

## 🔗 21. 查看或设置glTF几何体顶点

前面给大家介绍过几何体 `BufferGeometry` 对象，以及用来表示 `BufferGeometry` 顶点数据的属性缓冲对象 `BufferAttribute`。

```
geometry.attributes.position = new THREE.BufferAttribute();
geometry.attributes.normal = new THREE.BufferAttribute();
geometry.attributes.color = new THREE.BufferAttribute();
geometry.attributes.uv = new THREE.BufferAttribute();
geometry.index = new THREE.BufferAttribute();
```

js

下面通过加载一个外部glTF模型为例，给大家演示如何获取、修改外部模型的几何体顶点数据，也就是说获取、修改 `BufferAttribute` 对象里面的顶点数据。

### 获取glTF模型几何体顶点数据

```
loader.load("../地形.glb", function (glTF) { //glTF加载成功后返回一个对象
    model.add(glTF.scene); //三维场景添加到model组对象中
    //mesh表示地形网格模型
    const mesh = glTF.scene.children[0];
    // 顶点数据
    const att = mesh.geometry.attributes;
    console.log('att', att);
    // 顶点位置数据
    const pos = mesh.geometry.attributes.position;
    console.log('pos', pos);
})
```

js

### 几何体顶点索引属性 `geometry.index`

three.js大部分自带的几何体API默认有 `.index` 属性，不过外部加载的glTF等模型，`geometry.index` 数据可能有，也可能没有，具体看外部模型情况。

```
console.log('index', mesh.geometry.index);
```

js

## 顶点数量 `BufferAttribute.count`

```
const count = pos.count; //几何体顶点数量
console.log('count', count);
```

js

### `.getX()`、`.getY()` 和 `.getZ()`

`BufferAttribute` 对象具有 `.getX()`、`.getY()` 和 `.getZ()` 方法。

`BufferAttribute` 共有顶点数量 `count`，通过 `.getX(i)` 方法可以获取第 `i+1` 个点的x分量，`i`的范围就是`[0,count-1]`。

```
const pos = mesh.geometry.attributes.position;
// 获取几何体第一个顶点的x坐标
const x = pos.getX(0);
console.log('x', x);
```

js

`.getY()`、`.getZ()` 是获取顶点数据 `BufferAttribute` 的y、z分量,使用方式和 `.getX()` 方法一样。

### `.setX()`、`.setY()` 和 `.setZ()`

通过 `.getY()` 是获取顶点y坐标，通过 `.setY()` 是设置顶点y坐标。

`pos.setY(0,100)` 把索引为0，也就是第一个顶点的y坐标设置为100.

```
const pos = mesh.geometry.attributes.position;
pos.setX(0,100);
```

js

## 批量重置几何体顶点y坐标

```
loader.load("../地形.glb", function (glTF) {
    model.add(glTF.scene);
    //mesh表示地形网格模型
    const mesh = glTF.scene.children[0];
    // 顶点位置数据
```

js

```
const pos = mesh.geometry.attributes.position;
const count = pos.count; //几何体顶点数量
// 批量设置所有几何体顶点位置的y坐标
for (let i = 0; i < count; i++) {
    const y = pos.getY(i); //获取第i+1个顶点y坐标
    pos.setY(i, y*2) //设置第i+1个顶点y坐标为自身2倍
}
})
```

---

← [20. Color颜色渐变插值](#)

[22. 山脉地形高度可视化](#) →