- Change `SpotLight` parameters: its position and FOV angle to smaller. Setting `shadowCameraVisible` to *true*, enables to display lines showing the shadowing area. Please get the image of the light circle on the plane and the objects. This should be done for both materials, i.e.: `MeshLambertMaterial` and `MeshPhongMaterial`. It's worth looking that they behave completely different. For `MeshLambertMaterial` we need a fine mesh (light is interpolated between vertices). `MeshPhongMaterial` works directly on pixels and does not need any mesh. Please play with changing these parameters and compare the results with both materials. Choose the one, which you like.

```
// add spotlight for the shadows
var spotLight = new THREE.SpotLight(0xffffff);
spotLight.position.set(0, 3, 15);
spotLight.castShadow = true;
spotLight.shadowCameraVisible = true;

spotLight.angle = 60;
spotLight.penumbra = 1;
spotLight.target = sphere;
var helper = new THREE.CameraHelper( spotLight.shadow.camera );
scene.add( helper );
spotLight.shadow.camera.near = 0.12;
spotLight.shadow.camera.far = 80;
spotLight.shadow.camera.fov = 10;
spotLight.shadow.mapSize.width = 2048;
spotLight.shadow.mapSize.height = 2048;


scene.add(spotLight);
```

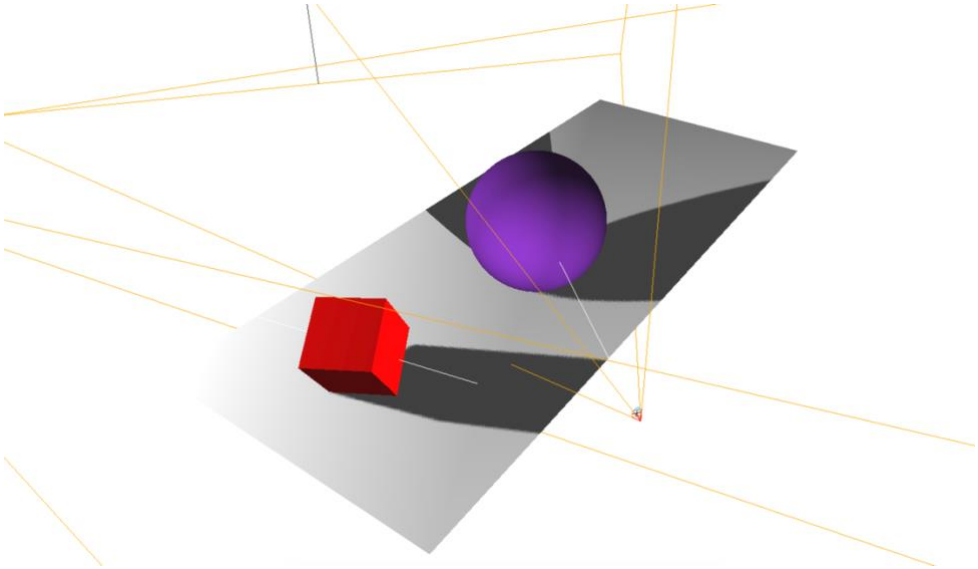
- Please add another light source, directional or spotlight, but with bigger FOV angle.

```
// add spotlight2
var spotLight2 = new THREE.SpotLight(0xffffff);
spotLight2.position.set(0, 10, 30);
spotLight2.castShadow = true;
spotLight2.shadowCameraVisible = true;

spotLight2.angle = 20;
spotLight2.penumbra = 1;
spotLight2.target = cube;
var helper2 = new THREE.CameraHelper( spotLight2.shadow.camera );
scene.add( helper2 );
spotLight2.shadow.camera.near = 0.12;
spotLight2.shadow.camera.far = 80;
spotLight2.shadow.camera.fov = 234;
spotLight2.shadow.mapSize.width = 2048;
spotLight2.shadow.mapSize.height = 2048;


scene.add(spotLight2);
```

- Check which parameters control shadow casting and shadow receiving. Can you cast a cube shadow on the sphere?



- Please change the material from `MeshLambertMaterial` to `MeshPhongMaterial` in order to get specular lights on the cube and sphere. ncrease `shadow.map.width` and `shadow.map.height` parameters to get better results. You may also play with `shadow.camera.near`, `shadow.camera.far` and `shadow.camera.fov` . Make the shadows look nice and natural.

- Please add a transparent cone to a spotlight. This cone should mimic light which we usually see in the fog, when light is diffused on the droplets of water. Partial transparency of such a cone can be achieved using `transparent` and `opacity` parameters, when defining materials e.g.

  ```
  new THREE.MeshLambertMaterial( { opacity:0.6, color: 0x44ff44, transparent:true } );
  ```

  Try to make the cone move along with the light source.

```
//Cone
var coneGeometry = new THREE.ConeGeometry(2.3, 6, 0);
var coneMaterial =  new THREE.MeshLambertMaterial( { opacity:0.6, color: 0x25dc0c, transparent:true } );
//var coneMaterial = new THREE.MeshLambertMaterial({color: 0x25dc0c});
var cone = new THREE.Mesh(coneGeometry, coneMaterial);

// position the cone
cone.position.x = -13;
cone.position.y = 3;
cone.position.z = 2;
cone.castShadow = true;

// add the cone to the scene
scene.add(cone);
```

- Please add more moving objects and more moving, preferabely color, light sources.

```
//add hemisphere light
var pointColor = "#ffffff";
//var spotLight = new THREE.SpotLight( pointColor);
var spotLight3 = new THREE.DirectionalLight(pointColor);
spotLight3.position.set(30, 10, -50);
spotLight3.castShadow = true;
spotLight3.shadowCameraNear = 0.1;
spotLight3.shadowCameraFar = 100;
spotLight3.shadowCameraFov = 10;
spotLight3.target = plane;
spotLight3.distance = 0;
spotLight3.shadowCameraNear = 2;
spotLight3.shadowCameraFar = 70;
spotLight3.shadowCameraLeft = -60;
spotLight3.shadowCameraRight = 60;
spotLight3.shadowCameraTop = 60;
spotLight3.shadowCameraBottom = -60;
spotLight3.shadowMapWidth = 2048;
spotLight3.shadowMapHeight = 2048;


scene.add(spotLight3);

// bounce the cone up and down
cone.position.x = -23 + ( -5 * (Math.cos(step)));
cone.position.y = 3 + ( 32 * Math.abs(Math.sin(step)));

// add spotlight for a bit of light
var spotLight0 = new THREE.SpotLight(0x003cff);
spotLight0.position.set(-40, 50, -10);
spotLight0.lookAt(plane);
scene.add(spotLight0);
```