

🔗 1. CSS2DRenderer(HTML标签)

下面给大家介绍一个threejs的扩展库 `CSS2DRenderer.js`，通过 `CSS2DRenderer.js` 可以把HTML元素作为标签标注三维场景。

引入扩展库 `CSS2DRenderer.js`

在threejs文件包目录 `examples/jsm/renderers/`，你可以找到 `CSS2DRenderer.js` 扩展库。

threejs扩展库CSS2DRenderer.js提供了两个类CSS2渲染器 `CSS2DRenderer`、CSS2模型对象 `CSS2DObject`。

```
// 引入CSS2渲染器CSS2DRenderer和CSS2模型对象CSS2DObject
import { CSS2DRenderer, CSS2DObject } from 'three/addons/renderers/CSS2DRenderer.js'
```

```
// 引入CSS2渲染器CSS2DRenderer
import { CSS2DRenderer } from 'three/addons/renderers/CSS2DRenderer.js';
// 引入CSS2模型对象CSS2DObject
import { CSS2DObject } from 'three/addons/renderers/CSS2DRenderer.js';
```

总体思路

本节课先为大家演示标签标注最基本的内容，更多需要完善优化的细节，可以关注后面几节课讲解。

- 1.HTML元素创建标签
- 2.CSS2模型对象 `CSS2DObject`
- 3.CSS2渲染器 `CSS2DRenderer`
- 4. `CSS2Renderer.domElement` 重新定位

1. HTML元素创建标签

你可以根据需要，使用HTML、CSS代码创建你想要的标签。如果基于vue或react开发，也可以用vue或react创建的UI组件表示标签。

```
<div id="tag">标签内容</div>
```

js

如果你想用HTML元素作为标签标注三维场景中模型信息，就需要考虑定位的问题。比如一个模型，在代码中你可以知道它的局部坐标或世界坐标xyz，但是你并不知道渲染后在canvas画布上位置，距离web页面顶部top和左侧的像素px值。

自己写代码把世界坐标xyz，转化为像素px表示屏幕坐标，比较麻烦，不过threejs扩展库 `CSS2DRenderer.js` 可以帮助你实现坐标转化，给HTML元素标签定位，下面给大家演示如何实现。

2. CSS2模型对象 `CSS2DObject`

```
// 引入CSS2模型对象CSS2DObject
import { CSS2DObject } from 'three/addons/renderers/CSS2DRenderer.js';
```

js

通过 `CSS2DObject` 类，可以把一个HTML元素转化为一个类似threejs网格模型的对象，换句话说就是你可以把 `CSS2DObject` 当成threejs的一个模型一样去设置位置 `.position` 或添加到场景中。

```
const div = document.getElementById('tag');
// HTML元素转化为threejs的CSS2模型对象
const tag = new CSS2DObject(div);
```

js

你想把HTML标签标注在那个位置，你通过 `.position` 属性设置标签模型对象的xyz坐标。

```
tag.position.set(50,0,50);
```

js

把HTML元素对应的CSS2模型对象添加到其它模型对象或三维场景中。

```
scene.add(tag);
```

js

```
const group = new THREE.Group();
group.add(tag);
```

js

3. CSS2渲染器 `CSS2DRenderer`

```
// 引入CSS2渲染器CSS2DRenderer
import { CSS2DRenderer } from 'three/addons/renderers/CSS2DRenderer.js';
```

js

```
// 创建一个CSS2渲染器CSS2DRenderer
const css2Renderer = new CSS2DRenderer();
```

js

CSS2渲染器 `CSS2DRenderer` 和WebGL渲染器 `WebGLRenderer` 虽然不同，但是有些属性和方法是相似的，可以类比记忆学习。比如 `.domElement`、`.setSize()`、`.render()`。可以先参考webgl渲染器写代码，然后再给大家解释其中的含义。

3.1 `CSS2Renderer.render()` 渲染HTML标签

CSS2渲染器 `CSS2DRenderer` 和常用的WebGL渲染器 `WebGLRenderer` 一样都是渲染器，只是渲染模型对象不同，`WebGLRenderer` 主要是渲染threejs自身的网格、点、线等模型，`CSS2DRenderer` 用来渲染HTML元素标签对应的CSS2模型对象 `CSS2DObject`。

```
// 用法和webgl渲染器渲染方法类似
css2Renderer.render(scene, camera);
// renderer.render(scene, camera);
```

js

3.2 `CSS2Renderer.setSize()`

设置 `CSS2Renderer.render()` 渲染输出标签的尺寸范围，一般和threejs canvas画布宽高度一致即可。

```
// width, height: canvas画布宽高度
css2Renderer.setSize(width, height);
```

js

3.3 渲染结果 `CSS2Renderer.domElement`

`CSS2Renderer.render()` 渲染会输出标签对应的HTML元素，也就是 `css2Renderer.domElement`，你可以插入到web网页中任何你想放的位置。

```
document.body.appendChild(css2Renderer.domElement);
```

js

查看 `css2Renderer.render()` 渲染结果 `CSS2Renderer.domElement`

threejs执行 `css2Renderer.render()` 之后，你打开浏览器控制台元素选项找到你创建的HTML标签 `<div id="tag">标签内容</div>`，你可以发现 `<div id="tag"></div>` 外面多了一层div父元素，`CSS2Renderer.domElement` 对应的就是 `<div id="tag"></div>` 外面的父元素。

```
document.body.appendChild(css2Renderer.domElement);  
// 渲染HTML标签对应的CSS2DObject模型对象  
css2Renderer.render(scene, camera);
```

js

外面父元素的宽高度也被 `CSS2Renderer` 设置为threejs canvas画布的宽高度，就是 `css2Renderer.setSize(width, height);` 设置的结果。

```
css2Renderer.setSize(width, height);
```

js

```
<!-- `<div id="tag"></div>` 外面多了一层div父元素 -->  
<div style="overflow: hidden; width: 600px; height: 300px;">  
</div>
```

html

4. `CSS2Renderer.domElement` 重新定位

`<div id="tag"></div>` 外面div父元素重新定位，叠加到canvas画布上，与canvas画布重合即可，你可以可以看到HTML标签的标注效果。

`CSS2Renderer.domElement` 定位方法很多，不过这都是普通前端CSS知识，也不一定要与课程相同，你可以根据你自己的前端CSS知识，自由发挥。你只要能让标签父元素叠加到threejs canvas画布上面且重合就行。

下面是本节课案例中布局方式写法，后面也会给大家演示其它的布局写法。

```
css2Renderer.domElement.style.position = 'absolute';  
css2Renderer.domElement.style.top = '0px';
```

js

你可以测试下面两个div元素的布局规律，会知道为什么设置 `.style.top = '0px'` 。

```
<div style="height: 300px;background: #999;">默认定位元素</div>
<div style="position: absolute;">绝对定位</div>
```

js

改变canvas画布在网页的布局位置，标签父元素 `css2Renderer.domElement` 也要跟着重新定位

```
// 改变canvas画布在网页位置，标签父元素也要重新定位
renderer.domElement.style.marginTop = '200px';
css2Renderer.domElement.style.top = '200px';
```

js

css2Renderer.render() 渲染本质

`<div id="tag"></div>` 本身也多了一些CSS位置相关属性，这些都是 `css2Renderer.render()` 渲染的结果。你也可以发现，你创建的HTML标签 `<div id="tag"></div>` 不在原来的位置了，其实是被 `CSS2Renderer` 改变了位置。

`css2Renderer.render()` 渲染HTML元素对应的CSS2模型对象，本质上就是根据CSS2模型对象的xyz世界坐标，计算HTML标签元素在canvas画布上的屏幕像素坐标位置。

```
const div = document.getElementById('tag');
// HTML元素转化为threejs的CSS2模型对象
const tag = new CSS2DObject(div);
```

js

```
// 渲染HTML标签对应的CSS2DObject模型对象
css2Renderer.render(scene, camera);
```

js

测试

改变模型mesh位置。

```
mesh.position.set(100,0,0);
```

js

如果你想用HTML元素标注模型，可以把CSS2模型对象也设置在mesh对应的位置。

```
tag.position.set(100,0,0);
```

js

← [7. 射线拾取Sprite控制场景](#)

[2. HTML标签遮挡Canvas画布事件](#) →