

## 🟡 19. 一段曲线颜色渐变

可以通过几何体顶点颜色 `.attributes.color` 数据，实现一段曲线颜色渐变效果。

### 样条曲线几何体

首先提供一个样条曲线生成的几何体。

```
const geometry = new THREE.BufferGeometry(); //创建一个几何体对象
// 三维样条曲线
const curve = new THREE.CatmullRomCurve3([
  new THREE.Vector3(-50, 20, 90),
  new THREE.Vector3(-10, 40, 40),
  new THREE.Vector3(0, 0, 0),
  new THREE.Vector3(60, -60, 0),
  new THREE.Vector3(70, 0, 80)
]);
const pointsArr = curve.getSpacedPoints(100); //曲线取点
geometry.setFromPoints(pointsArr); //pointsArr赋值给顶点位置属性
```

### 设置几何体顶点颜色 `.attributes.color`

曲线上不同位置设置不同的颜色，你可以按照点在曲线上索引值顺序来计算每个点对应的颜色数据。

```
const pos = geometry.attributes.position;
const count = pos.count; //顶点数量
// 计算每个顶点的颜色值
const colorsArr = [];
for (let i = 0; i < count; i++) {
  const percent = i / count; //点索引值相对所有点数量的百分比
  //根据顶点位置顺序大小设置颜色渐变
  // 红色分量从0到1变化，蓝色分量从1到0变化
  colorsArr.push(percent, 0, 1 - percent); //蓝色到红色渐变色
}
//类型数组创建顶点颜色color数据
```

```
const colors = new Float32Array(colorsArr);  
// 设置几何体attributes属性的颜色color属性  
geometry.attributes.color = new THREE.BufferAttribute(colors, 3);
```

## 线模型渲染渐变色曲线

```
const material = new THREE.LineBasicMaterial({  
  vertexColors: true, //使用顶点颜色渲染  
});  
const line = new THREE.Line(geometry, material);
```

---

← 18. 几何体顶点颜色数据

20. Color颜色渐变插值 →