**Las vistas en computación grafica**

**¿Qué es la vista perspectiva y en qué situaciones se aplica?**

Vista Perspectiva

La vista perspectiva se basa en crear una ilusión de profundidad y volumen determinada por un punto de fuga en el infinito del plano sobre el que se proyectan los lados hacia el eje “z”, lo que distorsiona la percepción de las dimensiones, pero mantiene una representación más cercana a la realidad de los objetos o escenas a reproducir.

Esta vista es comúnmente utilizada en la computación gráfica para la creación de modelos 3d, de videojuegos (igualmente en 3d), de animaciones y en general para representar cualquier estructura, plano u objeto 3d sobre un plano 2d.

**¿Qué es la vista ortográfica y en qué situaciones se aplica?**

Vista ortogonal

La vista ortográfica, por su parte, se basa en la representación de la realidad sobre un plano bidimensional totalmente perpendicular al ángulo de visión, es decir, la proyección se encuentra exactamente a 90 grados de los ojos del espectador y por esta razón se llama ortogonal (como los vectores perpendiculares).

Esta vista es muy útil para detallar cada cara de un objeto 3d (frontal, lateral, etc.) debido a que no se distorsionan las dimensiones sino que se mantiene la escala de las medidas originales y por ello resulta una herramienta muy práctica y eficaz para el dibujo técnico en donde se crean planos de estructuras u objetos que se quieran construir en la realidad, ya sea a mano o asistidos por programas de diseño, y de igual forma se utiliza en videojuegos 2d que no requieran demostrar profundidad sino un entorno completamente plano.

**¿Cómo se calcula una vista en perspectiva en la computación gráfica y qué parámetros se utilizan en su cálculo?**

En computación gráfica se calcula la vista perspectiva mediante los siguientes parámetros:

Ángulo de visión: se refiere a la apertura de la cámara respecto a la escena, a mayor ángulo mayor es lo que se visualiza de la escena.

Relación de aspecto: Se refiere a la proporción entre alto y ancho del plano

Punto de mira: Se refiere al centro de interés sobre el cuál se fija la vista

Posición del observador: Es el punto en el espacio tridimensional sobre el cual se posiciona la cámara u observador

Estos parámetros se asignan a la cámara como caracteres numéricos como se muestra a continuación

75° es el ángulo de visión, window.innerWidth / window.innerHeight se refiere a que la relación de aspecto va a estar dada por la ventana del computador en la cual se visualice el archivo, 0.1 y 1000 son el centro de interés pues nada menor a la distancia de 0.1 unidades de la cámara y nada mayor a la distancia de 1000 unidades de la cámara se va a renderizar.

const camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);

Para calcular la vista en perspectiva, se utiliza una matriz de proyección. Esta matriz se encarga de transformar los puntos 3D de la escena en puntos 2D en la imagen final. La matriz de proyección se calcula a partir de los parámetros mencionados anteriormente y se aplica a cada punto de la escena para proyectarlo en la imagen final.

**¿Cuáles elementos intervienen en la configuración de las vistas referidas y qué significado tiene cada uno de ellos en THREE.js?**

EnTHREE.js la vista se configura mediante los siguientes elementos.

Cámara (Camera): La cámara define el punto de vista desde el cual se observa la escena. Puede ser como se mencionó anteriormente perspectiva u ortogonal

Escena (Scene): la escena constituye a todos los objetos que se van a visualizar o renderizar y la iluminación de estos. Los objetos se agregan a la escena mediante el método add().

Renderizador (Renderer): El renderizador es el encargado de tomar la información de la cámara y de la escena, y de crear una imagen en la pantalla. El renderizador tiene varios métodos que permiten configurar su comportamiento, como .setSize(), para definir el tamaño de la ventana, y .setPixelRatio(), para definir la relación de aspecto de los píxeles.

Luz (Light): La luz define la iluminación de la escena. Existen varios tipos de luces en THREE.js que varían según el objeto o la escena a iluminar

**Referencias**

[**https://medium.com/@az.martinez4799/c%C3%A1mara-en-webgl-366be536d37c**](https://medium.com/@az.martinez4799/c%C3%A1mara-en-webgl-366be536d37c)

[**https://www.thecube3danimation.com/blog-1/2019/2/6/vista-en-perspectiva-u-ortogrfica-al-esculpir#:~:text=La%20vista%20en%20Perspectiva%20es,humana%20y%20la%20distancia%20focal**](https://www.thecube3danimation.com/blog-1/2019/2/6/vista-en-perspectiva-u-ortogrfica-al-esculpir#:~:text=La%20vista%20en%20Perspectiva%20es,humana%20y%20la%20distancia%20focal)**.**

[**https://tips.clip-studio.com/es-es/articles/807#:~:text=El%20dibujo%20en%20perspectiva%20es,tridimensional%20en%20un%20plano%20bidimensional**](https://tips.clip-studio.com/es-es/articles/807#:~:text=El%20dibujo%20en%20perspectiva%20es,tridimensional%20en%20un%20plano%20bidimensional)**.**

[**https://www.calameo.com/read/001409093df0f825f67e3**](https://www.calameo.com/read/001409093df0f825f67e3)

[**https://threejs.org/docs/index.html?q=perspe#manual/en/introduction/Creating-a-scene**](https://threejs.org/docs/index.html?q=perspe#manual/en/introduction/Creating-a-scene)