**THREE.js:**

**Web gl?**

**React 3 fiber.**

Three.js use GPL

We draw thing on canvas.

**Some repetative tasks:**

* **Scene:**

Scene is the whole area.

Area behind the camera + front of camera

All 3D world is is scene.

New THREE.Scene()

* **Camera:**

Camera is field/area in which we can see. Camera is perspective and the area in which we can see.

New THREE.perspeciveCamera(

65, 🡪 FOV (take camera new or far of the object according to value)

 window**.***innerWidth* **/** window**.***innerHeight***, 🡪 width and height ratio**

0.1, 🡪 near view( the thing at the distance of less than 0.1 will not be seen)

100 🡪 far view ( the thing at the distance of more than 100 will not be seen)

)

After creating camera we have to add camera into the scene.

Scene.add(camera)

Camera.postion(will) will decide the perspective, we can do in any of one dimension.

* **Mesh:🡪(geometry & Camera):**

Mesh.position.x/y/z=val; it tell the position in any axis

We can also use rotation in all direction. Suppose we have a pipe and a pencil. Pipe will take pencil as axis for rotation. Assume position of pencil in all axis then assume the rotation of pipe. It works like that.

For rotation we can use some mathematics, (1 pi means one rotation, we can divide in 4 parts as according to we need rotation like if we want to rotate only 90 degree then we can divide it by 2. For 45 degree we can use pi/4)

We have to know degree to radians or radians to degree so do some practice on geometry.

**Scaling can also be done in it.**

* **Renderer:**

It means print every timeframe pic, like we put camera and scene as arguments in the renderer then it shows about it one the screen.

Rendering will be according to the fps speed.

* **requestAnimationFrame:**

we use it inside a custome function, bcz we use anothers things with it.

* Window.requestAnimationFrame(animate) 🡪 it requestion the FPS speed of the screen
* Then we can render anything according to the requestion FPS, **like** renderer.render(scene,camera)
* As we are doing animation then we can rotate,scale anything by using mesh.
* As rendering speed will be according to the FPS speed of different devices so it will be different on different devices, so instead of fps speed we can set it on time, like in 1 second it do same rotaions on all te devices.

We will use

* Const clock= new THREE.clock()
* Mesh.rotation.x = clock.getElaspedTime();