

proyeccion-y-ventana.pdf



PruebaAlien



Informática Gráfica



3º Grado en Ingeniería Informática

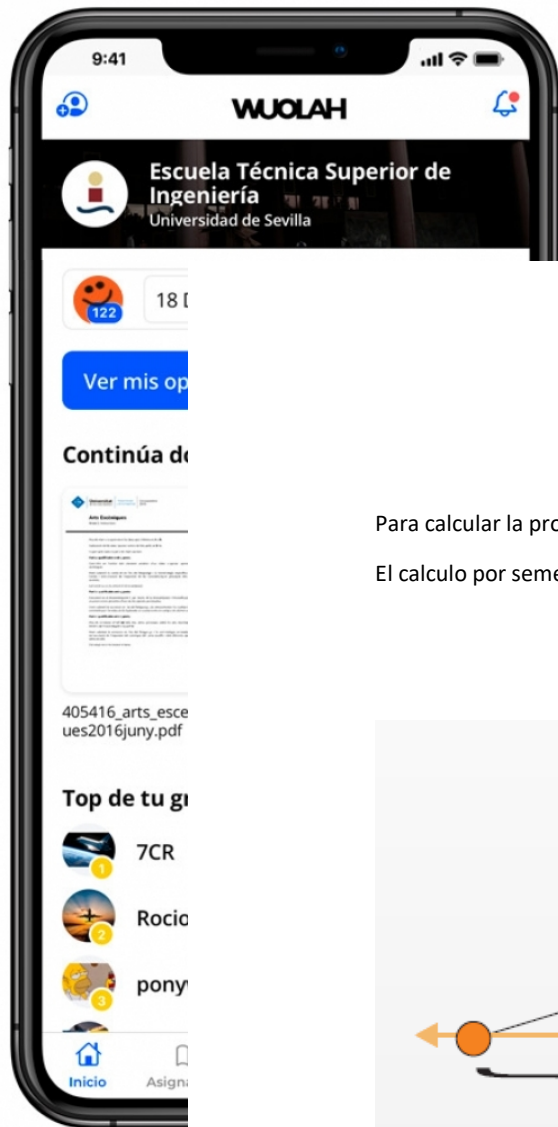


Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





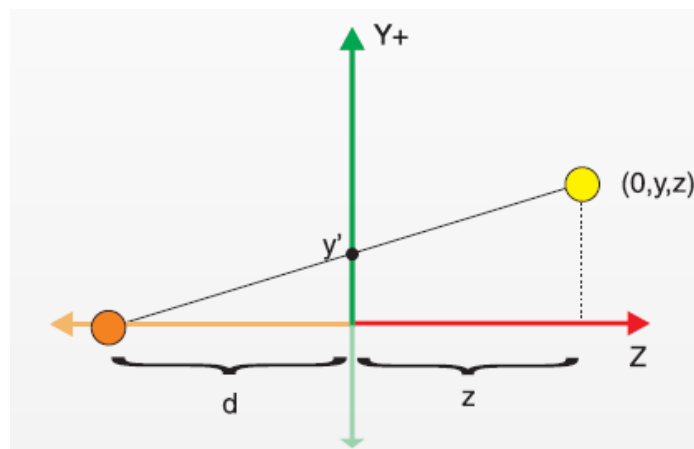
Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



Para calcular la proyección de perspectiva usamos:

El calculo por semejanza de triángulos:

$$\frac{y'}{d} = \frac{y}{z+d} \rightarrow y' = \frac{y * d}{z+d} \rightarrow y' = y * \left(\frac{1}{\frac{z}{d} + 1} \right)$$



El cálculo por ecuaciones paramétricas:

$$x' = x - (x * u); \quad y' = y - (y * u) \quad z' = z - (z + d) * u \text{ con } 0 \leq u \leq 1$$

De tal forma que la proyección $Z=0 \rightarrow Z'=0$

$$x' = x * \frac{d}{z+d}; \quad y' = y * \frac{d}{z+d}; \text{ (ESTO ES LO QUE NECESITO SABER PARA CALCULARLO)}$$

La proyección paralela nos permite medir, mientras que en una perspectiva no nos deja medir.

Despues con los resultados que nos da lo convertimos en una matriz 4x4 $(x,y,z,w) \rightarrow (x,y,0,1) =$

$$(x' = x * \frac{d}{z+d}, y' = y * \frac{d}{z+d}, 0, 1); \quad w = z/d + 1$$

Acortamiento perspectivo

Los objetos que están mas lejos se ven mas pequeños, es decir cuanto mas lejos esta, mas pequeña es la proyección.

Puntos de fuga

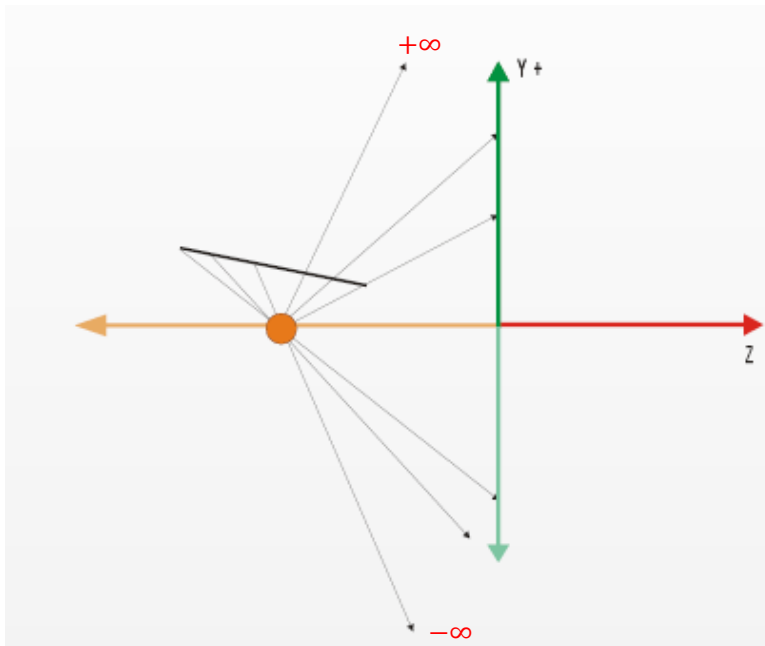
Dada 2 líneas paralelas, esas líneas convergerán en un punto llamado **punto de fuga**.

Confusión de vista

Es una inversión, de tal forma que si el objeto esta detrás del centro de proyección, el objeto proyectado se vera invertido, y en caso de que el objeto este delante del centro de proyección, el objeto se vera en la misma orientación.

Distorsión tipológica

Esto lo que provoca es que dado un objeto en el que uno de los extremos este delante del centro de proyección y otro extremo este detrás del centro de proyección, lo que provoca que al proyectar el objeto no representa la realidad, ya que proyecta 2 semirectas infinitas matemáticamente, una que se proyecta en el lado positivo, a partir de una posición positiva hasta el mas infinito y otro que se proyecta en el lado negativo, desde un punto negativo hasta el menos infinito.



Para definir los parámetros de la proyección necesitamos:

- El centro de proyección
- Los planos de corte
- Y una ventana

Con estos parámetros lo que definimos es el volumen de vista

Centro de proyección

Es el punto a través del cual atraviesan todos los proyectores, es decir donde convergen todos los proyectores.

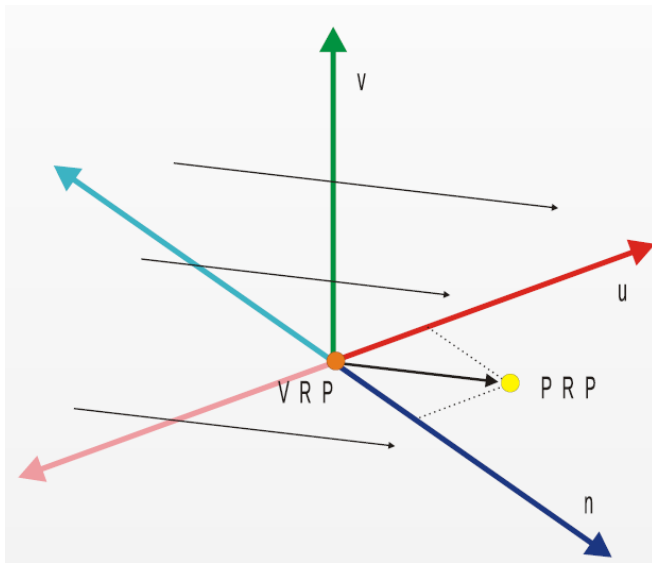
Centro de proyección en una paralela

Los proyectores son paralelos a la dirección del centro de proyección, de tal forma que convergen en el infinito en una proyección paralela.

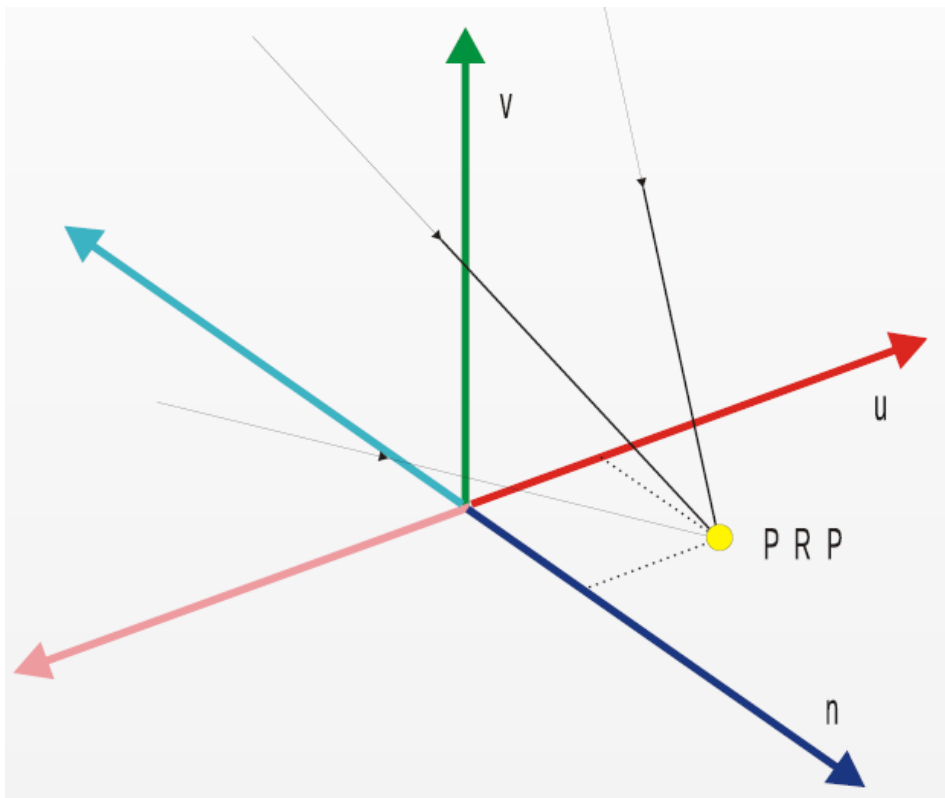


**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

Centro de proyección (PRP) (paralela)



Centro de proyección (PRP) (perspectiva)

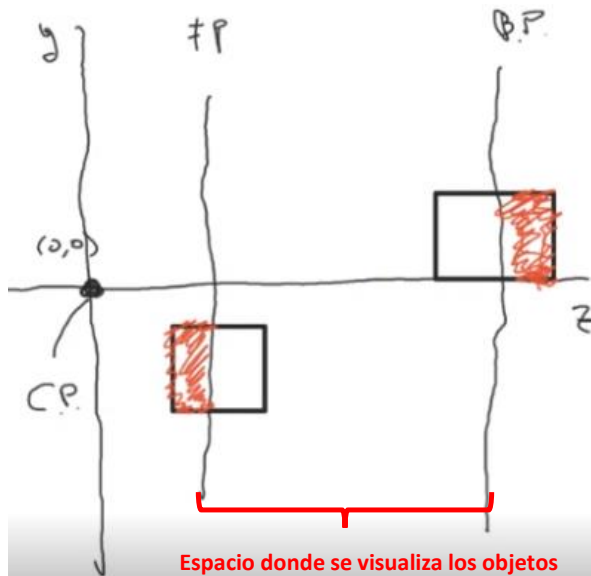


Para realizar los planos de corte necesitare 2 planos:

- Front Panel (FP) o plano delantero:
- Back Panel (BP) o plano trasero:



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



De tal forma que todo objeto que no esté en ese espacio, no se visualizara, en este ejemplo, lo que hace es cortar el objeto dejando solo la parte que esta dentro del espacio (FP y BP).

El **plano delantero** siempre tiene que estar delante del centro de proyección (CP), porque sino nos produciría los problemas de **distorsión topológica**. De tal forma que el **plano delantero** nos evita que pueda haber cosas delante y detrás del **centro de proyección (distorsión topológica)**.

El **plano delantero hace 2 cosas**, una evitar la distorsión topologica y dos es donde se proyecta.

El **plano trasero** hace lo mismo que el **plano delantero**, pero corta o elimina información que este detrás del plano, manteniendo lo que este delante del plano.

La función que tiene el **plano trasero** es que evita que objetos que se cargen en memoria, se vean muy lejos (en un puñado de pixeles), de tal forma que todo lo que este detrás del panel trasero no se mostrara. Ejemplo: un objeto que tiene 1 millon de triángulos, que están a una distancia tan grande que no se puede ni apreciar el objeto (se muestran en unos pocos pixeles), con lo cual no tiene sentido cargar un objeto complejo que se vera en pocos pixeles.

La distancia entre el plano delantero y trasero va de 0 a 1.

Ventana

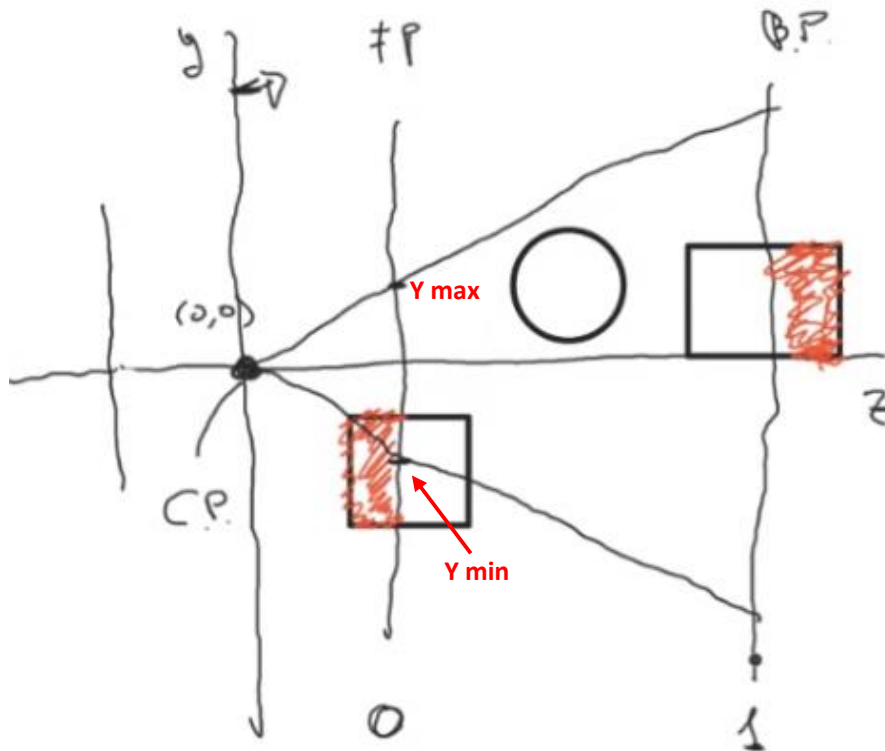
Es donde se visualizara los objetos, para ello vamos a calcular 2 esquinas:



Para definir una ventana:

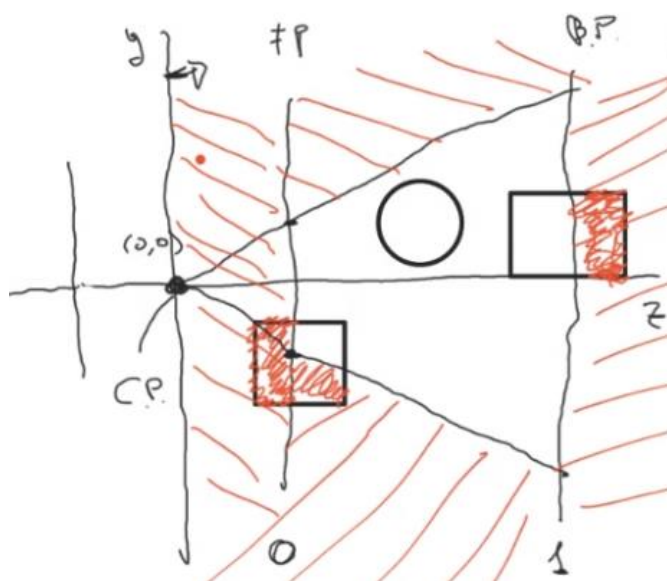
(CON Y)

Con el plano delantero vamos a definir una ventana, para ello hay que calcular el (y máxima) y el (y mínima)



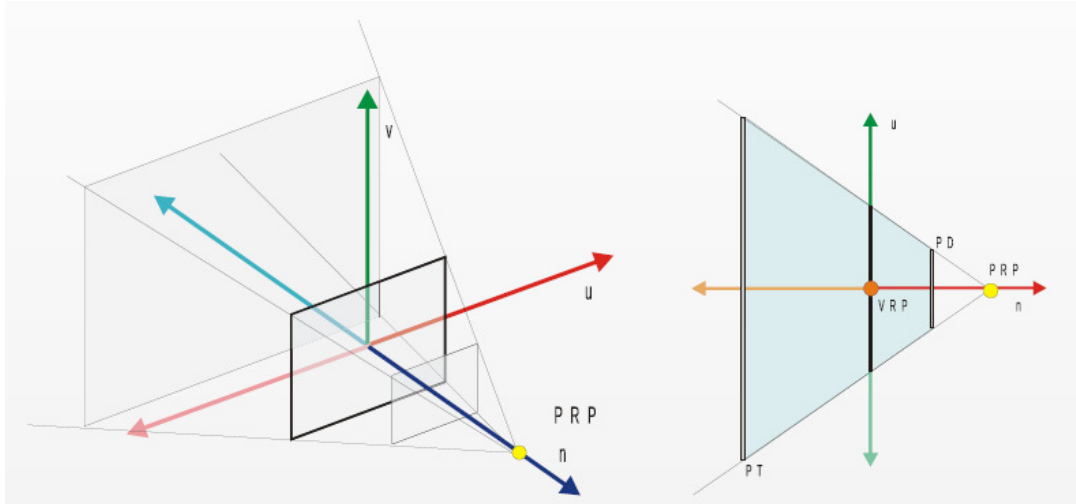
De tal forma que formamos la ventana, a través de los rayos que pasan por el centro de proyección, limitando la altura en (Y).

Resultado: (solo lo que esta dentro de la ventana se visualizara, el resto no)



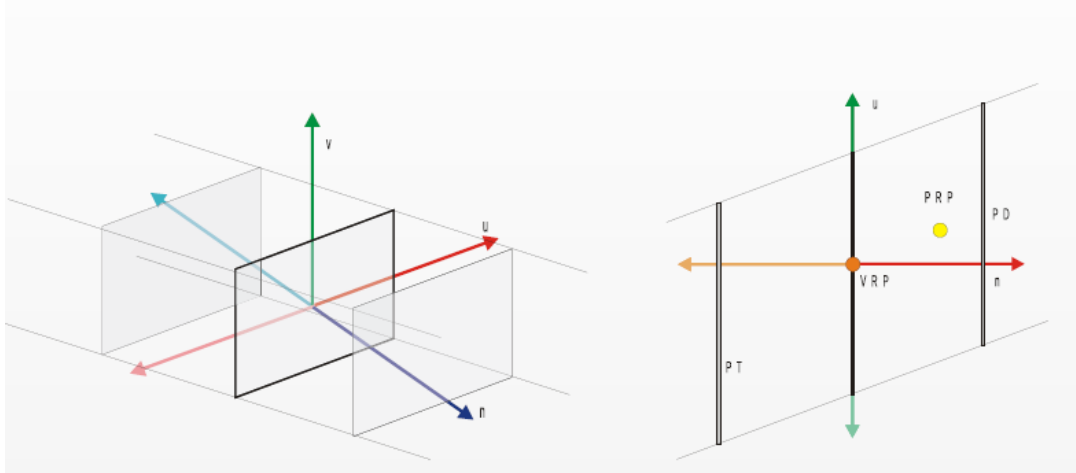
(CON X)

E lo mismo pero con el eje x, de tal forma que el resultado seria una pirámide cortada, llamada **frustum**.



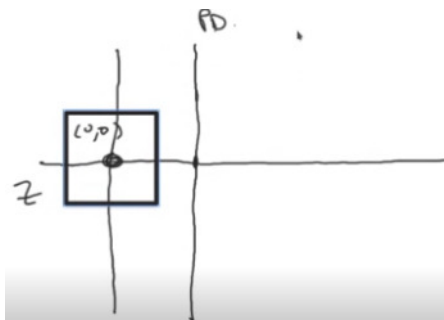
En el caso de la paralela el **frustum** es como un rectángulo:

► Volumen de visión (paralela)

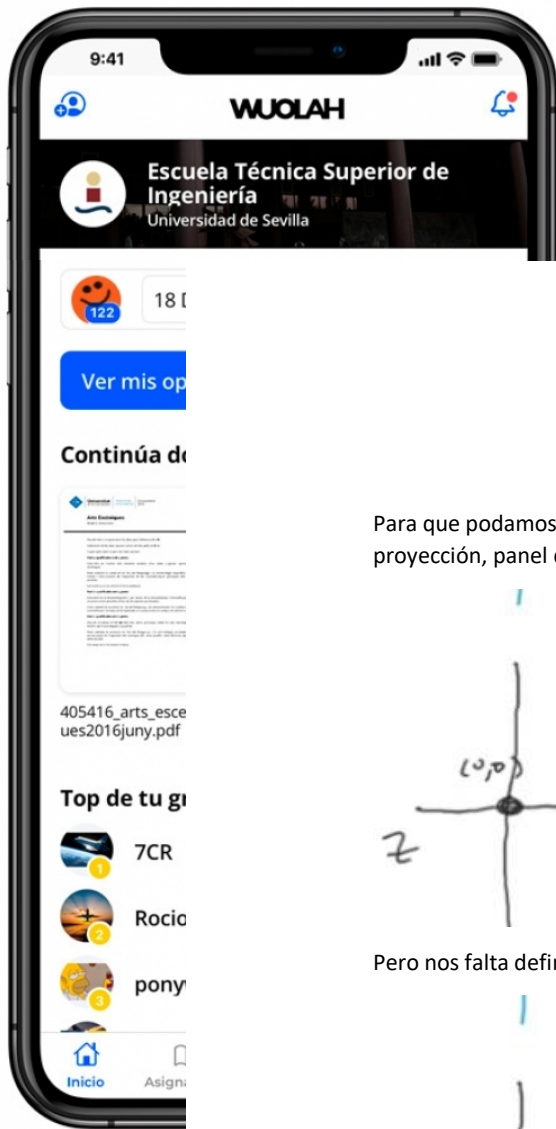


Ejemplo:

Si tenemos un objeto en la coordenada (0,0) y el centro de coordenadas también en el (0,0):



Y tenemos el Plano Delantero en 1.

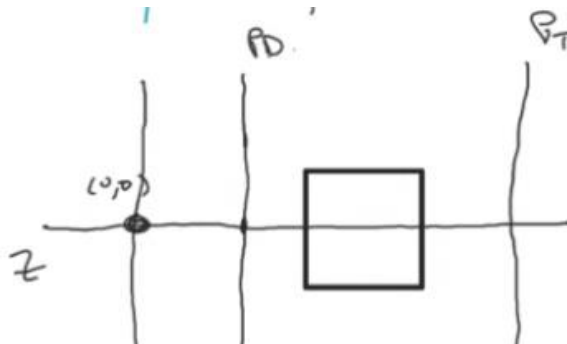


Descarga la APP de Wuolah.

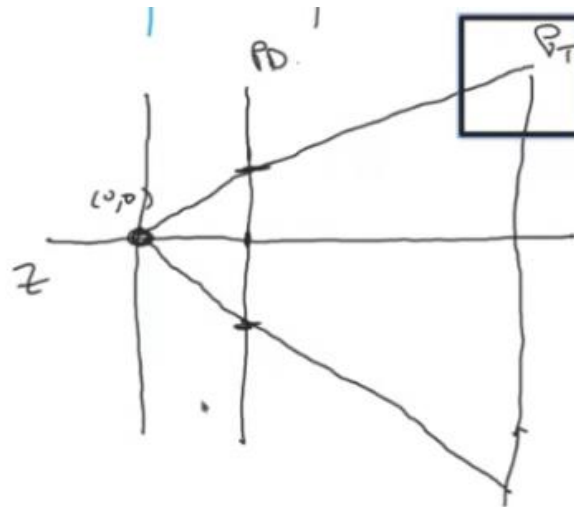
Ya disponible para el móvil y la tablet.



Para que podamos ver el objeto hay que desplazar el objeto, no la proyección (centro de proyección, panel delantero, ...)



Pero nos falta definir los límites de la ventana. Definimos los límites de la Y y la X:

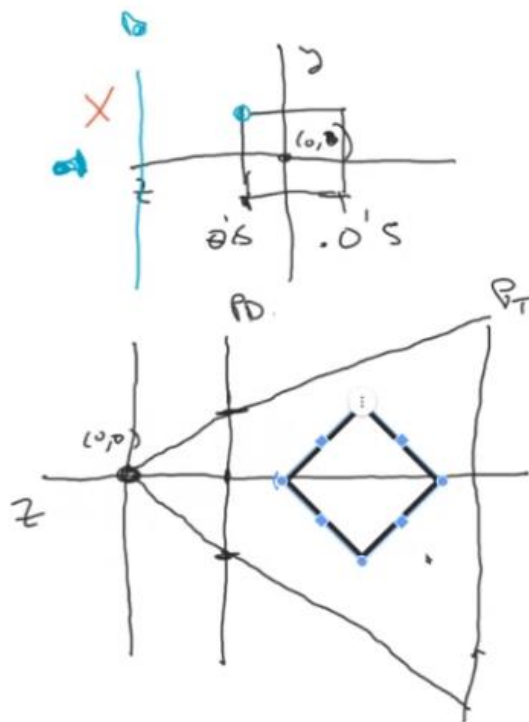


Con lo cual cuando movemos la cámara lo que se está moviendo en realidad es el objeto.

Entonces si queremos que el observador cee una esquina lo que hace es rotar el objeto 45 grados para ver la esquina.

1. Roto el objeto en el sentido contrario al del observador en el centro de coordenadas
2. Desplazo el objeto a la ventana

Resultado:



Lo que parece que hace (-45 grados)

Lo que hace realmente (45 grados)

Con lo cual estas transformaciones las hace la `modelView`, porque modela y aplica las transformaciones al objeto simulando el movimiento de la cámara.

Primero van el modelado y después la cámara, de tal forma que en la pila entra primero la cámara y después el modelado.

`glFrustum`, recibirá los parámetros X mínima y máxima e Y mínima y máxima, con lo cual definimos la ventana.

- Front plane > 0
- Back plane > Front plane

En mi código tengo `FRONT_PLANE_PERSPECTIVE=(X_MAX-X_MIN)/2`