需求：

你可以讲一下大地图分块吗？

设计：

首先，先说一下为什么要地图分块，如果说场景里的地形非常大，加上里面的各种模型、贴图、碰撞、渲染等等 这将是一笔很庞大的cpu、gpu、内存开销，所以我们需要对地形进行分割成多个地形块 ，这块我们就使用地图分块处理，通过编辑器对地形进行分割成合适的地图块，切割的地图块大小是根据我们人物视野来进行裁切的，到我们加载地图块这块呢，加载规则是根据人物所在地图块下标来计算的，下标发生变化，就去刷新地图块，这块我们也将它做为四个层来实现，我们人物当下脚下那块叫当前层，再去加载周围八个块，称为周围层（显示层），这八个块后会有十六个块，称为预加载层也称预显示层，这十六个块后呢就是我们的卸载层，说下我们这四个层，首先是我们的必显示层也就是当前和周围层，这是必须显示的，我们的缓存层则是提前加载到内存中，当地图块发生变化时快速显示衔接，防止视觉上的穿帮，也是为了避免频繁实例和释放造成的内存波动，卸载层则是卸载不必要显示的地图块，也就是可以释放的地图块，因为缓存层作为移动位置的时间延长，避免频繁实例和释放带来GC可能，卸载的规则则是看资源管理是如何规划的，地图上的资源这块则是采用了四叉树进行管理，把地图划分为一个层次结构，每个地图块分为四个节点，如果节点里面有资源就划分，没有则不用，如果资源太大，模型横跨了几个节点该怎么办，资源的显示就算是跨越了几个节点，它实际也只存在于一个节点中，四叉树的显示不是根据资源大小，而是根据资源的中心点来显示的，跨越其他节点是很正常的，资源出现的条件则是一个地图块是一个区域，一个区域分成四个节点，在这个节点区域绘制包围盒，碰撞后显示该节点内所有资源，使用四叉树也是它比较合适，使用四叉树可以提高性能和效率。帮助我们快速确定场景中的物体位置、进行碰撞检测、剔除不可见物体以及进行空间查询等操作。地图优化这块还可以用LOD技术，用若干个不同精度的模型来表示同一对象的技术，就是我们视野较远时提供低精度模型，较近时则提供高精度模型，虽然采用LOD会损失一定的逼真程度，但可以减少场景中绘制多边形的数据，降低了渲染压力，与它提高的性能来说，也是可以接受的。

再是关于地形如果不是规则的,就是不规则形状该如何切块的？我们的方案是取地图的最大长宽高，进行切块，然后标识好是可行走区域还是不可行走区域，区域的标识是地编或者主美那边做的 是有个地图编辑器，根据角色的步长和地图的尺寸进行划分，会生成一个配置表，表中是地图区域的一个状态，这就是我们项目中对地图分块和使用