郭隆邦 2023-02-25

3. Sprite模拟下雨、下雪

在实际开发的时候，物联网3D可视化、数字孪生、游戏等项目可能会模拟天气的效果。

如果你想模拟下雨效果，一个雨滴用一个3D水滴形曲面表示，假设一个水滴用40个三角形表示，1万个雨滴，就是40万个三角形，精灵模型Sprite在threejs内部就像相当于两个三角形构成的矩形，1万个精灵模型，相当于2万个三角形，Sprite模拟雨滴相比比3D曲面几何体表示雨滴顶点数量就会少很多，这样threejs渲染性能就更好。

`Sprite` 模拟雨滴

提供一个背景透明的png雨滴贴图，然后作为Sprite的颜色贴图，用来模拟雨滴3D几何体。

```js
const texture = new THREE.TextureLoader().load("./雨滴.png");
const spriteMaterial = new THREE.SpriteMaterial({
    map: texture,
});
const sprite = new THREE.Sprite(spriteMaterial);
```

雨滴在3D空间随机分布

批量创建多个精灵模型，在一个长方体空间上随机分布。

Sprite分布渲染范围和数量，根据渲染范围来预先给一个大概的值，然后可以根据需要，在调整雨滴分布范围尺寸。

```js
const group = new THREE.Group();
for (let i = 0; i < 16000; i++) {
    // 精灵模型共享材质
    const sprite = new THREE.Sprite(spriteMaterial);
    group.add(sprite);
    sprite.scale.set(1, 1, 1);
    // 设置精灵模型位置，在长方体空间上上随机分布
    const x = 1000 * (Math.random() - 0.5);
    const y = 600 * Math.random();
```

```
    const z = 1000 * (Math.random() - 0.5);
    sprite.position.set(x, y, z)
}
```

周期性改变雨滴Sprite位置

```js
function loop() {
    // loop()每次执行都会更新雨滴的位置，进而产生动画效果
    group.children.forEach(sprite => {
        // 雨滴的y坐标每次减1
        sprite.position.y -= 1;
        if (sprite.position.y < 0) {
            // 如果雨滴落到地面，重置y，从新下落
            sprite.position.y = 600;
        }
    });
    requestAnimationFrame(loop);
}
loop();
```

`loop()`执行时间间隔和渲染循环`render()`是一样的。

根据时间计算Sprite位置

```js
const clock = new THREE.Clock();
function loop() {
    // loop()两次执行时间间隔
    const t = clock.getDelta();
    group.children.forEach(sprite => {
        // 雨滴的y坐标每次减t*60
        sprite.position.y -= t*60;
        if (sprite.position.y < 0) {
            sprite.position.y = 600;
        }
    });
    requestAnimationFrame(loop);
}
loop();
```

相机镜头附近的雨滴偏大

相机在下雨的场景中，相机会渲染near~far范围的Sprite，距离相机0~near范围不会渲染，小部分Sprite会在相机镜头前经过，大家都知道透视投影远小近大，这时候相机near附近雨滴Sprite会显示比较大，你可以把near调整大一些，这样距离相机非常近的Sprite不会渲染。

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

`near`调整大一些，避免距离相机非常近的雨滴渲染非常大的现象。

```js
const camera = new THREE.PerspectiveCamera(30, width / height, 50, 3000);
```

下雪效果模拟

把雨滴代码中与雨滴贴图更换雪花纹理贴图，雪花下降速度可以适当调整。当然这也只是近似模拟，比如雪花随风飘动、角度旋转等等都没有模拟。

```js
const texture = new THREE.TextureLoader().load("./雪花.png");
```

← 2. 精灵模型标注场景(贴图)　　1. 后处理(发光描边OutlinePass)→