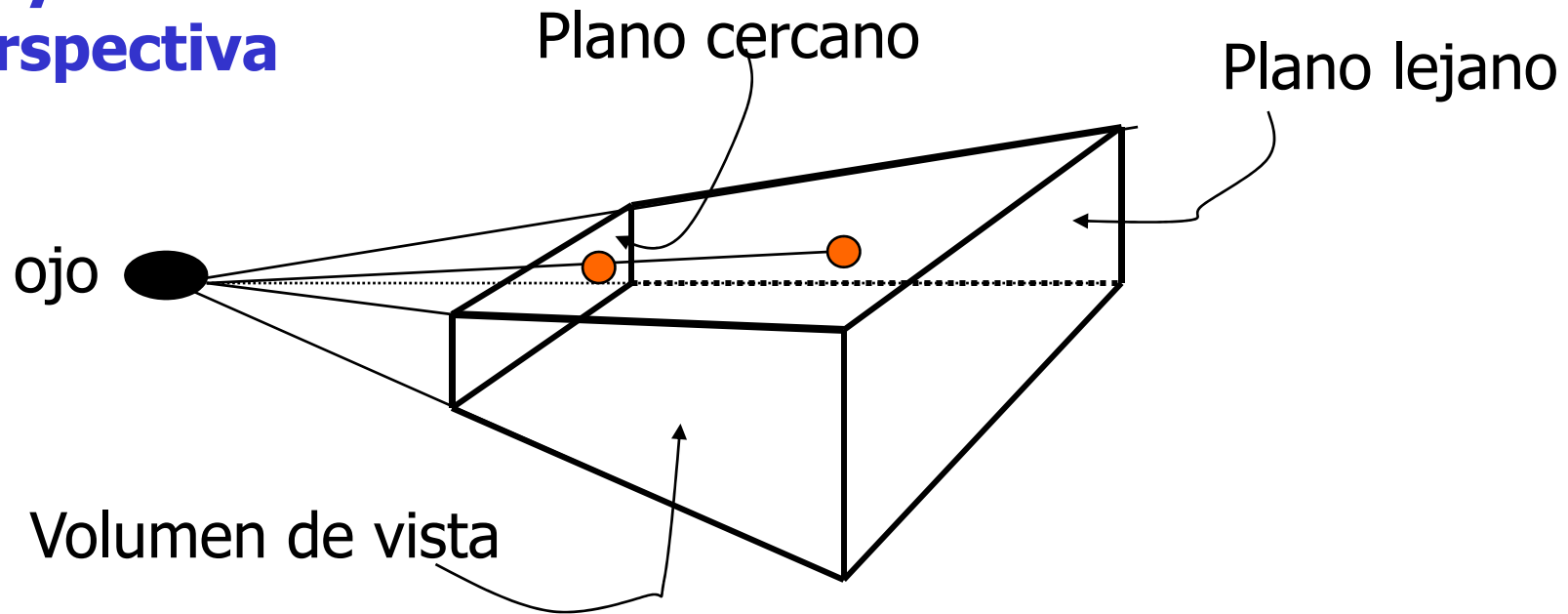


La tubería gráfica

- ❑ Diseñamos nuestro gráfico en un sistema de coordenadas.
- ❑ Posicionamos la **cámara** y elegimos el tipo de **proyección** que deseamos (ortogonal o perspectiva).
- ❑ Con esto definimos el **volumen de vista**. La parte del gráfico que caiga dentro del volumen será visible. El resto del gráfico no será visible.

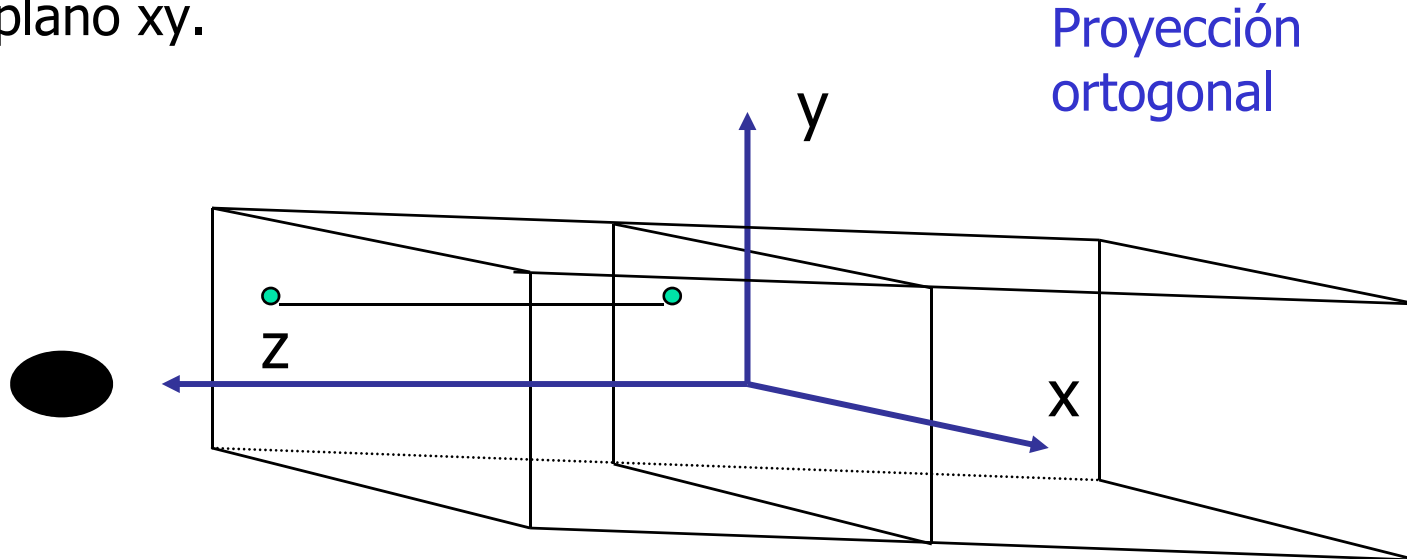
La tubería gráfica

Proyección perspectiva



2D versus 3D

- Los gráficos 2D son un caso particular de los gráficos 3D, en los que la componente z es 0.0.
- La cámara está en $(0.0, 0.0, 1.0)$, mirando a lo largo del eje de las z 's (dirección de las z 's negativas), el plano lejano y cercano son paralelos al plano xy .





La tubería gráfica

- La tubería gráfica de OpenGL trabaja con matrices.
- Cada vértice del gráfico pasa por una serie de transformaciones de matrices hasta que se visualiza en pantalla. En particular.
 - ✓ **Matriz de modelado y vista**
 - *Matriz de modelado*: recoge las distintas transformaciones (traslaciones, rotaciones, etc..) que ha sufrido el objeto.
 - *Matriz de vista*: recoge el sistema de coordenadas fijado por la cámara.
 - ✓ **Matriz de proyección**: Proyecta la escena 3D sobre un plano 2D (plano cercano), de acuerdo a la proyección elegida.
 - ✓ **Matriz del puerto de vista**: La imagen proyectada pasa a visualizarse en la parte de la ventana especificada por el puerto de vista.

La tubería gráfica

