

Vista tridimensional en OpenGL

Antes de empezar ... (1/2)

- Antes de definir un tipo de proyección, tenemos que haber cambiado (si es necesario) el modo de matriz actual.
- Por lo general, el modo va a ser `GL_PROJECTION`, dado que vamos a estar trabajando con la matriz de proyección de OpenGL.

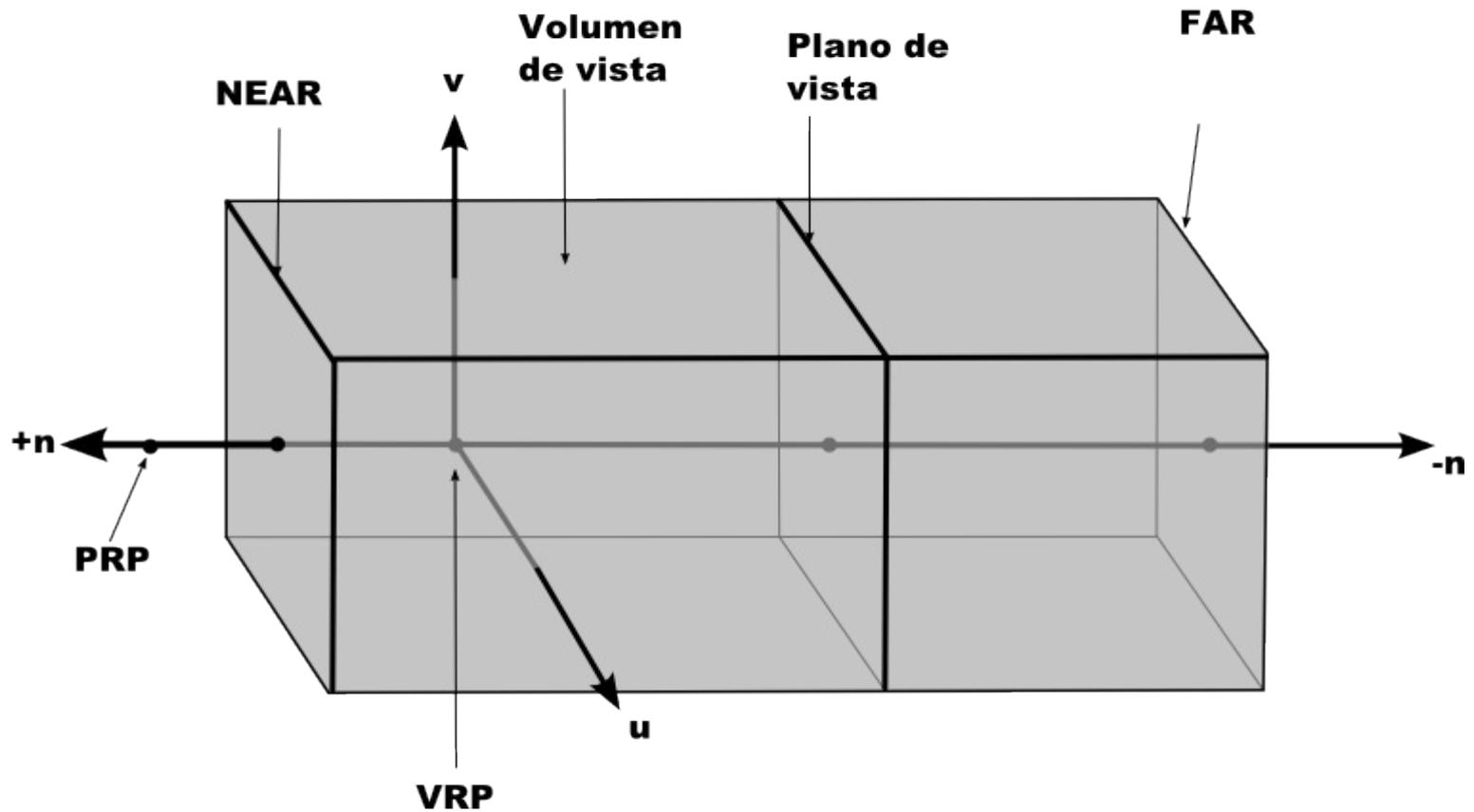
Antes de empezar ... (2/2)

- La inicialización de la matriz actual de trabajo (matriz de proyección), se realiza a través de la función `glLoadIdentity()` ;
- De esta manera cargamos la matriz identidad, lo cual es equivalente a colocar el punto de vista en el origen, mirando hacia el eje negativo de Z.

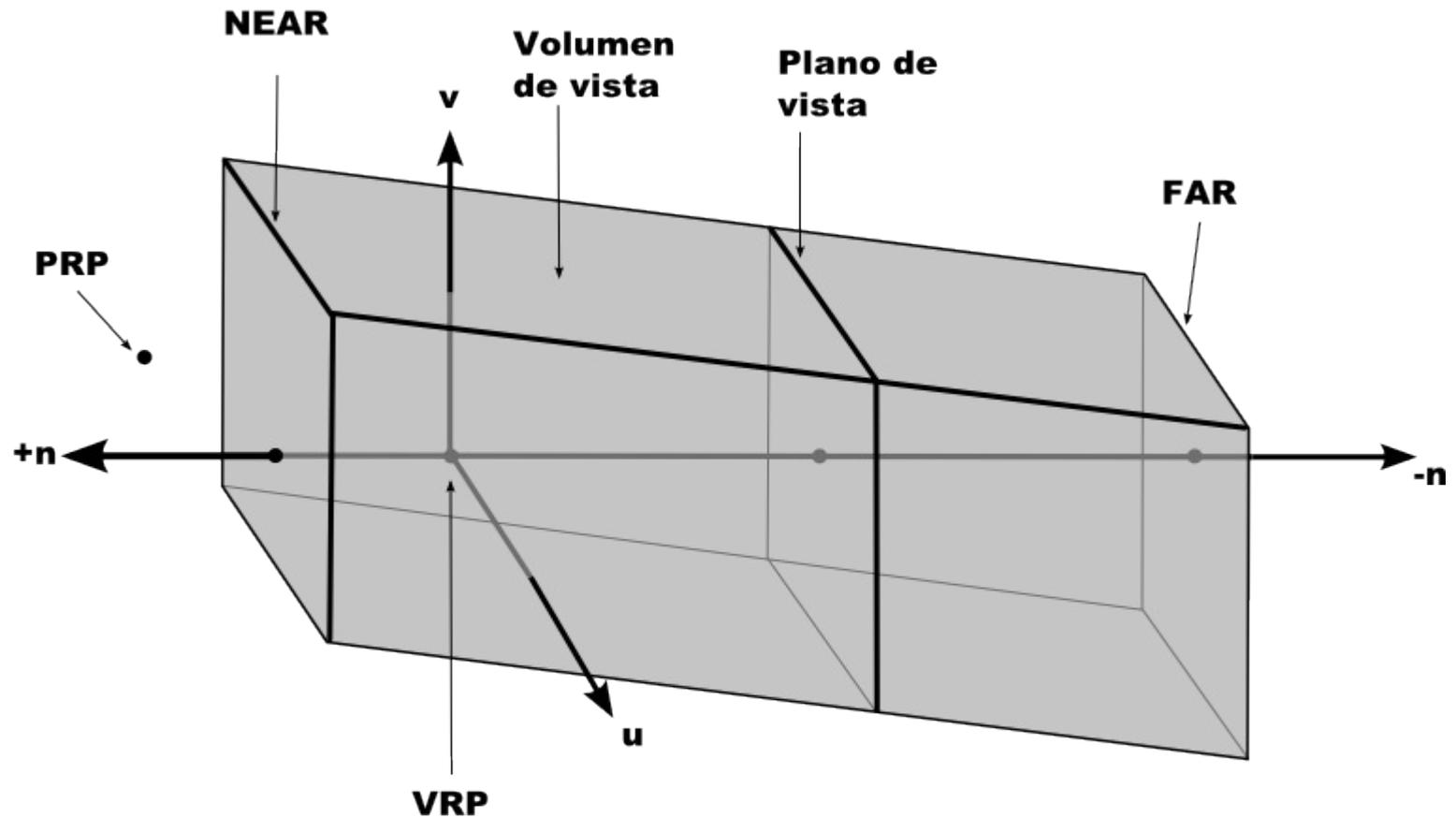
Definición de una proyección paralela

- `glOrtho(left, right, btm, top, near, far) ;`
- `left, right`: coordenadas de los planos de recorte laterales.
- `btm, top`: coordenadas de los planos de recorte superior e inferior.
- `near, far`: distancias hacia los planos de recorte anterior y posterior (pueden ser valores negativos o positivos).

Ejemplo de una proyección paralela (1/2)



Ejemplo de una proyección paralela (2/2)



A tener en cuenta ... (1/2)

- Asumiendo que el punto de vista se encuentra en el origen, los puntos (**left**, **btm**, **-near**) y (**right**, **top**, **-near**) definen las coordenadas del plano de recorte anterior, y son mapeados a las esquinas inferior izquierda y superior derecha de la pantalla.
- El valor de **-far** indica la posición del plano de recorte posterior.

A tener en cuenta ... (2/2)

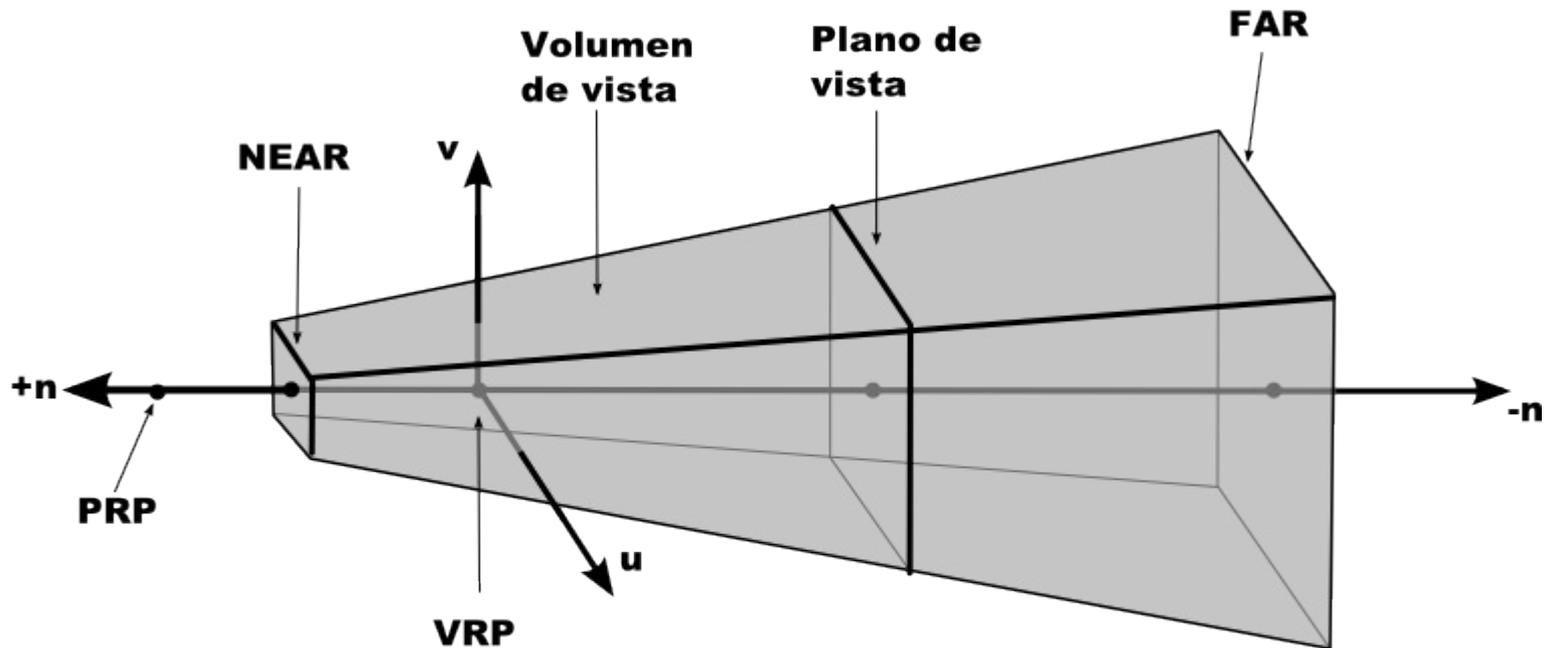
- Este tipo de definición para la perspectiva, nos da la posibilidad de definir una proyección no simétrica (proyección paralela oblicua), algo que no es muy común dado que en la mayoría de los casos se va a cumplir:

$$\text{left} = -\text{right} \quad \text{y} \quad \text{btm} = -\text{top}$$

Definición de una proyección en perspectiva

- `glFrustum(left, right, btm, top, near, far) ;`
- `left, right`: coordenadas de los planos de recorte laterales.
- `btm, top`: coordenadas de los planos de recorte superior e inferior.
- `near, far`: coordenadas de los planos de recorte anterior y posterior (ambos valores TIENEN QUE SER POSITIVOS!!).

Ejemplo de una proyección perspectiva



A tener en cuenta ... (1/3)

- Asumiendo que el punto de vista se encuentra en el origen, los puntos (**left**, **btm**, **-near**) y (**right**, **top**, **-near**) definen las coordenadas del plano de recorte anterior, y son mapeados a las esquinas inferior izquierda y superior derecha de la pantalla.
- El valor de **-far** indica la posición del plano de recorte posterior.

A tener en cuenta ... (2/3)

- Este tipo de definición para la perspectiva, nos da la posibilidad de definir una proyección no simétrica, algo que no es muy común dado que en la mayoría de los casos se va a cumplir:

$$\text{left} = -\text{right} \quad \text{y} \quad \text{btm} = -\text{top}$$

A tener en cuenta ... (3/3)

- Para cuando estas condiciones se cumplen, también se puede utilizar el método **`gluPerspective (fovy , aspect , near , far)`**
- **`fovy`**: ángulo de visión (en grados). Distancia angular entre el plano inferior y superior.
- **`aspect`**: relación entre el ancho y el alto (x/y)
- **`near , far`**: coordenadas de los planos de recorte anterior y posterior (ambos valores POSITIVOS!!).

Subclases de las proyecciones planas

- En el primer nivel de clasificación de las proyecciones planas, se hace una dicotomía entre proyecciones paralelas y en perspectiva
- Luego de este punto, las diferentes subclases se diferencian en:
 - Posición relativa del punto de vista con respecto a los ejes.
 - Dirección de proyección.
 - Simetría o asimetría de los planos de recorte.

Proyección isométrica (1/2)

- Para lograr una proyección isométrica primero que nada tenemos que setear que tenemos proyección paralela, como vimos anteriormente.
- Lo siguiente que tenemos que hacer es aplicar las siguientes transformaciones:
 - Rotar 35° sobre el eje y.
 - Rotar 45° sobre el eje x.
 - Proyectar en el plano $z=0$.

Proyección isométrica (2/2)

- La matriz para lograr esto es la siguiente:

$$\begin{bmatrix} 0.7071 & 0.0 & 0.7071 & 0.0 \\ 0.4082 & 0.8166 & -0.4082 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$