

算法性能分析报告

1. 概述

本报告对迷宫生成与路径可视化工具中的核心算法进行性能分析。该工具使用Python实现，结合Pygame进行可视化展示，主要包含迷宫生成算法和路径查找算法两部分。

2. 算法复杂度分析

2.1 迷宫生成算法

迷宫生成采用随机深度优先搜索(DFS)算法，主要步骤如下：

- 初始化迷宫矩阵(全为墙)
- 随机选择一个起点
- 递归地"打通"相邻的墙，形成通道
- 直到所有可达区域都被访问

时间复杂度分析：

- 初始化矩阵： $O(n^2)$ ， n 为迷宫边长
- 主生成循环： $O(n^2)$ ，每个单元格最多被访问一次
- 总体时间复杂度： $O(n^2)$

空间复杂度分析：

- 存储迷宫矩阵： $O(n^2)$
- 递归调用栈：最坏情况下 $O(n^2)$
- 总体空间复杂度： $O(n^2)$

2.2 路径查找算法

路径查找同样采用DFS算法：

- 从起点开始探索
- 尝试四个方向(右、下、左、上)
- 遇到死路时回溯
- 直到找到终点

时间复杂度分析：

- 最坏情况下需要遍历所有通道： $O(n^2)$
- 平均情况下： $O(n^2)$

空间复杂度分析：

- 存储访问标记矩阵： $O(n^2)$
- 路径栈：最坏情况下 $O(n^2)$
- 总体空间复杂度： $O(n^2)$

3. 结论

当前实现的迷宫生成与路径查找算法在中小规模迷宫(30×30 以内)表现良好，能够满足实时交互的需求。对于更大规模的迷宫，建议采用更高效的算法或进行性能优化。