

## 🔵 2. tweenjs相机运动动画

下面给大家讲解如何通过tweenjs实现threejs相机动画，具体说就是使用tweenjs改变相机的位置 `camera.position` 和视线方向。

### 相机飞行动画(从一个点飞到另一个点)

引入tweenjs，并在 `requestAnimationFrame` 动画中执行 `TWEEN.update()`；更新。

```
import TWEEN from '@tweenjs/tween.js';
// 渲染循环
function render() {
  TWEEN.update();
  renderer.render(scene, camera);
  requestAnimationFrame(render);
}
render();
```

相机动画:从一个点移动到另一个点

```
camera.position.set(202, 123, 125);
new TWEEN.Tween(camera.position)
  .to({x: 202,y: 123,z: 50}, 3000)
  .start();
```

### 相机飞行过程中重新计算相机视线

只改变相机位置，相机默认视线方向保持不变，如果你想重新计算相机视线方向，可以在相机位置改变的过程中不停地执行 `lookAt()` 即可。

```
camera.position.set(202, 123, 125);
camera.lookAt(0, 0, 0);
new TWEEN.Tween(camera.position)
  .to({x: 202,y: 123,z: -350}, 3000)
```

```
// tweenjs改变参数对象的过程中，.onUpdate方法会被重复调用执行
.onUpdate(function(){
    camera.lookAt(0, 0, 0);
})
.start()
```

## Tweenjs回调函数

tweenjs库提供了 `onStart`、`onUpdate`、`onComplete` 等用于控制动画执行的回调函数。

- `onStart`：动画开始执行触发
- `onUpdate`：动画执行过程中，一直被调用执行
- `onComplete`：动画正常执行完触发

`.onUpdate(function(obj){})` 结构中，obj对应的是 `new TWEEN.Tween(pos)` 的参数对象 pos。

```
const tween = new TWEEN.Tween(pos).to({x: 0}, 4000)
// 开始执行：动画片段tween开始执行的时候触发onStart
.onStart(function(obj){
    ...
})
```

## 相机圆周运动，且保持相机镜头对准坐标原点

```
const R = 100; //相机圆周运动的半径
new TWEEN.Tween({angle:0})
.to({angle: Math.PI*2}, 16000)
.onUpdate(function(obj){
    camera.position.x = R * Math.cos(obj.angle);
    camera.position.z = R * Math.sin(obj.angle);
    camera.lookAt(0, 0, 0);
})
.start()
```

