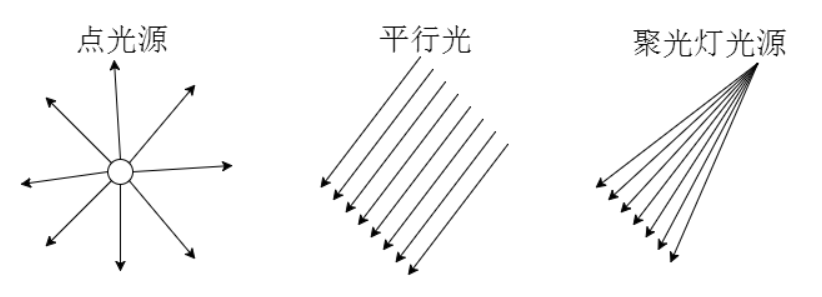
**光源**

如下three提供的光源



1. 环境光AmbientLight

环境光只需设置颜色即可

//环境光:环境光颜色RGB成分分别和物体材质颜色RGB成分分别相乘

var ambient = new THREE.AmbientLight(0x444444);

scene.add(ambient);//环境光对象添加到scene场景中

1. 点光源PointLight

点光源需要设置光源位置

//点光源

var point = new THREE.PointLight(0xffffff);

//设置点光源位置，改变光源的位置

point.position.set(400, 200, 300);

scene.add(point);

1. 平行光DirectionalLight

只要我们指定一个原点和一个目标点，那就可以确定光的方向，所以平行光需要提供一个原点和一个目标点

// 平行光

var directionalLight = new THREE.DirectionalLight(0xffffff, 1);

// 光的原点

directionalLight.position.set(80, 100, 50);

// 光的目标模型

directionalLight.target = mesh2;

scene.add(directionalLight);

1. 聚光源SpotLight

聚光源我们需要知道光的位置，光照射的目标，光照射的弧度

// 聚光光源

var spotLight = new THREE.SpotLight(0xffffff);

// 光源位置

spotLight.position.set(200, 200, 200);

// 光源目标

spotLight.target = mesh2;

// 发散弧度

spotLight.angle = Math.PI / 6;

scene.add(spotLight);//光对象添加到scene场景中

**阴影投影**

要想拥有投影效果，需要经过如下步骤

1. 渲染器开启投影

renderer.shadowMap.enabled = true;

1. 网格模型启用投影

    // 设置产生投影的网格模型

    mesh.castShadow = true;

1. 网格模型2（如玩游戏时的大地模型就可以接收投影）启用接收投影

    //创建一个平面几何体作为投影面

    var planeGeometry = new THREE.PlaneGeometry(300, 200);

    //阴影材质，该材质接收显示阴影，但不显示其本身

    var planeMaterial = new THREE.ShadowMaterial({

        color: 0x999999

    });

    // 平面网格模型作为投影面

    var planeMesh = new THREE.Mesh(planeGeometry, planeMaterial);

    planeMesh.rotateX(-Math.PI / 2);

    planeMesh.position.y = -50;

    // 设置接收阴影的投影面

    planeMesh.receiveShadow = true;

    scene.add(planeMesh); //网格模型添加到场景中

1. 光源启用投影并调整投影参数

    // 方向光

    var directionalLight = new THREE.DirectionalLight(0xffffff, 1);

    // 设置光源位置

    directionalLight.position.set(60, 100, 40);

    // 设置用于计算阴影的光源对象

    directionalLight.castShadow = true;

    // 设置计算阴影的区域，最好刚好紧密包围在对象周围

    // 计算阴影的区域过大：模糊  过小：看不到或显示不完整

    directionalLight.shadow.camera.near = 0.5;

    directionalLight.shadow.camera.far = 300;

    directionalLight.shadow.camera.left = -50;

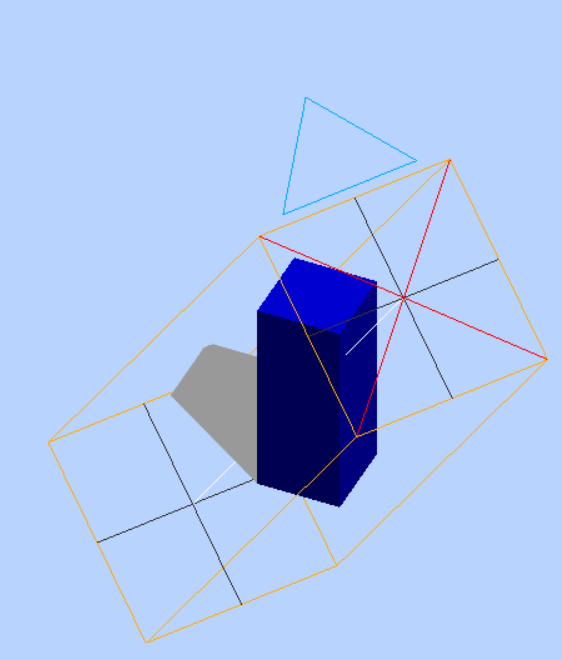
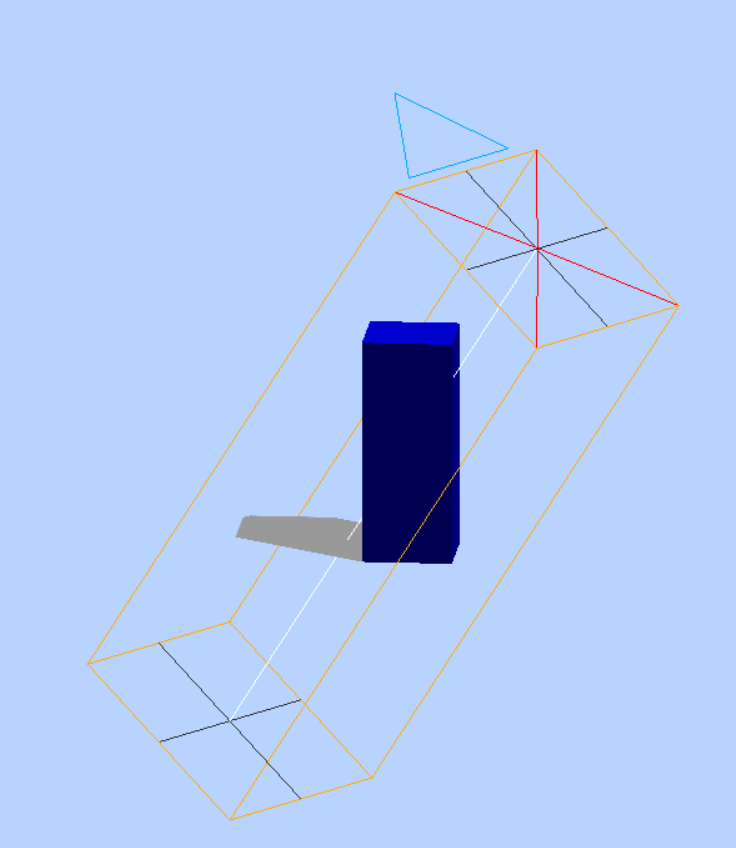
    directionalLight.shadow.camera.right = 50;

    directionalLight.shadow.camera.top = 50;

    directionalLight.shadow.camera.bottom = -100;

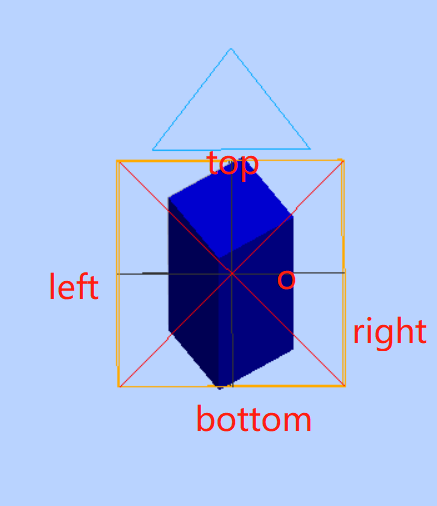
    scene.add(directionalLight);

- camere属性说明，看下图



灯光渲染阴影实际上是在一个正方体内渲染，不在这个正方体内的阴影是不进行渲染的

left、top、right、bottom分别是以o为原点的4条边的偏移，而far则是这个正方体的长



1. 最终效果

