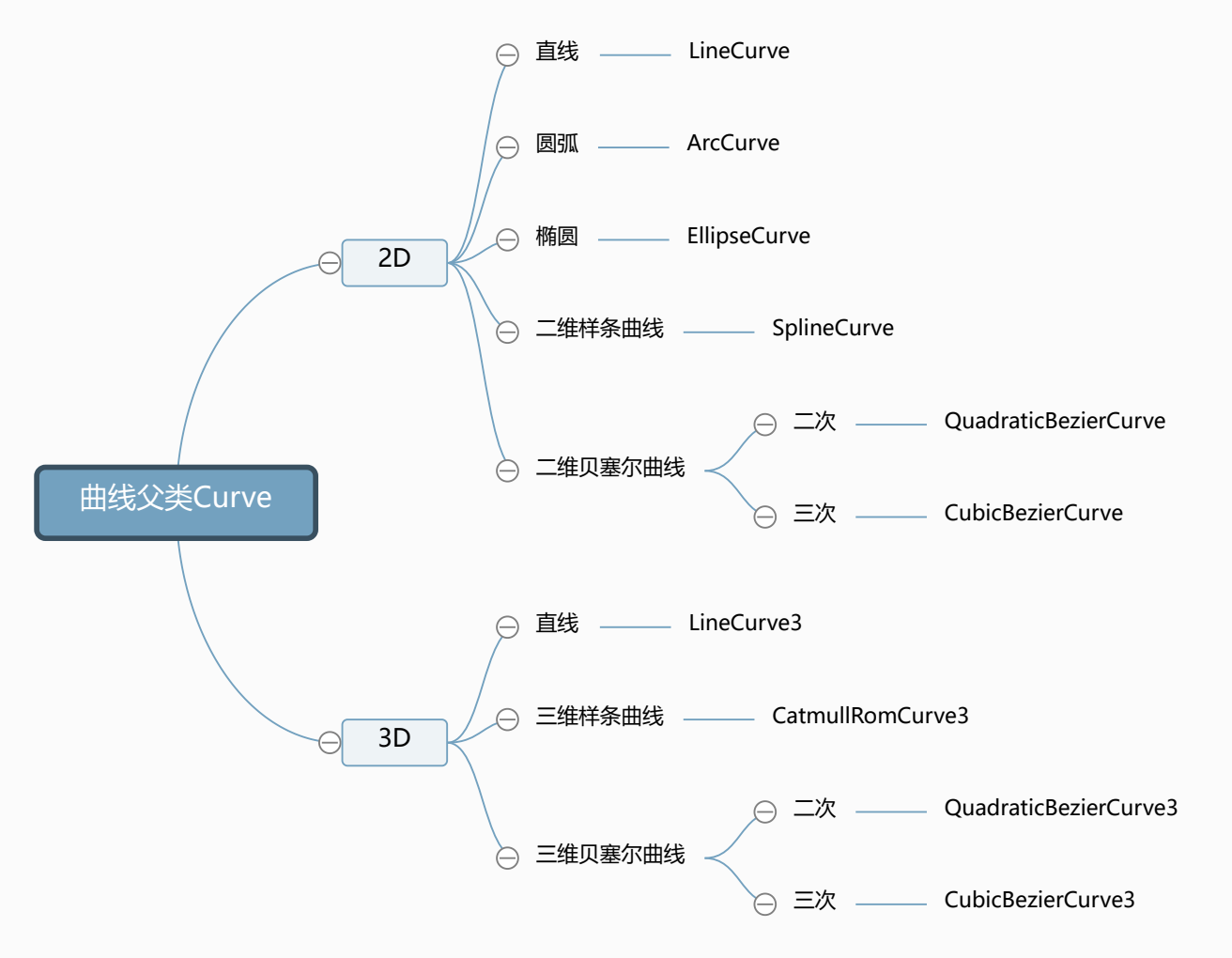
郭隆邦 2023-01-29

# 3. 曲线Curve简介

上节课程绘制一个圆弧线是自己通过算法实现，其实threejs提供了很多常用的曲线或直线API，可以直接使用。这些API曲线都有一个共同的父类`Curve`。




## 椭圆`EllipseCurve`例子

曲线API的使用，具体语法可以查询文档，下面以椭圆为例，给大家绘制一个椭圆曲线效果。

```js
// 参数1和2表示椭圆中心坐标  参数3和4表示x和y方向半径
const arc = new THREE.EllipseCurve(0, 0, 100, 50);
```

## 曲线 `Curve` 方法 `.getPoints()`

椭圆弧线 `EllipseCurve` 的父类是曲线 `Curve`,自然会继承父类曲线 `.getPoints()` 方法，通过 `.getPoints()` 可以从曲线上获取顶点数据。

通过方法 `.getPoints()` 可以从曲线上按照一定的细分精度返回沿着曲线分布的顶点坐标。细分数越高返回的顶点数量越多，自然轮廓越接近于曲线形状。方法 `.getPoints()` 的返回值是一个由二维向量 `Vector2` 或三维向量 `Vector3` 构成的数组， `Vector2` 表示位于同一平面内的点， `Vector3` 表示三维空间中一点。

```js
//getPoints是基类Curve的方法，平面曲线会返回一个vector2对象作为元素组成的数组
const pointsArr = arc.getPoints(50); //分段数50，返回51个顶点
console.log('曲线上获取坐标',pointsArr);
```

## `.setFromPoints()` 提取曲线坐标数据

把数组pointsArr里面的坐标数据提取出来，赋值给 `geometry.attributes.position` 属性

```js
const geometry = new THREE.BufferGeometry();
geometry.setFromPoints(pointsArr);
console.log('geometry.attributes',geometry.attributes);
```

## 点模型查看曲线上顶点坐标

```js
// 点材质
const material = new THREE.PointsMaterial({
    color: 0xffff00,
    size: 10.0 //点对象像素尺寸
});
// 点模型
const points = new THREE.Points(geometry, material);
```

## 曲线 `Curve` 方法 `.getSpacedPoints()`

通过`.getSpacedPoints()`和`.getPoints()`一样也可以从曲线Curve上返回一系列曲线上的顶点坐标。

通过`.getSpacedPoints()`是按照曲线长度等间距返回顶点数据，`.getPoints()`获取点的方式并不是按照曲线等间距的方式，而是会考虑曲线斜率变化，斜率变化快的位置返回的顶点更密集。

你可以通过案例源码测试对比，分别两种获取顶点方式曲线坐标，然后使用点模型渲染，观察点的分布规律。

```js
const geometry = new THREE.BufferGeometry();
geometry.getSpacedPoints(pointsArr);
console.log('geometry.attributes',geometry.attributes);
```

如果你有等间距取点的需求，可以选择`.getSpacedPoints()`方法，如果没有，就可以使用`.getPoints()`方法

## 线模型绘制曲线

```js
// 线材质
const material = new THREE.LineBasicMaterial({
    color: 0x00ffff
});
// 线模型
const line = new THREE.Line(geometry, material);
```

← 2. 几何体方法.setFromPoints()　　4. 椭圆、圆→