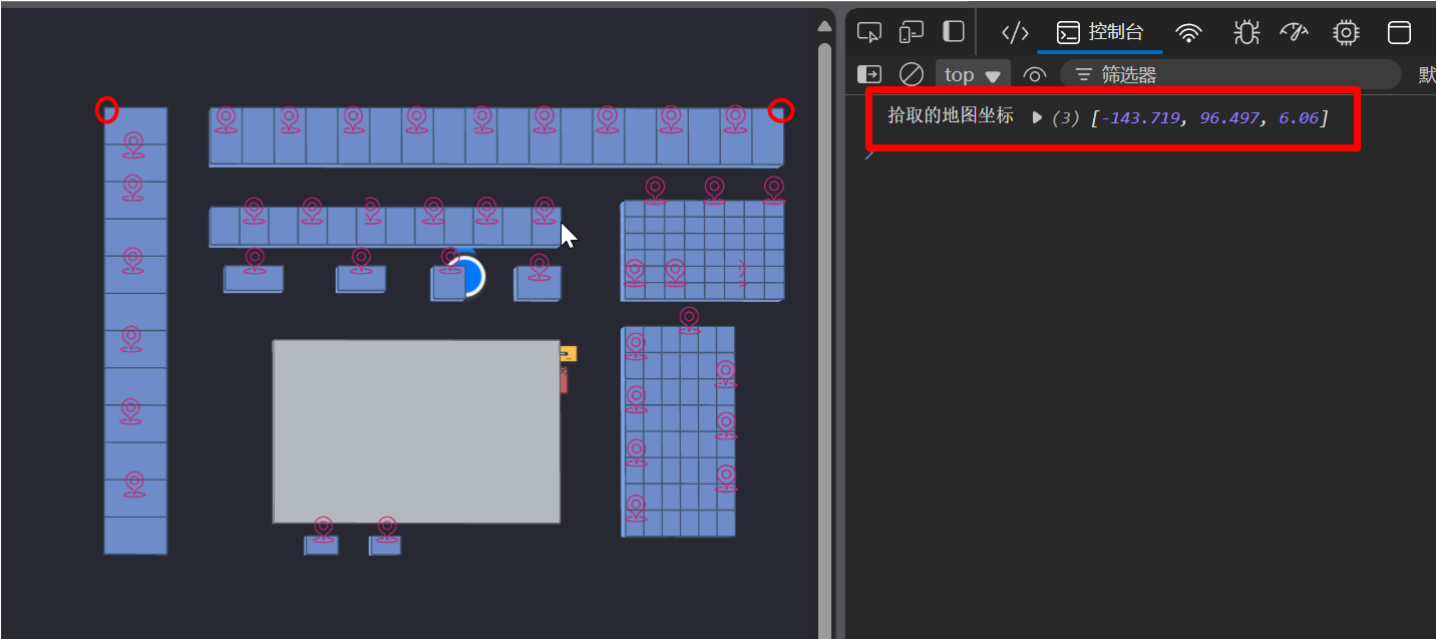


# 经纬度坐标与oomap平面坐标的映射

oomap地图坐标采用**两点坐标映射法**,可以简单快捷的完成与经纬度之间的坐标映射关系

## 一. 拾取oomap地图坐标点

- 1. 在oomap地图上,找到两个关键点, 并在地图上点击, 因为注册了地图点击事件,所以可以在控制台中获取点击位的地图坐标点



- 2. 构建坐标映射所需的oomap地图坐标数据

```
//////////////////// 用于坐标映射 //////////////////////
// 拾取的oomap地图上的平面坐标
let map_coord = {
  center: [-143.719, 96.853, 0],
  x_pos: [130.103, 96.494, 0]
}
```

第一个点

第二个点

其中 `center` 为第一个点的坐标数组, `x_pos` 为 横坐标第二个点 的坐标数组

## 二. 拾取对应的GPS经纬度坐标

- 1. 在**腾讯地图**拾取与oomap地图坐标点对应的经纬度坐标点

这两上坐标点不需要是水平的. 可以为任意角度的. SDK中会自动的进行旋转与缩放的适配



## 2. 构建坐标映射所需要的经纬度坐标数据

注意: 这里需要使用SDK提供的坐标将经纬度(lng,lat)转为 **墨卡托坐标** , 经度在前,纬度在后

```
18
19 // 需要将经纬度转为墨卡托坐标
20 let m1 = om.utils.coordTransform.lngLatToMercator(117.214555, 39.095636)
21 let m2 = om.utils.coordTransform.lngLatToMercator(117.215305, 39.09527)
22
23 // 经纬度对应的墨卡托坐标点
24 let real_coord = {
25   center: [m1.x, m1.y, 0],
26   x_pos: [m2.x, m2.y, 0]
27 }
```

## 3. 使用 **om.CoordProjection** 类构建坐标映射实例

```
28
29 // 构建坐标映射
30 const cp = new om.CoordProjection(real_coord, map_coord)
31
32 // 将经纬度转为墨卡托坐标
33 function transCoord(lng, lat) {
34   let coord = om.utils.coordTransform.lngLatToMercator(lng, lat)
35   return cp.realToMap(coord.x, coord.y)
36 }
37
```

使用 `cp.realToMap` 方法完成从经纬度坐标到ooomap地图坐标的转换, 返回类型为 `Vector3` 类型, 这里因为需要从 `经纬度` -> `web墨卡托` -> `ooomap平面坐标` 的转化, 所以可以将这个过程封装成一个方法

```
transCoord{lng, lat})
```

同理可以使用 `cp.mapToReal` 方法完成从ooomap地图坐标到经纬度坐标的转换, 返回的是 `Vector3` 类型的墨卡托坐标

可以通过 `om.utils.coordTransform.mercatorToLngLat( p.x, p.y )` 方法将墨卡托坐标转化为经纬度坐标