

Sistemas Coordinados

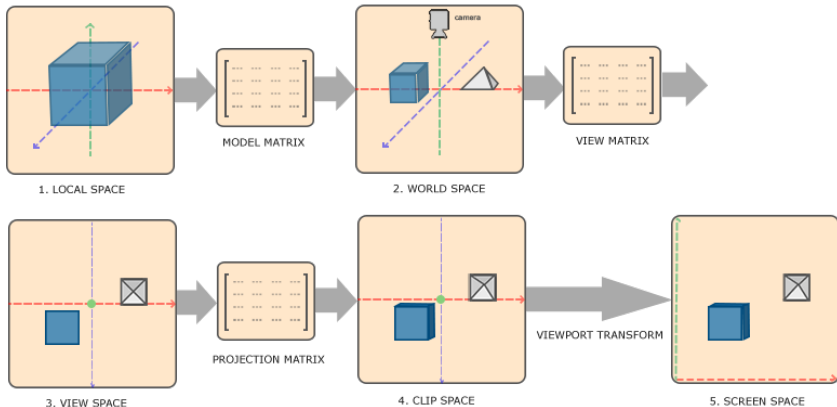
Reynaldo Martell

21 de agosto de 2019

Sistemas Coordenados.

- Desde que a OpenGL se le proporcionan coordenadas 3D hasta que aparecen en la pantalla, las coordenadas espaciales pasan por diferentes sistema de referencia.
- La transición entre los diferentes sistemas de referencia se lleva a cabo mediante transformaciones que se realizan antes de mostrarse en pantalla.
- La ventaja de realizar las transformaciones entre cada sistema coordenados se debe a que pueden hacer fácilmente ciertas operaciones en un sistema coordenados.
- Normalmente existen 5 tipos de sistemas coordenados.
 - Local space (Object space).
 - World space.
 - View space.
 - Clip space.
 - Screen space.

Transformaciones entre sistemas de coordenadas.



Local Space.

- Es el sistema de coordenadas relativo a su origen, es donde el objeto empieza.
- Imaginar que empezamos a modelar un cubo, el origen del cubo probablemente este en $(0, 0, 0)$
- Todos los vértices están referidos al sistema local, origen del modelo.
- En nuestros ejemplos el rango que se han explicado, los rectángulos están en el rango de $-0.5, 0.5$ en el origen 0.

World Space.

- Imaginar cuando queremos importar nuestros objetos, todos ellos se desplegaran uno encima de otro es decir su origen es $(0, 0, 0)$.
- Lo deseado es definir una posición para cada objeto que estén en un mundo mas grande.
- Todas las coordenadas son relativas al mundo.
- Colocar el objeto en escala, posición y orientación.
- Las coordenadas locales del objeto son transformados al sistema de coordenadas del mundo mediante una matriz llamada model matrix.
- Model matrix es una matriz que escala, traslada y rota los objetos en un lugar del mundo.

View Space

- El espacio del observador es lo que a menudo se refiere a la cámara de OpenGL.
- A veces se refiere como espacio de la cámara.
- Todas las coordenadas son relativas al mundo.
- El espacio del observador es el resultado de transformar del sistema de coordenadas del mundo a lo que está enfrente del usuario.
- Esto se realiza regularmente con trasladar/rotar la escena enfrente de la cámara.
- La combinación de las transformaciones que se realizan para posicionar los objetos enfrente de la cámara se almacena en una matriz llamada view matrix.

Clip Space

- OpenGL espera que los vértices se encuentran en un rango de valores.
- Los valores que no se encuentren en este rango son recortados y descartados del renderizado.
- Esto es conocido como clip space.
- Regularmente lo que se hace es manejar nuestro propio espacio de recorte por ejemplo -100 a 100 y después aplicarle una transformación para que sean recortadas las coordenadas en el espacio de -1 a 1 .
- La matriz que llamaremos matriz del proyección es la que especifica este rango para posteriormente hacer la normalización.

Clip Space

- Por ejemplo si la matriz de proyección especifica un rango entre -1000 a 1000 el punto (1250, 500, 750) no será visible debido a que x no esta en el rango de -1000 a 1000.
- Si una primitiva por ejemplo un triangulo esta fuera del volumen de recorte OpenGL reconstruye el triangulo para que muestre el pedazo que cae dentro del volumen de recorte.
- Este volumen de recorte es comúnmente llamado Frustum.
- Todas las coordenadas que se encuentran dentro de la caja del Frustum son mostrados en pantalla.

Clip Space

- Una vez que todos los vértices son transformados al espacio de recorte se realiza una operación final llamada perspective division.
- Esta operación consiste en dividir las coordenadas x , y y z del vector por la componente w de su coordenada homogénea.

Screen Space

- Después las coordenadas son mapeadas a coordenadas de la pantalla usando `glViewport()`.

Proyecciones

- Las proyecciones se encargan de realizar el proceso de recorte y de convertir la información de los vértices de la escena a coordenadas en dos dimensiones.
- Existen varias propiedades que se le asocian a una proyección y de éstas depende como se visualizara nuestra escena.

Tipos de proyecciones

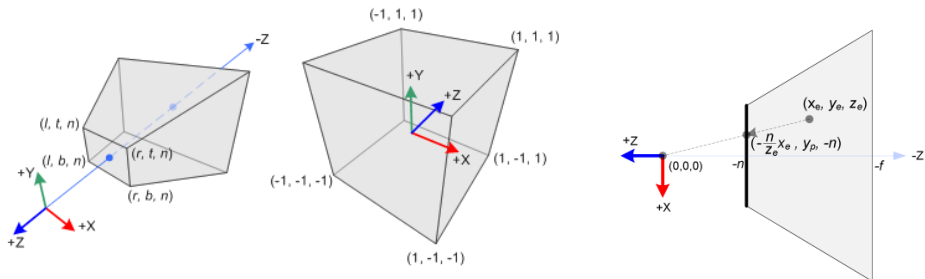
- Proyección ortogonal.
- Proyección en perspectiva.

Elementos de una proyección

Si importar cual sea el tipo de proyección, ambas tienen los siguientes elementos:

- Punto focal o punto de proyección: Lugar geométrico donde se encuentra el origen de apertura de proyección (Origen del observador).
- Líneas de proyección. Son líneas rectas que van desde el origen de apertura de la proyección a los vértices de la geometría.
- Plano de proyección: Plano que se encuentra entre el punto en coordenadas espaciales y el punto focal.
- Volumen de proyección. Es el espacio en el cual se visualiza nuestra escena.

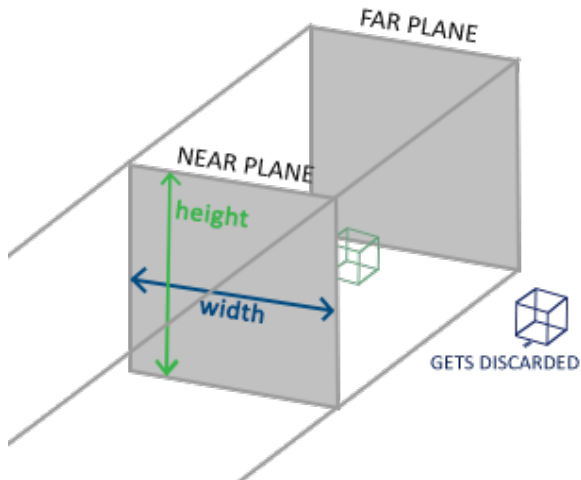
Elementos de una proyección



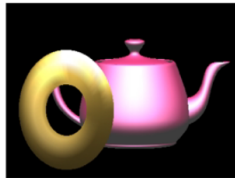
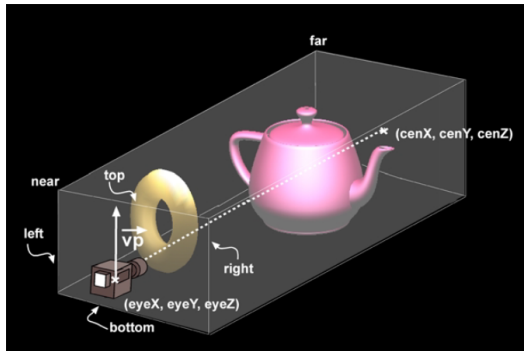
Proyección ortogonal

- La matriz de una proyección ortogonal define una especie de caja como frustum, la cual define el espacio de recorte.
- Los vértices que quedan fuera de la caja son recortados.
- Para especificar la matriz de proyección ortográfica es necesario definir ancho, alto y largo del frustum.

Proyección ortogonal



Proyección ortogonal



Matrix de Proyección ortogonal

$$M_{ortogonal} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

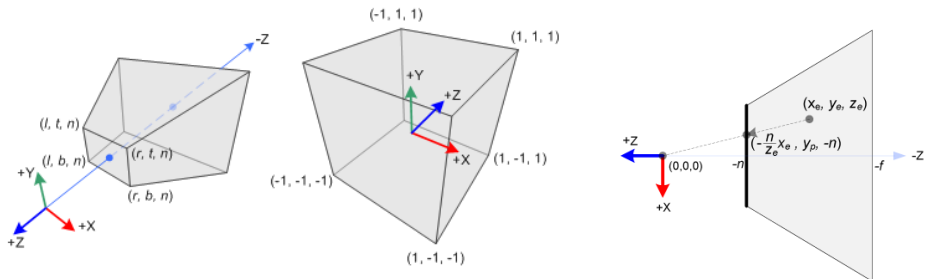
Proyección ortogonal OpenGL

- Para crear la matriz de proyección ortográfica.
`glm::ortho(-1.0f, 1.0f, -1.0f, 1.0f, 0.1f, 100.0f);`
- Los primeros dos parámetros especifican las coordenadas izquierda y derecha del frustum.
- Los parámetros 3 y 4 especifican la parte inferior y superior.
- Los parámetros 5 y 6 especifican la distancia entre el plano cercano y plano lejano.

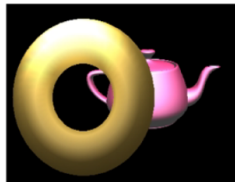
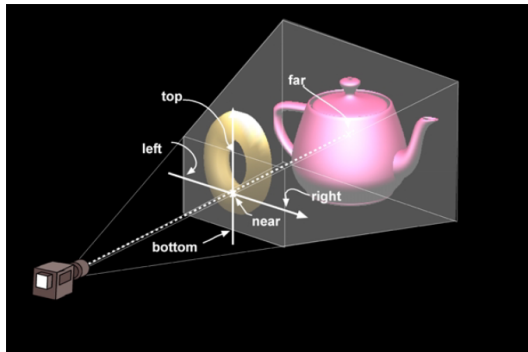
Proyección en perspectiva

- En gráficos para computadora el efecto de perspectiva se da cuando los objetos que están mas lejos de la escena aparecen mas pequeños.
- Esta técnica es usada para darle profundidad a los objetos.
- La matriz de proyección no solo mapea las coordenadas a un rango de valores al espacio de recorte.
- También manipula el valor w de cada vértice, de tal forma que los vértices que se encuentran mas alejados en profundidad z aparecen mas lejanos.

Proyección en perspectiva



Proyección en perspectiva



Matrix de Proyección en perspectiva

$$M_{perspective} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Proyección en perspectiva OpenGL

- La matriz en perspectiva también se puede crear con glm de la siguiente manera.
glm::perspective(45.0f, (float)width/(float)height, 0.1f, 100.0f);
- **glm::perspective** crea una pirámide que define el espacio visible.
- Todo lo que quede fuera de este volumen es recortado.
- El primer parámetro define el field of view (FOV).
- El segundo parámetro es el aspect ratio que se calcula dividiendo el ancho y alto de la pantalla.
- El tercer y cuarto parámetro es la distancia que existe en el plano cercano y plano lejano. en perspectiva.