

## 🔵 5. 相机沿着视线方向运动

利用上节课讲解的向量方向知识，写一个练习案例，就是three.js相机沿着自身视线方向运动。

### 单位向量表示相机视线方向

相机目标观察点，也就是lookAt参数，和相机位置相减,获得一个沿着相机视线方向的向量，然后归一化，就可以获取一个表示相机视线方向的单位向量。

```
camera.position.set(202, 123, 125);
camera.lookAt(0, 0, 0);
// 相机目标观察点和相机位置相减,获得一个沿着相机视线方向的向量
const dir = new THREE.Vector3(0 - 202, 0 - 123, 0 - 125);
// 归一化,获取一个表示相机视线方向的单位向量。
dir.normalize();
```

js

浏览器控制台查看计算结果

```
console.log('相机方向',dir);
console.log('单位向量',dir.length());
```

js

### camera.getWorldDirection() 获取相机视线方向

你通过相机对象的 `.getWorldDirection()` 方法，可以快速获取一个沿着相机视线方向的单位向量，不需要自己写代码计算视线方向了，`.getWorldDirection()` 方法进行了相关的封装。

```
const dir = new THREE.Vector3();
// 获取相机的视线方向
camera.getWorldDirection(dir);
console.log('相机方向',dir);
console.log('单位向量',dir.length());
```

js

## 相机沿着视线方向平移

```
// dis向量表示相机沿着相机视线方向平移200的位移量
const dis = dir.clone().multiplyScalar(200);
// 相机沿着视线方向平移
camera.position.add(dis);
```

js

## 相机沿着视线移动动画(tweenjs库辅助)

下面借助tweenjs动画库，写一个相机沿着相机视线方向移动的动画。前面基础内容中讲解过tweenjs的使用，学习下面内容之前，确保你已经熟悉tweenjs。

```
import TWEEN from '@tweenjs/tween.js';
```

js

相机沿着视线方向移动的动画。

```
const dir = new THREE.Vector3();
camera.getWorldDirection(dir); // 获取相机的视线方向
// dis表示相机沿着相机视线方向平移200
const dis = dir.clone().multiplyScalar(200);
// 相机动画: 平移前坐标—>平移后坐标
new TWEEN.Tween(camera.position)
  .to(camera.position.clone().add(dis), 3000)
  .start();
```

js

```
function render() {
  TWEEN.update();
  renderer.render(scene, camera);
  requestAnimationFrame(render);
}
render();
```

js

## GUI沿着相机视线方向拖动相机平移

```
// 从threejs扩展库引入gui.js
import { GUI } from 'three/addons/libs/lil-gui.module.min.js';
```

js

```
const pos0 = camera.position.clone();//记录相机初始位置
const gui = new GUI();
// L: 相机沿着视线移动距离, 从0~200
gui.add({L: 0}, 'L', 0, 200).onChange(function(v){
    const dis = dir.clone().multiplyScalar(v);//相机沿着视线偏移长度v
    const newPos = pos0.clone().add(dis);//相机初始位置+相机偏移向量
    camera.position.copy(newPos);//新的位置赋值给相机位置
});
```

js

---

← 4. 向量方向(归一化.normalize)

6. 箭头ArrowHelper→