

🎯 2. 正投影相机-Canvas尺寸变化

前面给大家讲解过(1.12、8.1两小节)，一些应用场景Canvas画布可能不是固定尺寸，而是变化的。这时候就需要重新设置WebGL渲染器、相机等参数。

前面给大家演示的时候，相机是用的透视投影，下面主要演示下，正投影相机的参数如何随着Canvas画布尺寸的变化而变化。

- [1.12.Canvas画布尺寸变化](#)🔗
- [8.1.three.js Canvas画布布局](#)🔗

WebGL渲染器更新Canvas画布尺寸

```
// onresize 事件会在窗口被调整大小时发生
window.onresize = function () {
  const width = window.innerWidth; //canvas画布宽度
  const height = window.innerHeight; //canvas画布高度
  // 重置渲染器输出画布canvas尺寸
  renderer.setSize(width, height);
};
```

js

```
window.onresize = function () {
  const width = window.innerWidth-200; //canvas画布宽度
  const height = window.innerHeight-200; //canvas画布高度
  // 重置渲染器输出画布canvas尺寸
  renderer.setSize(window.innerWidth, window.innerHeight);
};
```

js

相机参数更新

`.aspect` 属性受到canvas画布宽高度影响，当canvas画布尺寸发生变化时，需要更新透视投影相机 `PerspectiveCamera` 的 `.aspect` 属性。

js

```
window.onresize = function () {  
    // width、height表示canvas画布宽高度  
    camera.aspect = width / height;  
    camera.updateProjectionMatrix();  
};
```

你预览案例源码中正投影相机代码，可以看到，正投影相机 `OrthographicCamera` 的 `left`、`right` 属性受到canvas画布宽高比影响，所以需要随着canvas画布更新。

js

```
const k = width / height; //canvas画布宽高比  
const s = 50;  
//控制left, right, top, bottom范围大小  
const camera = new THREE.OrthographicCamera(-s * k, s * k, s, -s, 1, 8000);
```

js

```
// Canvas画布跟随窗口变化  
window.onresize = function () {  
    const width = window.innerWidth; //canvas画布宽度  
    const height = window.innerHeight; //canvas画布高度  
    // 1. WebGL渲染器渲染的Canvas画布尺寸更新  
    renderer.setSize(width, height);  
    // 2.1.更新相机参数  
    const k = width / height; //canvas画布宽高比  
    camera.left = -s*k;  
    camera.right = s*k;  
    // 2.2.相机的left, right, top, bottom属性变化了，通知threejs系统  
    camera.updateProjectionMatrix();  
};
```

← 1. 正投影相机

3. 包围盒Box3 →