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

Es un tipo de proyección en la que los objetos parecen más pequeños a medida que se alejan del observador y se reducen en tamaño aparente . En otras palabras, las cosas más cercanas a ti parecen más grandes que las cosas más lejanas se ven más pequeñas. Este efecto se logra mediante el uso de un punto de fuga o varios puntos de fuga y un amplio campo de visión.

Existen varias aplicaciones para la vista desde la perspectiva , desde la representación de objetos en el arte y la arquitectura hasta la visualización de escenas en gráficos de computadora y videojuegos. También se utiliza en fotografía, cine y televisión para dar la impresión de profundidad en una imagen.

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

Es un tipo de proyección en la que los objetos se representan en una escena sin perspectiva, todas las líneas paralelas son dibujadas de manera paralela en la imagen. De acuerdo a esto podemos decir que, es una vista de una imagen en dos dimensiones sin tener en cuenta la profundidad de los objetos en la escena o imagen.

La vista ortográfica se aplica en una representación precisa de los objetos en la escena sin que la perspectiva distorsione su tamaño o forma. Por ejemplo, en el diseño técnico o arquitectónico, donde se necesita representar objetos de manera precisa y a escala. También se utiliza en la animación 2D, en la que se necesita mantener el tamaño y la forma consistentes en todas las escenas o frames.

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

La perspectiva en computación gráfica viene hacer un constructor que toma cuatro argumentos: fov (campo de visión), aspect (relación de aspecto), near (distancia de la cámara al plano cercano) y far (distancia de la cámara al plano lejano). Estos valores definen los parámetros de la proyección de perspectiva de la cámara.

Esta clase también tiene una serie de propiedades como fov, zoom, near, far, aspect, view, filmGauge y filmOffset que se utilizan para configurar y calcular la proyección de la cámara.

Además, la clase perspectiva tiene varios métodos como copy, setFocalLength, getFocalLength, getEffectiveFOV, getFilmWidth, getFilmHeight, setViewOffset, clearViewOffset, updateProjectionMatrix y toJSON que se utilizan para realizar operaciones en la cámara, como copiar las propiedades de otra cámara, configurar el enfoque, calcular la longitud focal, obtener el campo de visión efectivo, establecer y limpiar el offset de vista, actualizar la matriz de proyección y convertir la cámara en un objeto JSON.

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

En la vista de la perspectiva en THREE.js está como un constructor llamada PerspectiveCamera esta viene con sus parámetros que son fov, aspect, near y far.

Para la vista ortogonal en THREE.js esta como el constructor llamado OrthographicCamera est toma seis argumentos opcionales que establecen los límites de la cámara ortográfica: left, right, top, bottom, near y far. También inicializa algunas propiedades de la cámara, como zoom, view, y llama al método updateProjectionMatrix(), que actualiza la matriz de proyección de la cámara.

Bibliografía:

<https://www.thecube3danimation.com/blog-1/2019/2/6/vista-en-perspectiva-u-ortogrfica-al-esculpir>

<https://threejs.org/docs/#api/en/cameras/PerspectiveCamera>

[src/cameras/PerspectiveCamera.js](https://github.com/mrdoob/three.js/blob/master/src/cameras/PerspectiveCamera.js)

[src/cameras/OrthographicCamera.js](https://github.com/mrdoob/three.js/blob/master/src/cameras/OrthographicCamera.js)

<https://threejs.org/docs/index.html#api/en/cameras/OrthographicCamera>