

🟡 6. 箭头ArrowHelper

基础课程中给大家介绍过一个three.js辅助开发的类，就是xyz三维坐标系 `AxesHelper`，这节课给大家介绍一个新的辅助类，three.js的箭头 `ArrowHelper`。

介绍 `ArrowHelper` 的目的是，后面几何计算的过程或结果，可以用 `ArrowHelper` 可视化表示出来，这样更容易理解抽象的概念或抽象的数学计算。

两个小球可视化A、B两点位置

```
const A = new THREE.Vector3(0, 30, 0); // A点
const B = new THREE.Vector3(80, 0, 0); // B点

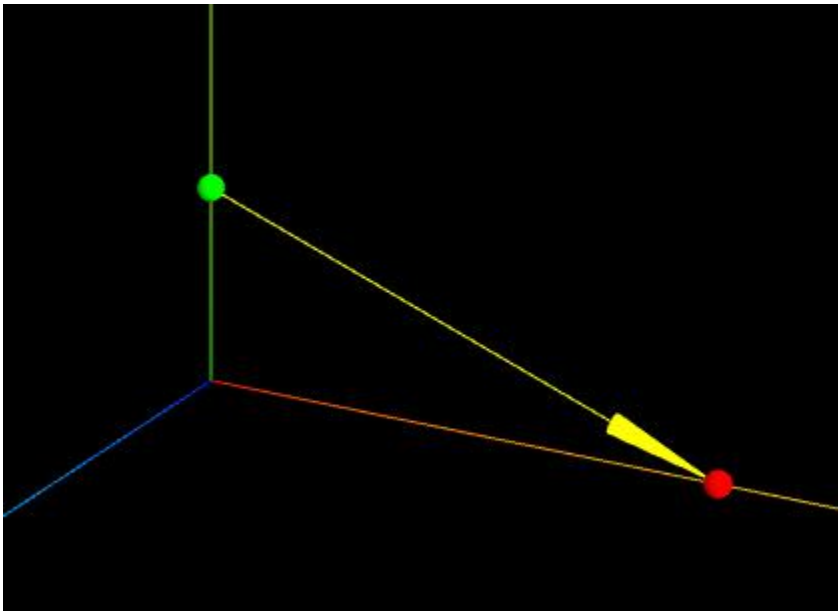
// 绿色小球可视化A点位置
const AMesh = createSphereMesh(0x00ff00, 2);
AMesh.position.copy(A);
// 红色小球可视化B点位置
const BMesh = createSphereMesh(0xff0000, 2);
BMesh.position.copy(B);

const group = new THREE.Group();
group.add(AMesh, BMesh);

function createSphereMesh(color, R) {
  const geometry = new THREE.SphereGeometry(R);
  const material = new THREE.MeshLambertMaterial({
    color: color,
  });
  const mesh = new THREE.Mesh(geometry, material);
  return mesh;
}
```

js

生成一个A指向B的箭头

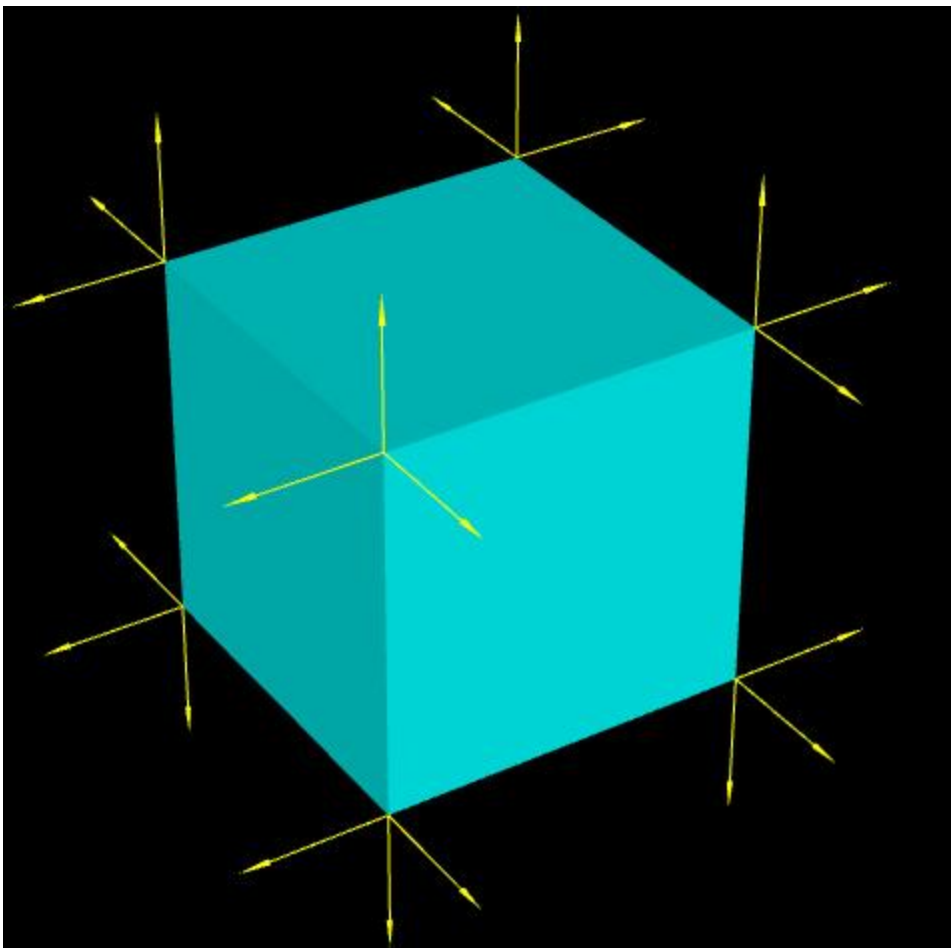


```
// 绘制一个从A指向B的箭头
const AB = B.clone().sub(A);
const L = AB.length(); // AB长度
const dir = AB.clone().normalize(); // 单位向量表示AB方向

// 生成箭头从A指向B
const arrowHelper = new THREE.ArrowHelper(dir, A, L)
group.add(arrowHelper);
```

js

箭头可视化一个立方体的法线方向



```
const geometry = new THREE.BoxGeometry(50, 50, 50);
const material = new THREE.MeshLambertMaterial({
  color: 0x00ffff,
});
const mesh = new THREE.Mesh(geometry, material);

const p = mesh.geometry.attributes.position;
const n = mesh.geometry.attributes.normal;
const count = p.count; // 顶点数量
for (let i = 0; i < count; i++) {
  // 顶点位置O
  const O = new THREE.Vector3(p.getX(i), p.getY(i), p.getZ(i));
  // 顶点位置O对应的顶点法线
  const dir = new THREE.Vector3(n.getX(i), n.getY(i), n.getZ(i));
  // 箭头可视化顶点法线
  const arrowHelper = new THREE.ArrowHelper(dir, O, 20);
  mesh.add(arrowHelper);
}
```

js

总结

一般做一些数学计算的时候，可以借助小球、箭头等方式可视化你的几何数据，通过具象的图像，有助于理解抽象的代码。

← 5. 相机沿着视线方向运动

1. 匀速动画(向量表示速度)→