Three.js Learning Guide

1. Setting Up Your Environment

HTML & JavaScript Basics: A basic understanding of HTML, CSS, and JavaScript is essential. Brush up on these if needed.

Installing Three.js:

- Include it in your project by adding <script src='https://cdnjs.cloudflare.com/ajax/libs/three.js/r128/three.min.js'></script> in your HTML file, or install via npm with `npm install three`.

- Set up a simple HTML file and import three.js to ensure everything works.

2. Core Concepts of Three.js

Scene: The 3D space where objects are placed.

Example:

```javascript
const scene = new THREE.Scene();
```

Camera: Determines the viewer's perspective. The `PerspectiveCamera` is commonly used for 3D.

Example:

```javascript
const camera = new THREE.PerspectiveCamera(fov, aspect, near, far);
```

Renderer: Renders the scene with the camera view.

Example:

```javascript
const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);
```

3. Adding Objects

Geometry and Materials:

- Geometry: Defines the shape (e.g., BoxGeometry, SphereGeometry).
- Material: Defines appearance (e.g., MeshBasicMaterial, MeshPhongMaterial).

Example:

```javascript
const geometry = new THREE.BoxGeometry();
const material = new THREE.MeshBasicMaterial({ color: 0x00ff00 });
const cube = new THREE.Mesh(geometry, material);
scene.add(cube);
```

4. Lighting

Three.js requires light for materials like `MeshPhongMaterial` and `MeshStandardMaterial`.

Example:

```javascript
const light = new THREE.DirectionalLight(0xffffff, 1);
light.position.set(1, 1, 1);
scene.add(light);
```

5. Animation Loop

Use `requestAnimationFrame` to create an animation loop.

Example:

```javascript
function animate() {
    requestAnimationFrame(animate);
    cube.rotation.x += 0.01;
    cube.rotation.y += 0.01;
    renderer.render(scene, camera);
}
animate();
```

6. Loading Models

For complex models, use `GLTFLoader` to load `.gltf` or `.glb` files.

Example:

```javascript
const loader = new THREE.GLTFLoader();
loader.load('path/to/model.glb', function(gltf) {
    scene.add(gltf.scene);
});
```

7. Controls and Interactivity

Use `OrbitControls` to add mouse-based camera controls.

Example:

```javascript
const controls = new THREE.OrbitControls(camera, renderer.domElement);
controls.enableDamping = true;
```

8. Responsive Design

Update camera aspect ratio and renderer size on window resize:

```javascript
window.addEventListener('resize', () => {
    camera.aspect = window.innerWidth / window.innerHeight;
    camera.updateProjectionMatrix();
    renderer.setSize(window.innerWidth, window.innerHeight);
});
```

Recommended YouTube Tutorials

Three.js Crash Course - 3D Web Graphics Made Easy by Academind

https://www.youtube.com/watch?v=4e5Vg4nzM8A

Three.js Beginner's Guide - Full Course by freeCodeCamp.org

https://www.youtube.com/watch?v=YK1Sw_hnm58

Three.js Journey by Bruno Simon

https://threejs-journey.xyz/

Getting Started with Three.js by Traversy Media

https://www.youtube.com/watch?v=Q7AOvWpIVHU

Three.js and WebGL Fundamentals by The Coding Train

https://www.youtube.com/playlist?list=PLRqwX-V7Uu6ZiZxtDDRCi6uhfTH4FilpH