郭隆邦 2023-01-29

5. Three.js渲染结果保存为图片

保存three.js渲染结果，其实就是保存three.js对应canvas画布上的图像。那么这个问题就转化为如何把canvas画布保存为一个图片。

超链接元素a下载文件

在学习下面内容之前，如果你有兴趣，可以选择补充下前端相关知识，具体说就是通过超链接元素a合成一个文件，并下载。

你通过下面代码，可以通过点击按钮“下载”，创建一个txt文件，下载到本地，txt文件包含字符串“一些数据”。

```
<button type="button" name="button" onclick="saveFile()">下载</button>
<script>
  function saveFile() {
    // 创建一个超链接元素，用来下载保存数据的文件
    const link = document.createElement('a');
    // 通过超链接herf属性，设置要保存到文件中的数据
    link.href = window.URL.createObjectURL(new Blob(['一些数据']));
    link.download = '文件名称.txt';//下载文件名
    link.click();//js代码触发超链接元素a的鼠标点击事件，开始下载文件到本地
  }
</script>
```

1. 配置webgl渲染器 preserveDrawingBuffer:true

```
// WebGL渲染器设置
const renderer = new THREE.WebGLRenderer({
    //想把canvas画布上内容下载到本地，需要设置为true
    preserveDrawingBuffer:true,
});
```

2. 按钮绑定鼠标事件

创建一个UI按钮"下载"，绑定一个鼠标单击事件，用于后面点击下载图片。

```js
// 鼠标单击id为download的HTML元素，threejs渲染结果以图片形式下载到本地
document.getElementById('download').addEventListener('click',function(){

})
```

3. 创建超链接元素a：用于保存下载文件

```js
// 鼠标单击id为download的HTML元素，threejs渲染结果以图片形式下载到本地
document.getElementById('download').addEventListener('click',function(){
    // 创建一个超链接元素，用来下载保存数据的文件
    const link = document.createElement('a');
    // 通过超链接herf属性，设置要保存到文件中的数据
    link.href = ;
    link.download = 'threejs.png'; //下载文件名
    link.click(); //js代码触发超链接元素a的鼠标点击事件，开始下载文件到本地
})
```

4. Cavnas方法 `.toDataURL()`

Canvas画布通过 `.toDataURL()` 方法可以获取画布上的像素信息。

`canvas.toDataURL("image/png");` 表示以png格式获取像素数据，可以直接赋值给超链接元素a的 `.herf` 属性下载到本地。

```js
const link = document.createElement('a');
// 通过超链接herf属性，设置要保存到文件中的数据
const canvas = renderer.domElement; //获取canvas对象
link.href = canvas.toDataURL("image/png");
```

以不同的格式获取像素信息

```js
canvas.toDataURL("image/png");
canvas.toDataURL("image/jpeg");
canvas.toDataURL("image/bmp");
```

← 4. Three.js背景透明度

6. 深度冲突(模型闪烁)→