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

La vista perspectiva es una técnica de dibujado que permite representar objetos tridimensionales en una superficie bidimensional como puede ser en una hoja o en la pantalla del pc, dándole así una ilusión de profundidad y distancia creando así una imagen que parece tener volumen y espacio. Su aplicación puede ser muy variada ya que sirve tanto como para dibujo, animación, computación grafica e incluso para el desarrollo de videojuegos.

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

La vista ortogonal es una técnica de graficacion que permite mostrar objetos tridimensionales en dos dimensiones, consiste en proyectar las diferentes vistas del objeto desde ángulos ortogonales es decir rectos entre sí, esta vista se puede representar de manera frontal, superior y lateral del objeto las cuales muestras una perspectiva diferente del objeto y se organizan en un plano donde se pueda visualizar el objeto desde todos los ángulos posibles. Su aplicación suele ser usada en el diseño industrial, la arquitectura y varias ramas de la ingeniera.

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

La proyección de la perspectiva es un proceso matemático y esta se calcula utilizando una matriz de proyección la cual se define por los siguientes para metros:

1. Campo se visión (FOV): Es el Angulo de visión de la cámara, que determina cuanto del mundo 3D se proyectara en la imagen 2D.
2. Relación de aspecto (aspect ratio): Es la relación entre la anchura y la altura de la imagen de salida, por ejemplo: Una relación 16:9 habla de 16 unidades de alto y 9 de ancho
3. Distancia de plano cercano y lejano (near and fa plane): Son las distancias desde la cámara al plano cercano y al plano lejano de la escena, respectivamente y todo lo que se encuentra dentro de estas distancias se proyectan en la imagen de salida.
4. Posición y orientación de la cámara (camera position and orientation): la posición de la cámara determina desde donde se está viendo el objeto y la orientación hacia donde está apuntando la cámara.
5. Transformación de perspectiva (perspective transformation): esta trasformación se aplica al vértice del objeto 3D para transformarlo desde su posición en la escena 3D a su posición den la imagen 2D.

La matriz de proyección se calcula combinando estos parámetros en una matriz 4x4 y se aplica a cada vértice del objeto 3D para calcular su posición en la imagen 2D

1. ¿Cuáles elementos intervienen en la configuración de las vistas referidas y qué significado tiene cada uno de ellos en THREE.js?
2. En THREE.js, la configuración de las vistas referidas implica varios elementos importantes. Los elementos principales que intervienen en la configuración de las vistas referidas son:
3. Cámara (Camera): La cámara define la posición, orientación y proyección de la vista. THREE.js proporciona varios tipos de cámaras, incluyendo la perspectiva y la ortográfica. La cámara determina cómo se verá la escena desde el punto de vista del usuario.
4. Escena (Scene): La escena es un contenedor que contiene todos los objetos que se mostrarán en la vista. Puede incluir geometrías, luces, cámaras y otros objetos.
5. Renderizador (Renderer): El renderizador es el encargado de tomar la información de la cámara y la escena y convertirla en una imagen 2D que se puede mostrar en la pantalla. THREE.js proporciona varios tipos de renderizadores, incluyendo el WebGLRenderer y el CanvasRenderer.
6. Luz (Light): Las luces en THREE.js son objetos que afectan la iluminación de la escena. Pueden ser direccionales, puntuales, de área, ambientales, entre otros.
7. Material (Material): Los materiales en THREE.js definen cómo se verán las superficies de los objetos en la escena. Pueden ser transparentes, reflectantes, metálicos, entre otros

Cada uno de estos elementos tiene un significado importante en la configuración de las vistas referidas en THREE.js. La cámara determina la perspectiva desde la cual se verá la escena, la escena contiene los objetos que se mostrarán, el renderizador convierte la información de la cámara y la escena en una imagen 2D, las luces afectan la iluminación de la escena y los materiales definen cómo se verán las superficies de los objetos. Juntos, estos elementos permiten crear vistas referidas en THREE.js que pueden ser personalizadas y ajustadas para satisfacer las necesidades del usuario.

1. Crear dos ejemplos (*perspectiva.htm* y *ortografica.htm*) para THREE.js en que se visualice el modelo (no renderizado) de un mismo escenario (una figura cualquiera, cubo, esfera, pirámide, o cualquiera otra generado a partir de los puntos vértices y no con las geometrías básicas predefinidas)
2. Relacionar las fuentes bibliográficas y/o web grafía utilizadas en el desarrollo del presente trabajo.

WEB GRAFIA

<https://es.wikipedia.org/wiki/Computaci%C3%B3n_gr%C3%A1fica>

<http://computaciongrafica-ulat.blogspot.com/p/proyeccion-y-perspectiva.html>