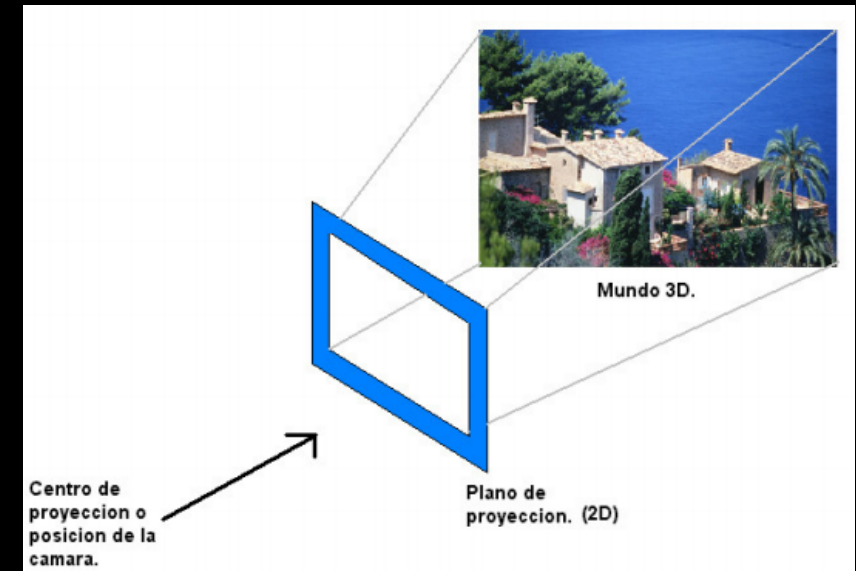


4.PROYECCIONES Y CÁMARA SINTÉTICA

PROYECCIONES

- Las proyecciones se encargan de realizar el “recorte” de la información en nuestro espacio y su procesamiento para visualización en un plano 2D.
- Es como si se viese detrás de una cámara, se tiene una **posición del observador(cámara)**, un **plano de proyección** y agrega un factor de **inclinación**
- Pasamos de coordenadas del mundo de 3D a coordenadas 2D con ayuda de las líneas de proyección y el plano de proyección



Tenemos que proyectar la geometría en un plano de proyección . Como el plano es obviamente 2D y suele situarse por convención en $Z=0$ (plano XY), La idea es proyectar primero para “discretizar “. Por discretizar entendemos pasar del mundo real (float o double) al entero (integer) que es el de los pixels de nuestro monitor.

TIPOS DE PROYECCIONES

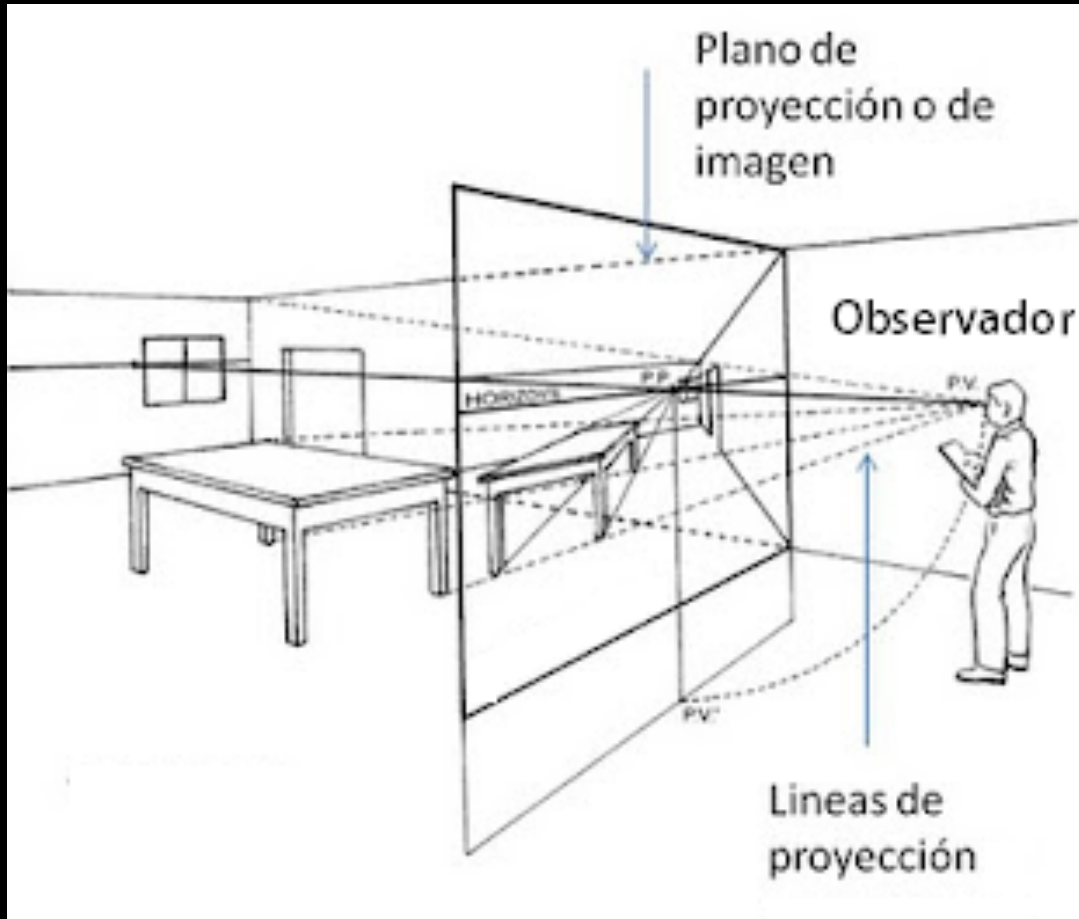
Nos centraremos en **proyecciones planares**. En esta se define una dirección de visión que va desde el observador hasta el objeto en cuestión

- Tenemos dos tipos de proyecciones
 - Ortogonal
 - Perspectiva

Sus elementos son:

- **Punto de proyección**, es el lugar donde se encuentra el observador. También llamado foco o posición
- **Líneas de proyección**, son líneas rectas que emanan del punto de proyección a la información de la escena.
- **Plano de proyección**, es un plano que se encuentra entre el punto de proyección y la información de la escena. Las líneas de proyección que pasen por el plano, representarán a la información que será visualizada en el dispositivo de despliegue en dos dimensiones.
- **Volumen de visualización**, es el espacio dentro de nuestro escenario que podrá ser visualizado.

TIPOS DE PROYECCIONES



Contamos con un mundo 3D que estamos observando desde una determinada posición . Podemos pensar en el observador como una cámara filmando nuestro mundo virtual. Ese punto es el centro de proyección . Evidentemente el mundo es tridimensional pero su proyección en un plano (plano de proyección) es bidimensional. Este plano es nuestro frame buffer antes de dibujar o bien la pantalla del monitor.

```
glm::mat 4 projection;
```

PROYECCIÓN ORTOGONAL

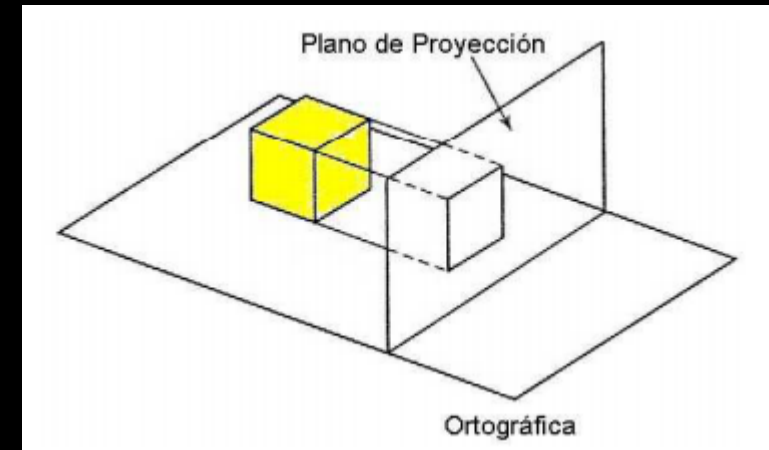
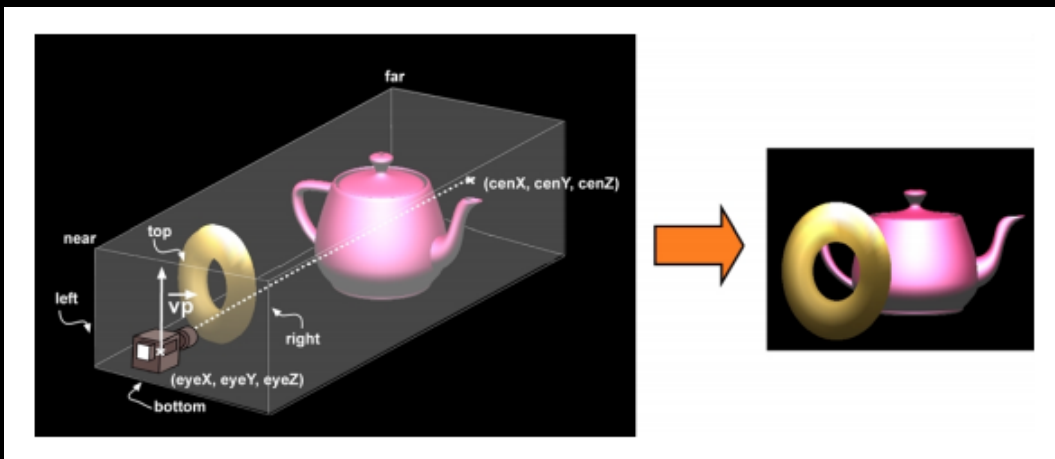
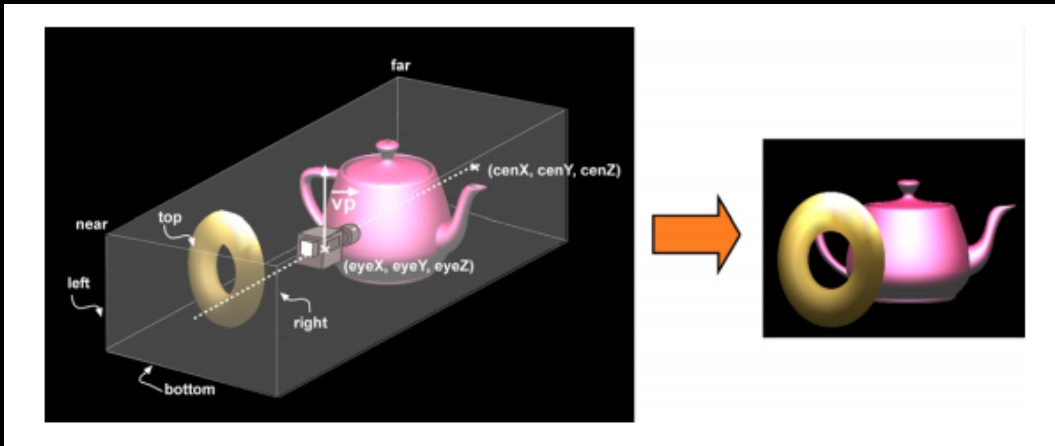
Para el caso de la proyección ortogonal la analogía de la cámara no es tan evidente, porque en realidad estamos definiendo una caja, o volumen de visualización alrededor del **ojo** de la cámara (esto es, podemos ver detrás) Para hacer esto, llamamos a:

`glm::ortho(left, right, bottom, top, near, far)`

CARACTERÍSTICAS:

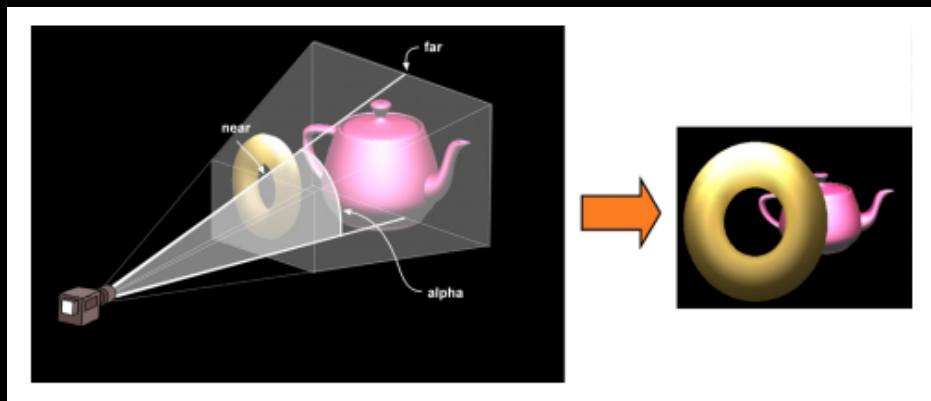
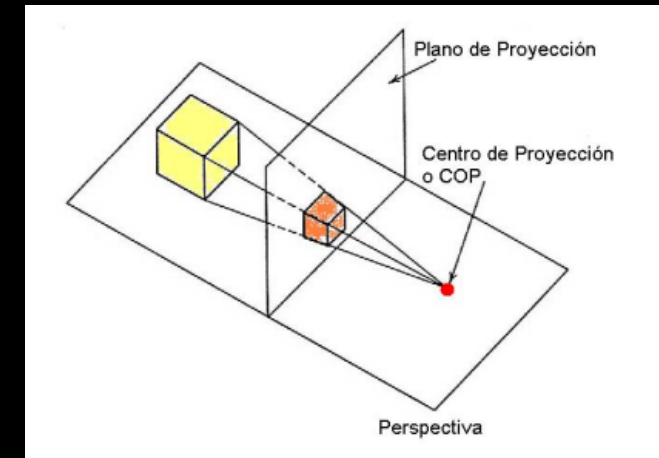
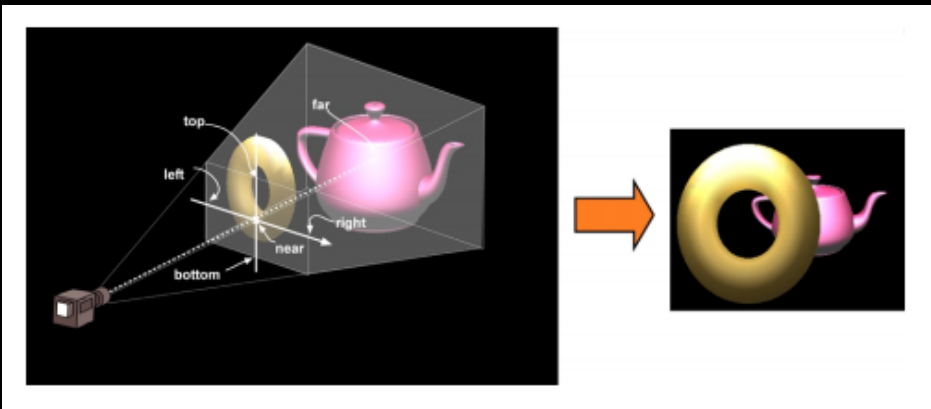
- El centro de proyección se encuentra en el infinito(COP).
- Las líneas de proyección son perpendiculares al plano de proyección y paralelas entre sí.
- No preserva las dimensiones reales de los objetos.
- No se toma en cuenta la profundidad

PROYECCIÓN ORTOGONAL



Este tipo de proyecciones no preserva las dimensiones reales de los objetos según la distancia hasta ellos. Es decir si nos acercamos o alejamos de ellos no se producen cambios de tamaño, con lo cual el realismo no es total.

PROYECCIÓN EN PERSPECTIVA



Podemos observar una proyección perspectiva con un solo punto de fuga (COP). Todas las líneas de proyección emanan de él y se dirigen hasta el objeto intersectando el plano de proyección. Como podemos observar las líneas paralelas no son percibidas como tal.

PROYECCIÓN EN PERSPECTIVA

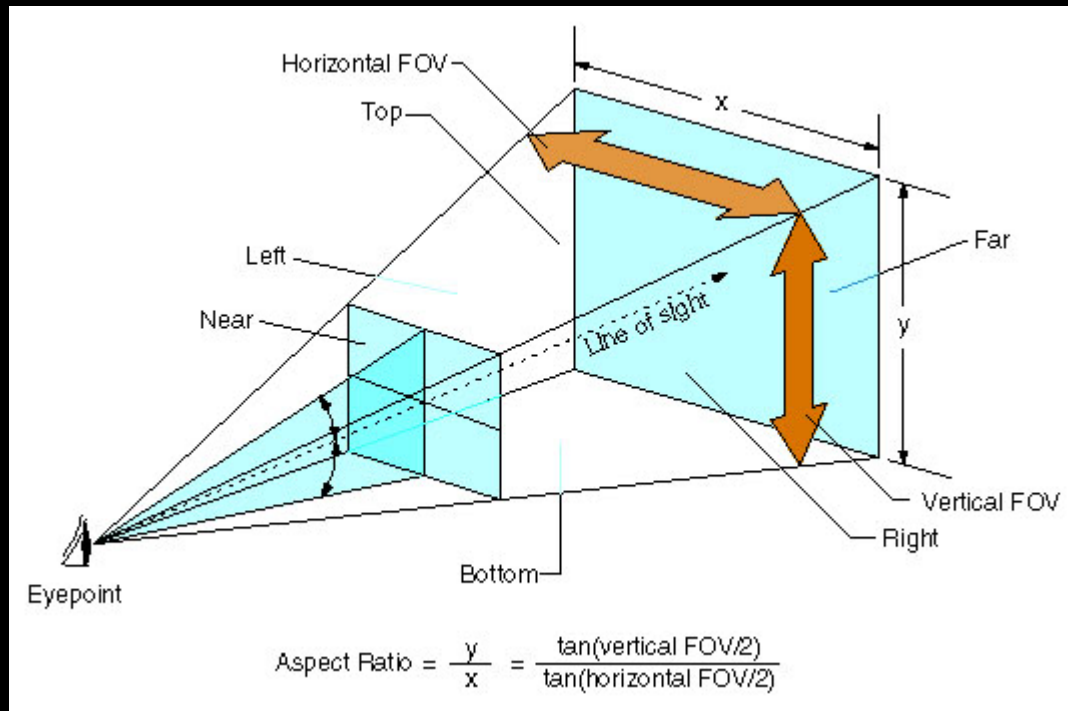
Una manera más simple de definir la proyección perspectiva es mediante la función
`glm::perspective(45.0f, (GLfloat)width / (GLfloat)height, 1.0f, 150.0f);`

Donde:

- **fov** corresponde al ángulo de apertura del campo de visión en la dirección de y.
- **aspect** define la relación entre el ángulo de visión horizontal y vertical (es la anchura / altura) típicamente debería ser de las dimensiones de la ventana.
- **near** la distancia mínima a la que deben estar los objetos para salir en la foto.(no usar 0)
- **far** la distancia máxima (ambas siempre positivas).
- Se genera una matriz de 4x4 usada en el shader para transformar puntos.
- **La relación entre far / near debería ser lo más cercana posible a 1.

PROYECCIÓN EN PERSPECTIVA

`glm::perspective(fov, aspect, near, far)`



CARACTERÍSTICAS:

- Respeta la profundidad
- Distorsiona los objetos
- La distancia entre el centro de proyección y el plano de proyección es finita.
- Muestra un efecto visual realista salvo factores de escalamiento
- Las proyecciones en perspectiva producen vistas más realistas pero en general no mantienen las distancias ni los ángulos relativo

CÁMARA SINTÉTICA

- La Cámara son nuestros ojos virtuales. Todo lo que ella vea será proyectado, discretizado y finalmente mostrado en nuestra ventana. Podemos imaginar que de la cámara emana el volumen de visualización de forma que se traslada con ella. Los parámetros a definir en cuanto a la cámara son:

- **Posición XYZ**
- **Orientación**
- **Dirección “UP”**

Tenemos que definir no sólo la posición de la cámara (o donde está), sino también hacia dónde mira y con qué orientación (no es lo mismo mirar con la cara torcida que recta... aunque veamos lo mismo). Para hacer esto, basta con modificar la matriz ModelView para mover toda la escena de manera que parezca que hemos movido la cámara.

CÁMARA SINTÉTICA

- La librería GLM nos facilita esto con la función:

glm::LookAt(vec3(posX, posY, posZ), vec3(viewX, viewY, viewZ), vec3(up_X, up_Y, up_Z));

Donde:

- pos corresponde a la posición de la cámara,
- view corresponde al punto hacia donde mira la cámara
- up es un vector que define la orientación de la cámara.

