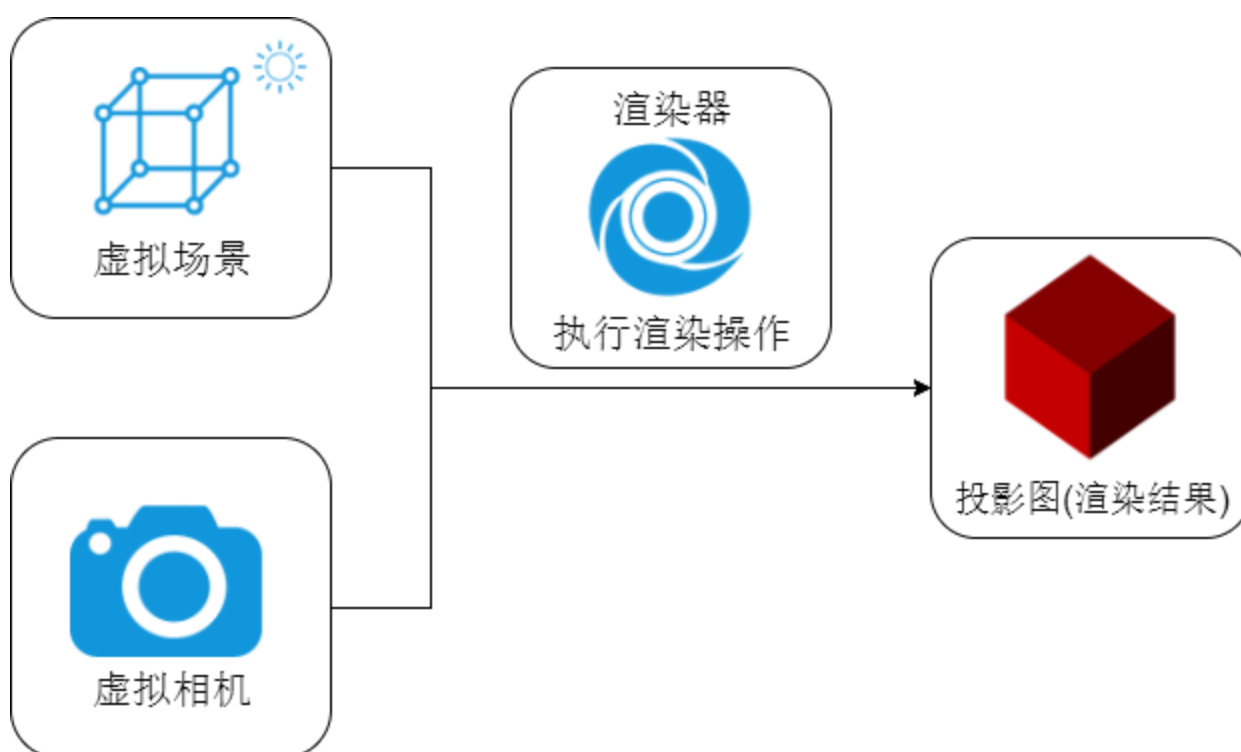


🔗 6. 第一个3D案例—渲染器

视频讲解🔗

生活中如果有了**景物**和**相机**，那么如果想获得一张照片，就需要你拿着相机，按一下，咔，完成拍照。对于threejs而言，如果完成“咔”这个拍照动作，就需要一个新的对象，也就是WebGL渲染器**WebGLRenderer**🔗。



WebGL渲染器 **WebGLRenderer**

通过WebGL渲染器**WebGLRenderer**🔗 可以实例化一个WebGL渲染器对象。

```
// 创建渲染器对象
const renderer = new THREE.WebGLRenderer();
```

设置Canvas画布尺寸 **.setSize()**

```
// 定义threejs输出画布的尺寸(单位:像素px)
const width = 800; //宽度
const height = 500; //高度
renderer.setSize(width, height); //设置three.js渲染区域的尺寸(像素px)
```

渲染器渲染方法 `.render()`

渲染器 `WebGLRenderer` 执行渲染方法 `.render()` 就可以生成一个Canvas画布(照片), 并把三维场景Scene呈现在canvas画布上面,你可以把 `.render()` 理解为相机的拍照动作“咔”。

```
renderer.render(scene, camera); //执行渲染操作
```

渲染器Canvas画布属性 `.domElement`

渲染器 `WebGLRenderer` 通过属性 `.domElement` 可以获得渲染方法 `.render()` 生成的Canvas画布, `.domElement` 本质上就是一个HTML元素: Canvas画布。

```
document.body.appendChild(renderer.domElement);
```

Canvas画布插入到任意HTML元素中

```
<div id="webgl" style="margin-top: 200px;margin-left: 100px;"></div>
```

html

```
document.getElementById('webgl').appendChild(renderer.domElement);
```

