

1. 算法流程

- 1.构建格栅地图
- 2.将每个格栅初始化为 GridNode
- 3.将 start_GridNode 加入 Openlist
- 4.将 start_GridNode 从 Openlist 取出,扩展 start_GridNode 的非障碍物和非 Close_list 中 Node 的近邻,并将他们加入 Openlist
- 5.取出 Openlist 中 fscore 最小的 Node 取出,按要求扩展近邻节点
- 6.近邻节点中 id = 0 的节点,加入 Openlist, id = 1 的 Node,有机会更新更小的 fscore, id = -1 的 Node,不处理
- 7.当 current_Node 为终点时,算法终止,通过不断追溯 camefrom 节点,生成路径

2.统计欧式距离

- 1.78.791932 ms, path cost: 7.082551 m
- 2.65.918046 ms, path cost if 6.555416 m

3.统计曼哈顿距离

- 1.78.989823 ms, path cost if 7.654658 m
- 2.55.448749 ms, path cost if 6.917546 m
- 3.67.432632 ms, path cost if 5.462596 m

4.统计 diagonal 距离:

- 1.79.578047 ms, path cost if 7.064636 m
 - 2.59.147040 ms, path cost if 7.630321 m
 - 3.71.359858 ms, path cost if 5.932585 m
- Tie Breaker:
- 1.67.220971 ms, path cost if 6.228281 m
 - 2.74.208609 ms, path cost if 7.630321 m
 - 3.69.745797 ms, path cost if 5.789051 m
 - 4.55.377999 ms, path cost if 4.794113 m

结论:1.3种启发式距离差距不大, Tie_Breaker 明显效率更高.

