编号



面向对象程序设计

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 苏业纯 学号 11503070233

指导教师 刘恒洋

时 间 2016年6月

目录

[第一章：功能说明和总体设计 3](#_Toc455094284)

[1.1 已完成功能说明： 3](#_Toc455094285)

[1.2 系统结构图（总体设计）： 3](#_Toc455094286)

[第二章：算法说明 4](#_Toc455094287)

[2.1 生成迷宫的算法： 4](#_Toc455094288)

[2.2寻找路径的算法： 6](#_Toc455094289)

[2.3遍历迷宫的算法: 8](#_Toc455094290)

[第三章：功能设计 10](#_Toc455094291)

[所有类的名称和简单介绍： 10](#_Toc455094292)

[每个类的详细介绍：（其中每个类中的数据的访问器和修改器省略） 10](#_Toc455094293)

[绘制迷宫的实现方法： 12](#_Toc455094294)

[动画效果的实现： 12](#_Toc455094295)

[清空迷宫的实现： 12](#_Toc455094296)

[退出游戏的实现： 12](#_Toc455094297)

[第四章：类图以及协作图（系统静态模型和动态模型） 13](#_Toc455094298)

[一、各个类的UML图 13](#_Toc455094299)

[二、各个类之间的协作图： 15](#_Toc455094300)

[第五章：个人总结 16](#_Toc455094301)

# 第一章：功能说明和总体设计

## 1.1 已完成功能说明：

1. 迷宫的随机生成：

能随机生成单元格像素宽度为10，迷宫行列长度为40\*40的迷宫。

1. 寻找路径：

能寻找从起点到终点的路径。

1. 遍历迷宫：

能走遍迷宫内的所有点获得迷宫地图。

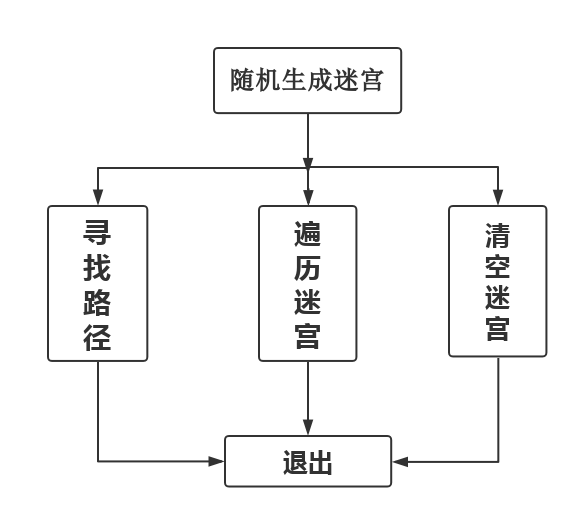
1. 清空迷宫：

能清除已经生成的迷宫，以便再次生成不同的随机迷宫。

1. 动态实现走迷宫的过程：

即运用动画效果，用一个小球模拟电脑鼠在计算机上展示出走迷宫的过程。

## 1.2 系统结构图（总体设计）：

****

# 第二章：算法说明

## 2.1 生成迷宫的算法：

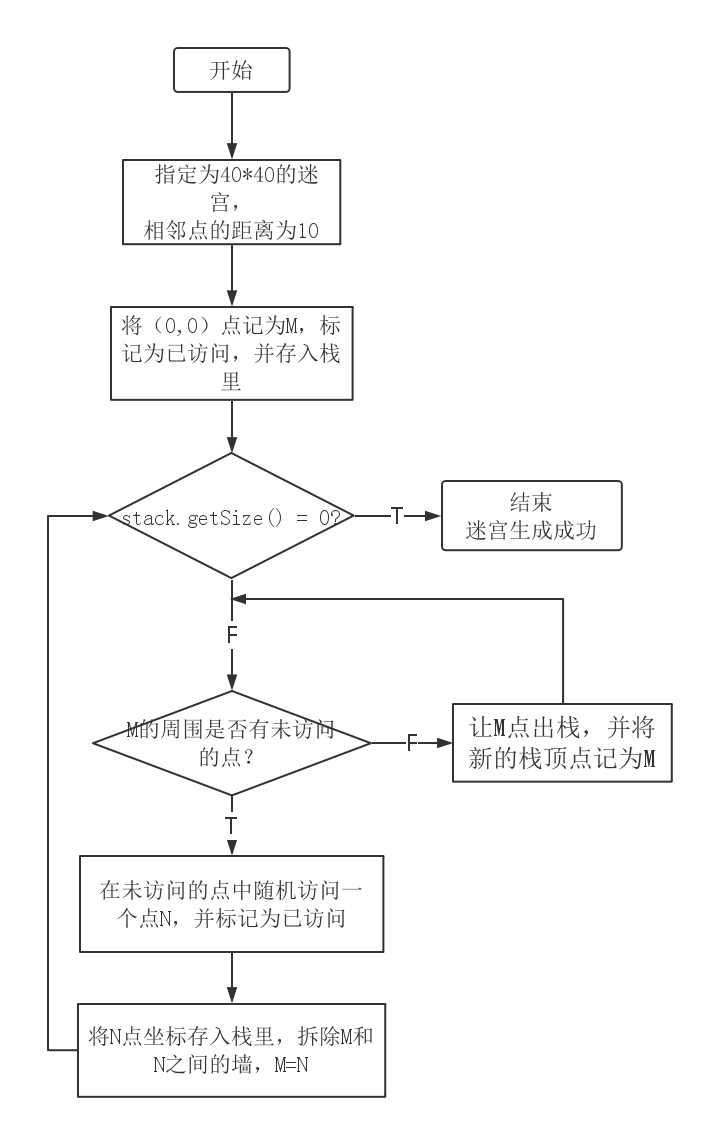
2.1.1算法思路：

指定40\*40大小的迷宫，从迷宫的（0,0）点开始访问,访问四个方向中的随机一个点(每访问到一个可访问的点,就去掉该点的那个方向的墙),被访问点继续以这种方识向下进行访问。对每个被访问的点都被标识为已访问,当一个点对某个方向进行访问时我们首先会判断被访问点是否已被访问,或者触到边界.如果该点四个方向皆已访问或已无法访问,就退回上一个点，上一个点继续这个过程。

2.1.2算法实现：

主要是利用栈,第一次,先把第一个点压进栈里,每访问到一个点,就把该点压进栈里,再对栈顶的点进行四个方向的随机访问,访问到新点,又把新点压进去,一旦这个点四个方向都无法访问了,就让该点退栈,再对栈顶的点的四个方向进行访问,以此类推,直到栈里的点都全部退出了,迷宫成功生成了。

2.1.3算法流程图：

****

## 2.2寻找路径的算法：

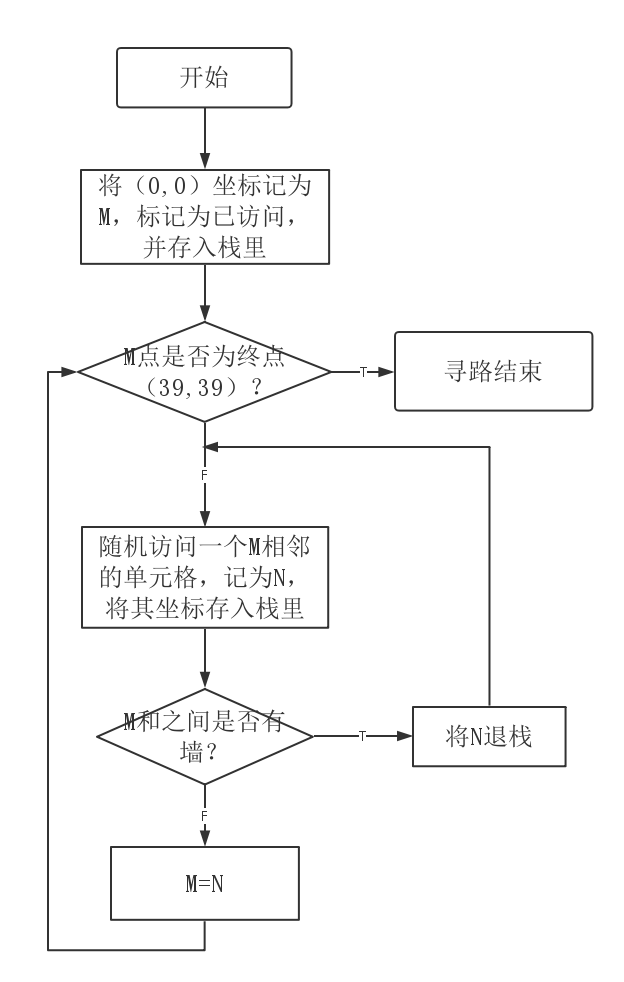
2.2.1算法思路：

从迷宫的第一个单元格开始，随机访问它相邻的四个单元格的其中一个格子，判断他们之间是否有通路，有通路则继续以这种方式往下访问，若无通路或者四个单元格都已经访问，则返回上一个单元格继续访问，直至访问到迷宫中的最后一个单元格。

2.2.2算法实现：

主要也是利用栈。将第一个单元格的行列坐标压入栈里，随机访问一个它相邻的四个单元格的一个，将其标记为已访问并将其坐标压入栈里，判断该单元格的坐标是否是终点坐标，若是，则停止访问，若不是，则判断该单元格与上一个单元格之间是否有通路，若有，则继续往下访问，若没有，则将该单元格的坐标从栈顶删除，并返回上一个单元格重新访问，直至访问至终点坐标，这样栈里面就存储了从起点到终点的可行路径。在访问时，利用右手法则进行访问。

2.2.3算法流程图：

****

## 2.3遍历迷宫的算法:

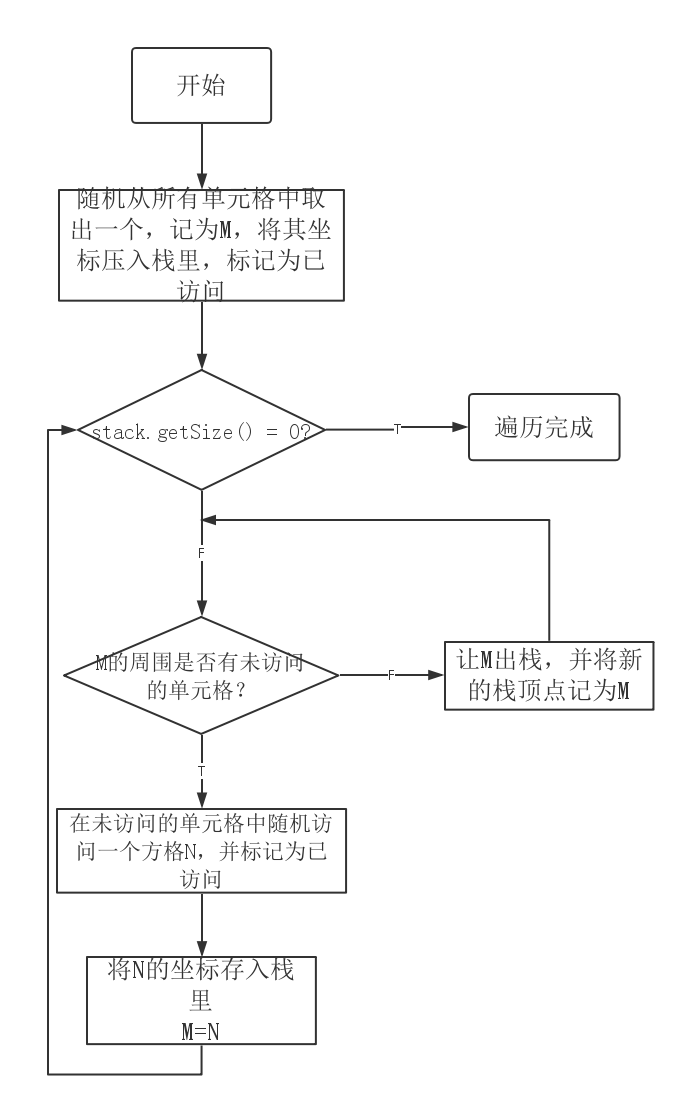
2.3.1 算法思路：

随机访问迷宫里的一个单元格，并标记为已访问，然后随机访问该单元格相邻的四个单元格，并标记为已访问。判断已访问的单元格数是否等于全部单元格的个数，若两者相等，则停止访问，遍历完成，若不相等，则继续往下访问。若正访问的单元格相邻四个单元格均已访问过，则随机从所有单元格随机再取出一个单元格继续访问，以此类推，直到所有单元格都被访问过为止。

2.3.2 算法实现：

主要利用栈。从所有单元格里随机取出一个单元格，标记为已访问，将其坐标压入栈里，接着随机访问一个相邻的单元格，并标记为已访问，将其坐标压入栈里。此时判断栈的长度是否等于等于0，若相等，则停止访问，遍历完成，若不相等，继续往下访问。若正访问的单元格相邻四个单元格均已访问过，则删除栈顶单元格的坐标，然后随机从所有单元格随机再取出一个单元格继续访问，以此类推，直到所有单元格都被访问过为止。采用左手法则进行遍历。

2.3.3算法流程图：

****

# 第三章：功能设计

## 所有类的名称和简单介绍：

1. UI类：主要用于显示界面和完成各个功能。
2. Maze类：主要用于创建迷宫。
3. mazePoint类：主要用于表示迷宫里面每一个点。
4. cell类：主要用于表示迷宫里面每一个单元格。
5. myStack类:主要用于存储单元格的行列坐标。

## 每个类的详细介绍：（其中每个类中的数据的访问器和修改器省略）

1. UI类**：**
2. Start(Stage primaryStage):重写start面板，
3. creatXY():绘制X,Y坐标
4. creatButton():控制按钮的生成，
5. paintMaze():绘制出迷宫，
6. paintCell(String string):绘制单元格的墙壁，
7. creatLine(int x1,y1,x2,y2):绘制单元格的一面墙，
8. paintball():绘制小球，
9. Mazefile():随机生成迷宫的数据文件，将其写入磁盘，
10. PaintMoveBall1(int i,int j,int m,int n):绘制寻找路径时小球的行走路线，
11. PaintMoveBall2(int i,int j,int m,int n):绘制遍历迷宫时小球的行走路线，
12. InputMaze():将迷宫地图导入栈中，
13. moveBall():控制小球移动，
14. travelMaze():遍历迷宫时控制动画的函数，
15. deleteMzae():清除迷宫，
16. exit():用于退出程序，
17. findWay():用于寻找路径的函数，
18. travel():用于遍历迷宫的函数。
19. Maze类：

迷宫的每个单元格长度为10，且迷宫长度为40\*40。类中提供各个数据的访问器和修改器。

1. mazePoint类：

每一个点都有上下左右四个属性，在迷宫中的坐标，还有每个点四面空白墙壁的数量。类中提供各个数据的访问器和修改器。

1. cell类：

每个单元格都有上下左右四个属性，在迷宫中的行列坐标，访问标记，以及每个单元格四面空白墙壁的数量。类中提供各个数据的访问器和修改器。

1. myStack类：

利用一个cell类的数组来存储单元格的坐标，栈的默认长度为1600。类中提供各个数据的访问器和修改器。

## 绘制迷宫的实现方法：

利用Line类创建，先画出坐标轴的线。然后再根据每个点四个面的信息画线。

## 动画效果的实现：

创建一个TimeLine类的对象，利用模拟的小球圆心坐标的移动画线，从而展示出动态的效果。

## 清空迷宫的实现：

利用面板的clear()方法清除装迷宫面板即可。

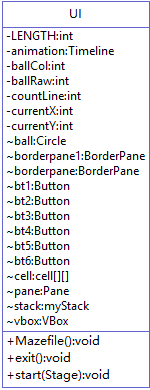
## 退出游戏的实现：

利用exit（）退出游戏即可。

# 第四章：类图以及协作图（系统静态模型和动态模型）

## 一、各个类的UML图

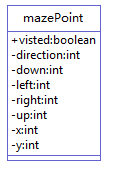
1、UI类：



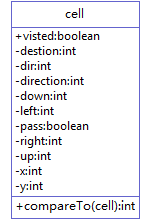
2．Maze类：



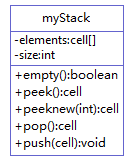
3. mazePoint类：



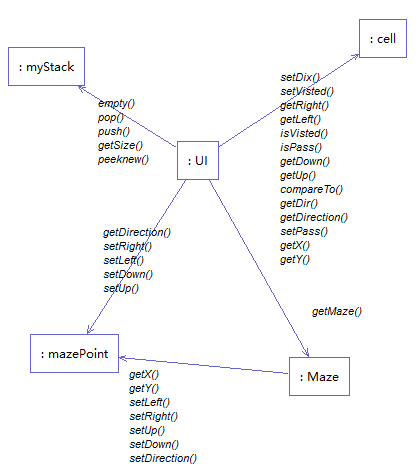
4．cell类：



5. myStack类：



## 二、各个类之间的协作图：



# 第五章：个人总结

1、通过这次JAVA程序设计，让我学会了两种很有用的算法。一是广度优先算法，二是深度优先算法。

2、通过这次JAVA程序设计，我发现自己的java知识还远远不够，还应该好好看书，继续学习。还有，我打代码的效率还是比较低，所以打代码的能力也有待提高。

3、我这次程序设计其实并未完全实现“最短路径”这一功能，让我很是懊恼，只不过值得高兴的是，最短路径的算法我已经非常清晰了，所以我下来之后会继续去实现，以达到完美。

4、这次程序设计过程中，我遇到了很多问题，也深深地苦恼过，幸好有老师和同学的帮助，我才得以完成至此。所以我想在此对帮助我的同学和老师致以真诚地感谢。

5、通过本次程序设计，我还懂得了遇到困难就要努力的思考，并去克服困难，而不应该自暴自弃。这对我个人以后的发展具有深远影响。