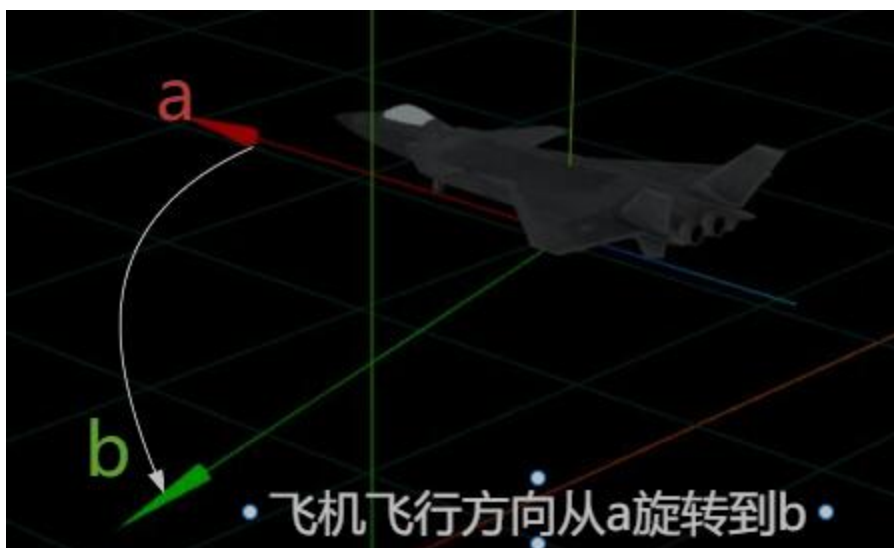


🎯 5. 四元数表示两个向量旋转

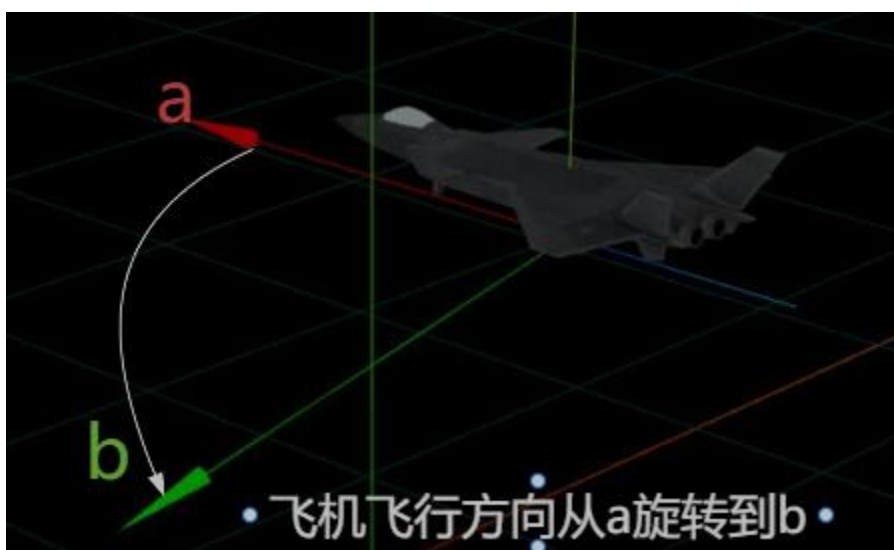


3D空间中有两个向量，一个向量向另外一个向量旋转，这个过程可以用一个四元数表示。

下面给大家介绍一个四元数 `Quaternion` 的方法 `.setFromUnitVectors(a, b)` ,计算参数a到b的旋转。

已知条件

已知飞行原来的飞行方向是a表示的方向，需要把飞机飞行方向旋转到向量b表示的方向。



```

const model = new THREE.Group();
loader.load("../飞机.glb", function (gltf) {
    const fly = gltf.scene
    model.add(fly);
    fly.position.set(10, 10, 0); // 相对世界坐标系坐标原点偏移
    const axesHelper = new THREE.AxesHelper(10);
    fly.add(axesHelper); // 用一个坐标轴可视化模型的局部坐标系(本地坐标系)

    const a = new THREE.Vector3(0, 0, -1); // 飞机初始姿态飞行方向
    // 飞机姿态绕自身坐标原点旋转到b指向的方向
    const b = new THREE.Vector3(-1, -1, -1).normalize();

})

```

箭头可视化飞机旋转前后的方向

```

// 可视化飞机方向
const a = new THREE.Vector3(0, 0, -1); // 飞机初始姿态飞行方向
const O = fly.position.clone(); // 飞机位置坐标箭头起点
model.add(new THREE.ArrowHelper(a, O, 30, 0xff0000));
// 飞机姿态绕自身坐标原点旋转到b指向的方向
const b = new THREE.Vector3(-1, -1, -1).normalize();
model.add(new THREE.ArrowHelper(b, O, 30, 0x00ff00));

```

.setFromUnitVectors(a, b) 生成四元数旋转飞机

四元数 `Quaternion` 的方法 `.setFromUnitVectors(a, b)` 可以通过两个向量参数a和b，创建一个四元数，表示从向量a表示的方向旋转到向量b表示的方向。(参数a, b是单位向量)

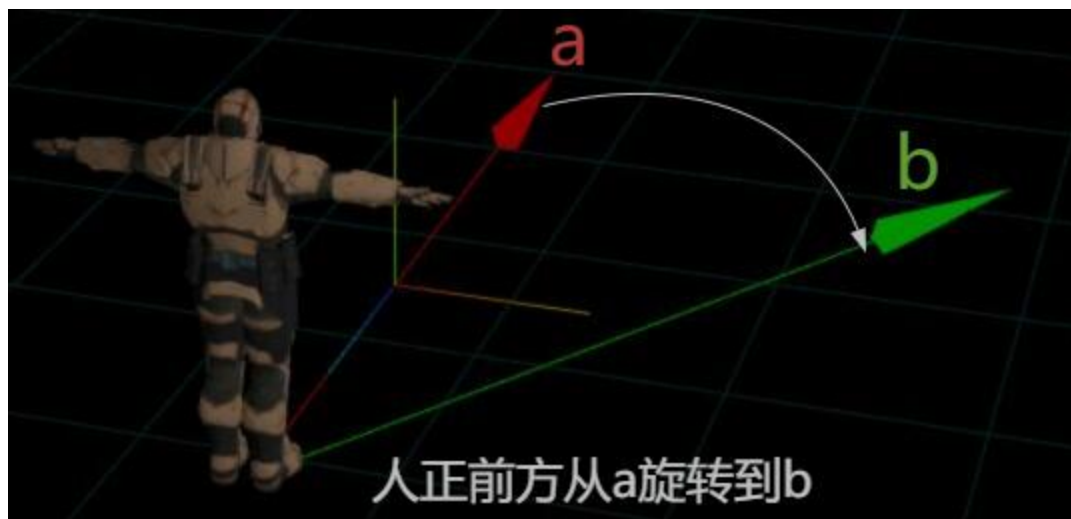
```

// 飞机初始姿态飞行方向a
const a = new THREE.Vector3(0, 0, -1);
// 飞机姿态绕自身坐标原点旋转到b指向的方向
const b = new THREE.Vector3(-1, -1, -1).normalize();
// a旋转到b构成的四元数
const quaternion = new THREE.Quaternion();
// 注意两个参数的顺序
quaternion.setFromUnitVectors(a, b);
// quaternion表示的是变化过程，在原来基础上乘以quaternion即可
fly.quaternion.multiply(quaternion);

```

练习题：人旋转

已知人原来的正前方是a表示的方向，需要把人正前方旋转到向量b表示的方向。



这个旋转过程，可通过 `.setFromUnitVectors(a, b)`

```
//人正前方从向量a表示方向旋转到向量b表示方向
const q = new THREE.Quaternion();
q.setFromUnitVectors(a,b);//a转向b
person.quaternion.multiply(q);
```

← 4. 四元数乘法运算

1. 数学基础(平移、旋转、缩放矩阵) →