**METODOS DE OCULTAMIENTO THREE.JS**

Los métodos de ocultamiento son métodos que sirven para mejor la optimización de un renderizado 3D, estos métodos se encargan de cargar un objeto siempre y cuando el programador lo requiera o dados unos parámetros según el método implementado. Algunos de estos métodos son:

* **.visible:**

Permite ver o quitar un objeto con true or false

Ventajas: Este es un método sencillo y rápido para ocultar objetos. Puede ser útil para ocultar o mostrar objetos de forma rápida.

Desventajas: No es adecuado si necesitas animar la visibilidad gradualmente o si quieres mostrar u ocultar objetos de manera más compleja.

* **.opacity:**

Método que cambia los valores de trasparencia de un objeto a ciertos valores dados por el usuario

Ventajas: Puedes usar la propiedad de opacidad para hacer que un objeto sea gradualmente más transparente, lo que puede crear efectos de transición suaves.

Desventajas: No oculta completamente un objeto, solo lo vuelve transparente. No es adecuado si deseas ocultar un objeto por completo.

* **.remove():**

Contrario al .add(), remuve o quita un objeto de la escena que pida .

Ventajas: Eliminar un objeto de la escena lo oculta por completo. Puedes volver a agregarlo más tarde si es necesario.

Desventajas: Puede ser ineficiente si necesitas ocultar y mostrar objetos con frecuencia, ya que agregar y eliminar objetos de la escena puede ser costoso en términos de rendimiento.

* **.renderOrder:**

Establece el orden de renderizado de los objetos

Ventajas: Permite controlar el orden de renderización de objetos, lo que puede simular la ocultación relativa de objetos sin cambiar la visibilidad o eliminarlos de la escena.

Desventajas: No oculta completamente los objetos y puede ser más complejo de configurar si tienes muchos objetos en la escena.

* **.frustumCulled:**

Método que habilita o deshabilita que los objetos se rendericen fuera del campo de visión de la cámara

Ventajas: Puedes desactivar la comprobación del frustrum (frustum culling) para que un objeto nunca se oculte debido a la cámara.

Desventajas: Puede afectar negativamente el rendimiento si se utiliza en exceso, ya que los objetos fuera de la vista de la cámara seguirán siendo procesados.