

🎨 6. 工厂光源(平行光模拟太阳光)

继续上节课内容讲解。

- 环境贴图(上节课)
- 环境光(上节课)
- 平行光模拟太阳光

平行光模拟太阳光

比如实际生活中一个三维场景中，可能会有太阳光照射，太阳距离地球足够远，可以近似认为太阳光是平行光。

工厂地面坐标为0的话，光源从斜上方照射到物体上，可以y给予一个大于0的高度值，xz设置不同的值，控制不同的光线照射方向。

在设置光源高度的时候，你可以先大概测量下三场场景的尺寸，布局具体值，知道大概数量级即可，比如几十、几百，工厂长宽方向100量级，高度方向10量级。

```
const directionalLight = new THREE.DirectionalLight(0xffffff, 1);
directionalLight.position.set(100, 60, 50);
scene.add(directionalLight);
```

js

平行光辅助对象 DirectionalLightHelper

通过平行光 `DirectionalLightHelper` 可以可视化平行光 `.position` 和照射方向。

```
// 参数2表示平行光.position附近方框的尺寸
const dirHelper = new THREE.DirectionalLightHelper( directionalLight, 5);
scene.add( dirHelper );
```

js

平行光位置和方向GUI可视化控制

平行光y坐标相对工厂地面设置一定高度，比如通过GUI控制在0~300范围。

```
gui.add(directionalLight.position, 'y', 0, 300);
```

js

平行光位置在XOZ平面上绕着y轴旋转，可以给一个半径 `R`，一个角度值 `angle`，然后改变平行光在XOZ平面上旋转角度，通过三角函数计算x和z坐标。

```
const obj = {
  R: 100,
  angle: 0,
};
gui.add(obj, 'R', 0, 300).onChange(function(value){
  directionalLight.position.x = value * Math.cos(obj.angle);
  directionalLight.position.z = value * Math.sin(obj.angle);
});
gui.add(obj, 'angle', 0, Math.PI*2).onChange(function(value){
  directionalLight.position.x = obj.R * Math.cos(value);
  directionalLight.position.z = obj.R * Math.sin(value);
});
```

js

更新平行光辅助对象 `DirectionalLightHelper`

当平行光位置变化的时候，可以执行 `helper.update()` 更新行光辅助对象

`DirectionalLightHelper` 的姿态同步变化。

```
gui.add(directionalLight.position, 'y', 0, 300).onChange(function(value){
  dirHelper.update();
});
```

js

```
const obj = {
  R: 100,
  angle: 0,
};
gui.add(obj, 'R', 0, 300).onChange(function(value){
  directionalLight.position.x = value * Math.cos(obj.angle);
  directionalLight.position.z = value * Math.sin(obj.angle);
  dirHelper.update();
});
gui.add(obj, 'angle', 0, Math.PI*2).onChange(function(value){
  directionalLight.position.x = obj.R * Math.cos(value);
```

js

```
directionalLight.position.z = obj.R * Math.sin(value);  
dirHelper.update();  
});
```

← [5. 工厂光源\(环境贴图和环境光\)](#)

[7. 工厂\(模拟太阳光阴影\)](#) →