郭隆邦 2023-01-20

12. Canvas画布布局和全屏

视频讲解

threejs渲染输出的结果就是一个Cavnas画布，canvas画布也是HTML的元素之一，这意味着three.js渲染结果的布局和普通web前端习惯是一样的。

通过 `renderer.domElement` 属性可以访问threejs的渲染结果，也就是HTML的元素 `canvas` 画布。

非全屏局部布局

你可以把threejs的渲染结果 `renderer.domElement` ，插入到web页面上任何一个元素中，只要符合你项目的布局规则即可。

```html
<div id="webgl" style="margin-top: 100px;margin-left: 200px;"></div>
<script type="module">
// width和height用来设置Three.js输出的Canvas画布尺寸(像素px)
const width = 800; //宽度
const height = 500; //高度

const camera = new THREE.PerspectiveCamera(30, width / height, 1, 3000);

/**
 * 创建渲染器对象
 */
const renderer = new THREE.WebGLRenderer();
renderer.setSize(width, height); //设置three.js渲染区域的尺寸(像素px)
renderer.render(scene, camera); //执行渲染操作
//three.js执行渲染命令会输出一个canvas画布，也就是一个HTML元素，你可以插入到web页面中
// document.body.appendChild(renderer.domElement);
document.getElementById('wegbl').appendChild(renderer.domElement);
```

全屏渲染

```js
// width和height用来设置Three.js输出的Canvas画布尺寸(像素px)
const width = window.innerWidth; //窗口文档显示区的宽度作为画布宽度
const height = window.innerHeight; //窗口文档显示区的高度作为画布高度
const renderer = new THREE.WebGLRenderer();
document.body.appendChild(renderer.domElement);
```

全屏布局注意CSS的设置。

```html
<style>
    body{
        overflow: hidden;
        margin: 0px;
    }
</style>
```

canvas画布宽高度动态变化

canvas画布宽高度动态变化,需要更新相机和渲染的参数，否则无法正常渲染。

```js
// onresize 事件会在窗口被调整大小时发生
window.onresize = function () {
    // 重置渲染器输出画布canvas尺寸
    renderer.setSize(window.innerWidth, window.innerHeight);
    // 全屏情况下：设置观察范围长宽比aspect为窗口宽高比
    camera.aspect = window.innerWidth / window.innerHeight;
    // 渲染器执行render方法的时候会读取相机对象的投影矩阵属性projectionMatrix
    // 但是不会每渲染一帧，就通过相机的属性计算投影矩阵(节约计算资源)
    // 如果相机的一些属性发生了变化，需要执行updateProjectionMatrix ()方法更新相机的投
    camera.updateProjectionMatrix();
};
```

← 11. 动画渲染循环

13. stats查看threejs渲染帧率→