

Taller

Redactar (en sus propias palabras) un documento sobre las vistas perspectiva y ortográfica en computación gráfica.

Responder entre otras cuestiones:

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

- Definición:

La vista perspectiva es aquella en donde se intenta reproducir la profundidad del espacio de una superficie plana y hacer que se vea tridimensional con los objetos de la imagen.

- Aplicaciones:

- ❖ En las pinturas, para hacer sentir profundidad en sus representaciones.
- ❖ En videojuegos, películas.
- ❖ Animación.

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

- Definición:

La vista ortográfica es aquella que utiliza una proyección cúbica de algún objeto, en donde nos presenta cada una de las partes que lo componen, de esta manera diferenciándolo de la vista perspectiva.

- Aplicaciones:

- ❖ Se usa en dibujo técnico para la creación de figuras con mayor eficacia.
- ❖ Arquitectura.
- ❖ Diseño industrial.

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

En computación gráfica la vista perspectiva se calcula utilizando una técnica llamada "proyección perspectiva" que hace una simulación de cómo el ojo del ser humano observa el mundo.

- Parámetros:

- ❖ Posición de la cámara: determina la posición en que se está observando el objeto tridimensional.
- ❖ Dirección de la cámara: determina el lugar hacia donde se está mirando.
- ❖ Ángulo de visión: define el ángulo que abarca el punto de vista.
- ❖ Relación de aspecto: define la relación del tamaño de la imagen.

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

Los elementos que intervienen en la configuración de las vistas referidas son:

1. Cámara: define el punto de vista desde donde se ve la escena.
2. Escena: aquí es donde se ubican los objetos y elementos a visualizar.
3. Renderizador: es el encargado de crear la imagen final.
4. Luces: son los elementos que dan la iluminación a la escena.
5. Geometría: define la forma y la estructura de los objetos.
6. Materiales: determinan el aspecto visual de los objetos.

5. 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 la geometrías básicas predefinidas). Incluir como mínimo ejes principales XYZ, mall de plano XZ y un componente OrbitControls.

- Vista perspectiva:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Perspectiva</title>
  <style>
    body {
      margin: 0;
      overflow: hidden;
    }
  </style>
</head>
<body>
  <script>
    const scene = new THREE.Scene();
    const camera = new THREE.PerspectiveCamera(
      75,
      window.innerWidth / window.innerHeight,
      0.1,
      1000
    );
    const renderer = new THREE.WebGLRenderer();
    renderer.setSize(window.innerWidth, window.innerHeight);
    document.body.appendChild(renderer.domElement);

    const controls = new THREE.OrbitControls(camera, renderer.domElement);

    const geometry = new THREE.Geometry();
    geometry.vertices.push(new THREE.Vector3(-1, 0, 0));
```

```

geometry.vertices.push(new THREE.Vector3(1, 0, 0));
geometry.vertices.push(new THREE.Vector3(0, 1, 0));
geometry.faces.push(new THREE.Face3(0, 1, 2));
geometry.computeFaceNormals();

const material = new THREE.MeshBasicMaterial({
  color: 0xffffff,
  wireframe: true,
});

const mesh = new THREE.Mesh(geometry, material);
scene.add(mesh);

const gridHelper = new THREE.GridHelper(10, 10);
scene.add(gridHelper);

const axesHelper = new THREE.AxesHelper(5);
scene.add(axesHelper);

camera.position.z = 5;

function animate() {
  requestAnimationFrame(animate);
  renderer.render(scene, camera);
}

animate();
</script>
</body>
</html>

```

- Vista ortográfica:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Ortografía</title>
  <style>
    body {
      margin: 0;
      overflow: hidden;
    }
  </style>
</head>
<body>
  <script>
    const scene = new THREE.Scene();
    const camera = new THREE.OrthographicCamera(
      window.innerWidth / -20,
      window.innerWidth / 20,
      window.innerHeight / 20,

```

```

    window.innerHeight / -20,
    0.1,
    1000
  );
  const renderer = new THREE.WebGLRenderer();
  renderer.setSize(window.innerWidth, window.innerHeight);
  document.body.appendChild(renderer.domElement);

  const controls = new THREE.OrbitControls(camera, renderer.domElement);

  const geometry = new THREE.Geometry();
  geometry.vertices.push(new THREE.Vector3(-1, 0, 0));
  geometry.vertices.push(new THREE.Vector3(1, 0, 0));
  geometry.vertices.push(new THREE.Vector3(0, 1, 0));
  geometry.faces.push(new THREE.Face3(0, 1, 2));
  geometry.computeFaceNormals();

  const material = new THREE.MeshBasicMaterial({
    color: 0xffffff,
    wireframe: true,
  });

  const mesh = new THREE.Mesh(geometry, material);
  scene.add(mesh);

  const gridHelper = new THREE.GridHelper(10, 10);
  scene.add(gridHelper);

  const axesHelper = new THREE.AxesHelper(5);
  scene.add(axesHelper);

  camera.position.z = 5;

  function animate() {
    requestAnimationFrame(animate);
    renderer.render(scene, camera);
  }

  animate();
</script>
</body>
</html>

```

6. Relacionar las fuentes bibliográficas y/o webgrafía utilizadas en el desarrollo del presente trabajo.

→ Asale, R.-. (s. f.). *perspectivo, perspectiva* / *Diccionario de la lengua española*. «Diccionario de la lengua española» - Edición del Tricentenario. <https://dle.rae.es/perspectivo>

→ School, C. 3. A. (2019). ¿Vista en Perspectiva u Ortográfica al esculpir? — THE CUBE. *THE CUBE*.

<https://www.thecube3danimation.com/blog-1/2019/2/6/vista-en-perspectiva-u-ortografica-al-esculpir#:~:text=La%20vista%20Ortogr%C3%A1fica%2C%20es%20aquella.un%20objeto%20a%20la%20c%C3%A1mara.>

→ *Vistas en Computación Grafica.* (s. f.). Scribd.

<https://es.scribd.com/document/292667596/Vistas-en-Computacion-Grafica>

→ MonkeyWit. (2020, 31 mayo). *Tutorial de Threejs - Parte 4: Camera | Cómo realmente funciona?* | *Three js tutorial for beginners* [Vídeo]. YouTube.

<https://www.youtube.com/watch?v=1rL589rMKa0>

7. **Publicar los tres archivos referidos anteriormente en un repositorio de nombre CG-231-B-301**