编号



面向对象程序设计I课程设计报告

二级学院 计算机科学与工程学院

专 业 计算机科学与技术

班 级 计算机科学与技术二班

学生姓名 张晓君 学号 11503070234

指导教师 刘恒洋

时 间 2016年6月

目录

[目录 2](#_Toc455103278)

[迷宫鼠 3](#_Toc455103279)

[1 功能说明 3](#_Toc455103280)

[1.1 问题描述 3](#_Toc455103281)

[1.2 题目要求 3](#_Toc455103282)

[1.3 功能图 3](#_Toc455103283)

[2 算法说明 4](#_Toc455103284)

[2.1 生成迷宫 4](#_Toc455103285)

[2.2 寻找路径 5](#_Toc455103286)

[2.3 遍历迷宫 6](#_Toc455103287)

[2.4 最短路径 6](#_Toc455103288)

[3 系统模型 7](#_Toc455103289)

[4 系统测试 8](#_Toc455103290)

[5 课程设计总结 9](#_Toc455103291)

[5.1 程序亮点 9](#_Toc455103292)

[5.2 程序不足 9](#_Toc455103293)

迷宫鼠

# 1 功能说明

## 1.1 问题描述

用JAVA实现电脑鼠走迷宫的程序，一个假想的小车能在图示的迷宫中穿行输出其可能的组合式。

## 1.2 题目要求

根据国际比赛规则，电老鼠走迷宫分为三个阶段

从起点走到终点从终点

进一步遍历完整个迷宫，获得整个迷宫的地图（墙和通路）

从起点选择最短路径冲刺到终点

因此程序分为三部分

从任意一点走到另外给定点

遍历完整个迷宫的程序

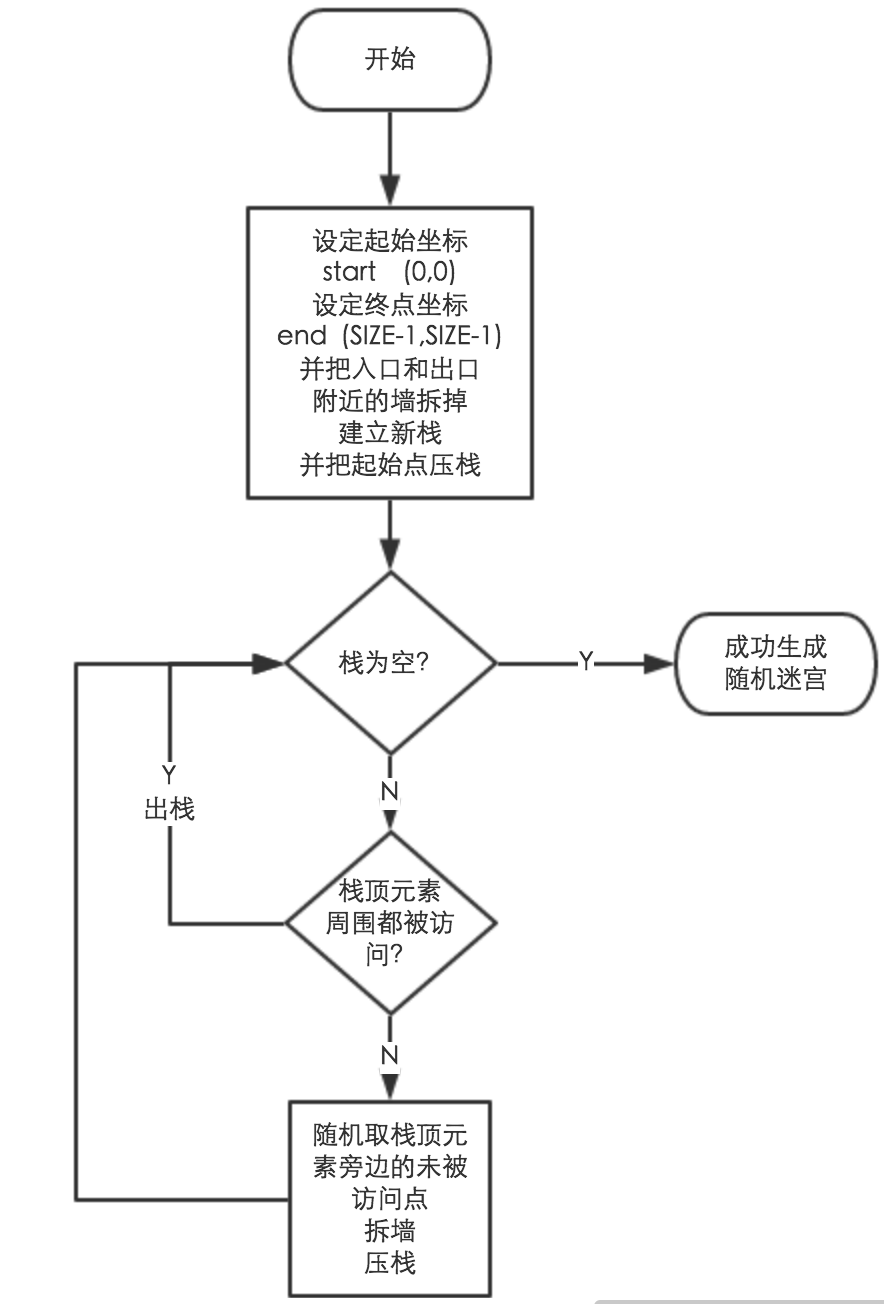
计算最短路径（计算等高表，按路径行规定走）

## 1.3 功能图

# 2 算法说明

## 2.1 生成迷宫

关键字：递归回溯

首先将迷宫分成若干个正方形的单元格，并选中左上角（0，0）作为起始点（start）。将正被访问的单元格标记为已访问,得到它所有相邻单元格。在这些相邻的单元格中随机选择一个，如果这个被选中的单元格没有被访问过，那么移掉正被访问单元格和被选中单元格之间的墙体，并将这个被选中单元格作为正被访问单元格。如果正被访问单元格的所有相邻单元格都被访问过，那么在所有被访问过的单元格（这里指迷宫中所有已被访问过的单元格）中选上一个被访问的作为正被访问单元格，如此递归下去，直到迷宫中所有的单元格都被访问过为止（即访问到终点）。这样迷宫就生成了。

## 2.2 寻找路径

先将老鼠置于起始点（start）。将正老鼠所在的单元格标记为已访问,得到它所有相邻的没有被访问过的单元格。从这个（些）单元格中随机选一个，移掉正被访问单元格和被选中单元格之间的墙体，并将老鼠移动到这个被选中单元格。如果老鼠旁边没有未被访问过的单元格，那么在所有被访问过的单元格（这里指迷宫中所有已被老鼠访问过的单元格）中选上一个被访问的作为正被访问单元格，如此递归下去，直到迷宫中的老鼠访问到终点。这样迷宫就走完了。

## 2.3 遍历迷宫

在之前寻找到路径的基础上，探索整个迷宫地图。原理和之前的差不多，只是终止条件变成了栈为空（即老鼠访问了迷宫的所有格子）。

## 2.4 最短路径

由于生成迷宫时使用的算法可以保证生产的随机迷宫有且只有一条通路，所以在寻找最短路径时只需要把寻找路径过程中找到的迷宫通路去除死路即可以得到最短路径。

事实上，我们的栈已经把最短路径存放好了，我们只需要稍作处理即可。

# ../Code/11503070234/diagram.png3 系统模型

# 4 系统测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 001 | | 版本号 | | 0.1 |
| 测试环境 | OS X 11.11 | | | | |
| 用例名称 | Reigning’S Maze | | | | |
| 前提条件 | Java 1.8 + | | | | |
| 测试步骤 | 打开程序，测试是否存在未知BUG | | | | |
| 输入数据 | 5x5 50 | | | | |
| 预期输出 | 5x5的迷宫 | | | | |
| 实际输出 | 5x5的迷宫 | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 张晓君 | 设计日期 | | 16.06.27 | |
| 测试人 | 张晓君 | 测试日期 | | | 16.06.28 |
| 再测试人 | 无 | 再测试日期 | | | 无 |
| 问题修改摘要 | 无 | | | | |
| 修改人 | 无 | | 修改日期 | | 无 |

# 5 课程设计总结

本次课程设计时间紧任务重，对于我来说是个不小的挑战。虽然算是完成了，但仍有一些做的不是很好需要优化。

## 5.1 程序亮点

1、格子的访问状态和墙的面数使用4位二进制数据（由于是JAVA所以使用了整形）储存，使用位运算进行数据运算。众所周知，位运算是基于系统底层级别的运算，在现在的计算机中，位运算比其他运算更为迅速，也更能优化程序性能，大量节省CPU占用时间，避免程序未响应。

2、提高空间复杂度以降低时间复杂度。现在的计算机的内存普遍都很大，参考OS X的内存管理制度，我们完全可以提高内存的使用率来换取用户对于程序的一些体验。程序中有一些代码片段会占用一些看起来不必要的内存，但是其对于减少用户等待时间是非常重要的。

3、使用了JAVAFX作为了GUI语言并且使用了动画编程SequentialTransition 类作为动画编程的核心。在JAVAFX资料少和课本内容不全面的情况下，我通过阅读JAVA源代码及官方英文文档掌握了SequentialTransition 类的用法并且获得了不错的动画效果。

4、尽量避免使用访问器（getter）和修改器（setter）。我之前开发安卓的经验告诉我，getter和setter往往是安卓程序卡顿停滞的重要原因。在本程序中，尽量避免了使用getter和setter，性能有所优化。

## 5.2 程序不足

1、由于没有使用getter和setter，时间不是很足，直接后果是类的封装太差了，不利于后期维护和调试。

2、生成迷宫的算法单一，只能生成有且仅有一条通路的迷宫，最短路径的算法无法得到实现和体现。后期应该添加其它的生成迷宫算法，获得具有多条通路的迷宫，这样就可以很好的体现最短路径的算法。

3、占用内存偏高。因为程序是JAVA编写，加上一些占内存的代码及动画效果，不推荐在05年以及之前的电脑上运行。