

🟡 9. CSS3DRenderer渲染HTML标签

CSS3渲染器 `CSS3DRenderer` 和CSS2渲染器 `CSS2DRenderer` 整体使用流程基本相同，只是在HTML标签渲染效果方面不同，比如CSS3渲染的标签会跟着场景相机同步缩放，而CSS2渲染的标签默认保持自身像素值。

下面就在CSS2渲染器代码基础上给大家讲解。

设置CSS3渲染器代码

和CSS2渲染器代码一样设置，只需要把CSS2换成CSS3即可。

```
// 引入CSS3渲染器CSS3DRenderer
import {CSS3DRenderer} from 'three/addons/renderers/CSS3DRenderer.js';
```

```
// 创建一个CSS3渲染器CSS3DRenderer
const css3Renderer = new CSS3DRenderer();
css3Renderer.setSize(width, height);
// HTML标签<div id="tag"></div>外面父元素叠加到canvas画布上且重合
css3Renderer.domElement.style.position = 'absolute';
css3Renderer.domElement.style.top = '0px';
//设置.pointerEvents=none，解决HTML元素标签对threejs canvas画布鼠标事件的遮挡
css3Renderer.domElement.style.pointerEvents = 'none';
document.body.appendChild(css3Renderer.domElement);
```

```
// 渲染循环
function render() {
  css3Renderer.render(scene, camera);
  // ...
  requestAnimationFrame(render);
}
```

```
window.onresize = function () {
  ...
}
```

```
// HTML标签css3Renderer.domElement尺寸重新设置
css3Renderer.setSize(width,height);

};
```

CSS3对象模型 `CSS3DObject`

CSS3对象模型 `CSS3DObject` 可以类比前面介绍的CSS2模型对象 `CSS2DObject` 学习。

```
// 引入CSS3模型对象CSS3DObject
import { CSS3DObject } from 'three/addons/renderers/CSS3DRenderer.js';
```

js

通过 `CSS3DObject` 类，可以把一个HTML元素转化为一个CSS3模型对象，就像threejs的网格模型一样，可以添加到场景中，可以设置位置，可以作为其它模型对象的子对象。

```
const div = document.getElementById('tag');
// HTML元素转化为threejs的CSS3模型对象
const tag = new CSS3DObject(div);
//标签tag作为mesh子对象，默认标注在模型局部坐标系坐标原点
mesh.add(tag);
// 相对父对象局部坐标原点偏移80,刚好标注在圆锥
tag.position.y += 80;
```

js

`tag.position.y += 80;` 标注圆锥模型的顶部

```
const geometry = new THREE.ConeGeometry(25, 80);
geometry.translate(0, 40, 0);
const mesh = new THREE.Mesh(geometry, material);
mesh.add(tag);
// 相对父对象局部坐标原点偏移80,刚好标注在圆锥顶部
tag.position.y += 80;
```

js

`CSS3DObject` 渲染效果测试

CSS3模型对象 `CSS3DObject` 渲染结果，就像一个矩形平面网格模型一样。你通过相机控件 `OrbitControls` 旋转、缩放三维场景，CSS3模型对象 `CSS3DObject` 跟着旋转、缩放。

旋转过程中HTML元素标签的正反面都可以看到。

一个网格模型被另一个不透明网格模型遮挡，canvas画布上不会显示，不过注意一点 `CSS3DObject` 模型本质上渲染到网页上还是HTML元素，这就是说模型背面的HTML标签并不会被遮挡，`CSS3DObject` 标签是以HTML元素形式叠加在canvas画布上的，不受threejs内部模型对象影响。

禁止 `CSS3DObject` 标签对应HTML元素背面显示

```
<div id="tag" style="backface-visibility: hidden;">标签内容</div>
```

js

标签相对标注点的位置

默认情况下CSS3模型对象渲染的标签的几何中心默认和标注位置的坐标重合。

CSS3DRenderer渲染的HTML标签尺寸

CSS2DRenderer渲染HTML元素标签，默认情况下，HTML元素会保持本身尺寸的像素值，除非你通过代码缩放。

CSS3DRenderer渲染的HTML元素，你可以把HTML标签对象想象为一个矩形平面网格模型 Mesh，HTML标签对象在threejs中的尺寸来源是HTML元素的像素尺寸值，比如HTML像素高度40px，那么HTML标签对象在threejs中的数字相当于高度为40的矩形平面网格模型。

测试验证上面规律总结：把标签的高度设置为160px，160的一半是80，也就是圆锥的高度，这样HTML标签下半部分和圆锥底部重合。

```
<div id="tag" style="height: 160px;">标签内容</div>
```

js

border、padding、height、width都会影响标签渲染大小，你可以分别测试体验。

```
<style>
  #tag {
    padding: 0px 10px;
    border: #00ffff solid 1px;
    height: 40px;
    border-radius: 5px;
    width: 65px;
  }
```

html

```
</style>
```

缩放标签

```
const div = document.getElementById('tag');
const tag = new CSS3DObject(div);
tag.scale.set(0.5,0.5,1);//缩放标签尺寸
```

js

标签偏移

CSS2渲染HTML标签偏移方式

```
const div = document.getElementById('tag');
// id="tag"元素高度322px,默认标签中心与标注点
div.style.top = '-161px'; //平移-161px, 指示线端点和标注点重合
```

CSS2渲染的标签和CSS3渲染的标签偏移方式不同, CSS3标签, 直接按照threejs模型尺寸修改方式改变, 比用HTML像素方式更方便准确。

```
tag.scale.set(0.5,0.5,1);//缩放标签尺寸
tag.position.y += 10;//累加标签高度一半, 标签底部和圆锥顶部标注位置重合
```

js

CSS3精灵模型 CSS3DSprite

```
// 引入CSS3精灵模型对象CSS3DSprite
import { CSS3DSprite } from 'three/addons/renderers/CSS3DRenderer.js';
```

js

CSS3对象模型 `CSS3DObject` 渲染效果类似矩形平面网格模型 `Mesh` 。

CSS3精灵模型 `CSS3DSprite` 渲染效果类似以前学习的精灵模型对象 `Sprite` 。

```
const div = document.getElementById('tag');
// HTML元素转化为threejs的CSS3精灵模型`CSS3DSprite`
const tag = new CSS3DSprite(div);
//标签tag作为mesh子对象, 默认标注在模型局部坐标系坐标原点
mesh.add(tag);
// 相对父对象局部坐标原点偏移80,刚好标注在圆锥
```

js

```
tag.position.y += 80;
```

CSS3精灵模型 `CSS3DSprite` 渲染特点

CSS3精灵模型 `CSS3DSprite` 对应的HTML标签，可以跟着场景缩放，位置可以跟着场景旋转，但是自身的姿态角度始终平行于canvas画布，不受旋转影响，就像精灵模型一样 `Sprite`

CSS3精灵模型 `CSS3DSprite` 尺寸、位置、缩放等渲染规律和CSS3对象模型 `CSS3DObject` 基本一致。

标签局部遮挡鼠标事件

HTML标签 `<div id="tag"></div>` 外面的父元素 `css3Renderer.domElement` 防止鼠标遮挡 canvas事件方式，和CSS2渲染器一样。

```
//设置.pointerEvents=none，解决HTML元素标签对threejs canvas画布鼠标事件的遮挡  
css3Renderer.domElement.style.pointerEvents = 'none';
```

js

标签 `<div id="tag"></div>` 在CSS3渲染器渲染的时候，默认会被设置为 `pointer-events: auto;`，这时候虽然 `css3Renderer.domElement` 不遮挡canvas画布的鼠标事件，但是 `<div id="tag"></div>` 遮挡canvas画布的鼠标事件。

这时候你可以通过代码强制改变CSS3渲染器给标签设置的 `.style.pointerEvents = 'auto'`，设置为 `.style.pointerEvents = 'none'`，这时候注意一点，修改 `.style.pointerEvents`，要在实例化 `new CSS3DObject(div)` 之后，因为执行 `new CSS3DObject(div)` 的时候，会把HTML标签设置为 `.style.pointerEvents = 'auto'`。

```
const div = document.getElementById('tag');  
// HTML元素转化为threejs的CSS3模型对象  
const tag = new CSS3DObject(div);  
// new CSS3DObject(div);之后设置style.pointerEvents  
div.style.pointerEvents = 'none';
```

js

