

地图寻路网格数据结构

2017年2月17日 10:27

1.得到所有地形prefab上的有效点

对于一个不规则的三维地形prefab，暂时的储存思路是，先将其网格上的点投影到xz平面上（即只考虑xz坐标，先忽略y坐标），得到它所在的矩形的边界信息（left，right，top，bottom），并根据地形prefab的rotation和position对所得到的边界信息进行平移和旋转，之后根据边界信息和所设定的cubeWidth，生成MxN的矩形矩阵（Dictionary<index,data>）。对矩阵中每个矩形的中心点，通过使用在上方使用垂直向下的RayCast是否与地形prefab碰撞，判断这个点是否位于地形上，如果在地形上，则对其标记有效。最终再遍历一次整个矩阵，删除所有不在地形上的点。得到所有有效点。

2.压缩储存

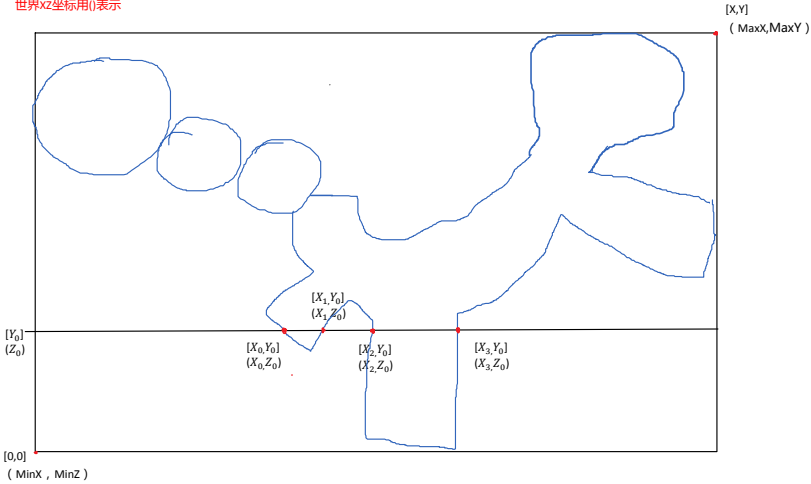
对于所有的有效点，对于每一行数据，储存为【起始点坐标；起始index，终点index；起始index，终点index】，这样就可以缩减大部分需要储存的数据。

3.要注意的问题

由于实际并不会储存MxN矩阵中的无效点，只是靠index（row * cols + col）来作为每个格子的唯一标识。所以在寻路算法中，需要通过计算才能得到目标点的行列，并且搜索完成之后，根据所有路径上点的行列，算出index，再通过index找出具体点坐标。
算法的最开始，传入的是起始点和终点的世界坐标，浮点数，再通过计算换算成整数的行列时，会有误差，这种误差在格子小的时候（0.3）可能并无影响,但是对于边长为1的格子的话，会发生左边转换时行列误差的情况，目前发现只会向x轴负向点击和z轴负向点击时会发生这种情况。拟解决问题的方法：1.看是否起始点的行列计算时有误差。2.看看为什么只有点击负轴的时候会出现这种情况。

网格坐标用()表示

世界xz坐标用[]表示



图中蓝色为地图边界上的有效点的大致位置，地图内部的有效点太多，已经省略。
对于一行信息，只保存(X0,Z0)，以及[X0,Y0]、[X1,Y0]、[X2,Y0]、[X3,Y0]的index即可