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 117039904

学生姓名 学号 11703990404

指导教师 傅由甲，黄贤英，曹琼，范伟

时 间 2018年10月

## 功能说明

Maze：迷宫类，用以设计迷宫，具有一些迷宫的典型特点。

MazeStack：迷宫栈类，用以存储迷宫中寻路时的节点。

ShowMaze：迷宫展示类，用以展示迷宫的界面程序类。

1. oneKey：ShowMaze类中的一键寻路。
2. oneStep：ShowMaze类中的单步寻路。
3. reset：ShowMaze类中的重置。
4. setStart：ShowMaze类中的设置指定点到指定点的寻路的起点和终点坐标。

**流程图：**

MazeSystem

ShowMaze

Maze

MazeStack

setStart

reset

oneStep

oneKey

## 算法说明

随机生成迷宫算法：随机生成迷宫。

迷宫栈算法：自主设计的一个栈用来存储迷宫路径节点。

Javafx界面程序（包括各种UI组件操作及事件驱动操作）：用来展示迷宫的算法。

**流程图：**

迷宫系统

随机生成迷宫

界面展示

界面操作

oneKey

setStart

reset

oneStep

栈存储路径

## 广度优先搜索算法流程图



图1

广度优先搜索阐述

1. 先添加当前点到列表，并标记当前位置，然后进入重复以下操作，直到head>=mouse（head是当前的编号，mouse是列表的大小）或把所有能走的点都放置trace并记录终点的trace编号。
2. 先从列表中获取当前编号的trace并得到当前坐标（trace作用是一个记录坐标和自己上一级的序号，以及到达当前位置所需步数）。
3. 判断当前位置是否为终点，若是，则记录终点trace的编号并是boolean变量pass为true。接着有一个for，将当前位置的四个方向下一点，分别判断下一点能否放置trace，若能，则标记当前位置，并添加到列表。
4. 最后编号head++。见图1

## 深度优先搜索算法流程图



图2

2.2 深度优先搜索算法阐述

起点先入栈，然后循环，先判断栈是否为空，为空退出（意味者无法到达终点），然后判断栈顶元素是否为终点，为终点清除栈并退出。然后进入for循环，判断栈顶元素四周的点是否有效（若超出边界，为墙，被标记，则无效），有效则入栈并被标记，然后退出for再重新循环，若四周都无效，则弹出栈顶元素。见图2

## 1.3 功能设计

界面布局

1.3.1.1 中间部分——迷宫地图

根据程序代码中的实际边界尺寸，showMaze类会在放置在BorderPane对象中央的Pane对象中绘制迷宫地图，且位置不会因为迷宫大小发生变化。

1.3.1.2 右边部分——指定点设定区

* 整体放在一个VBox中，从上到下分别是标签、文本域交替共8个。
* Label l1= new Label("EnterX：");
* Label l2= new Label("EnterY：");
* Label l3= new Label("ExitX：");
* Label l4= new Label("ExitY：")
* TextField t1，t2，t3，t4
* SetStart按钮：点击触发SetStart事件设置指定点的坐标

1.3.1.3 左边部分——栈内节点坐标（路径位置）

根据程序代码，showMaze类会在放置在BorderPane对象左边的带ScrollPane的Pane对象中绘制栈节点坐标，且节点不会被隐藏而看不见，它们随数量的增加逐渐下移，与此同时，Scroll会逐渐变短来适应。

3.1.4 下边部分——功能按钮操作

* 整体放在一个HBox中，从左到右分别是oneKey、oneStep、reset。。
* oneKey：一键寻路，寻找最短路径。
* oneStep：单步寻路。
* reset：重置。

1.3.2 各功能实现

1.3.2.2 按钮区面板——HBox

主要由三个Button构成，分别为oneKey，oneStep，reset。

1、一键寻路（oneKey）

调用public void oneKey ()，通过广度优先搜索算法设计的类返回最短路径，然后一键走到终点。

2、单步寻路（oneStep）

调用public void oneStep ()，中，利用深度优先搜索，每一步都调用绘画迷宫对象每一步都实时绘画到地图中，碰见死路会返回并可能产生一个标记点。

3、重置归零（reset）

在public void reset ()为该按钮设置一个处理器。当按下该按钮后调用绘画迷宫对象的clear()，该方法会重新绘画迷宫

## 1.3.1 系统静态模型

MazeStack

**-**

**-**

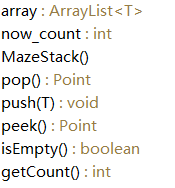
**+**

**+**

**+**

**++**

**+**



**+**

**+**

Maze

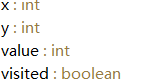


**-**

**-**

**--**

Point



**+**

**+**

**+**

**+**

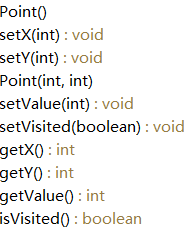
**++**

**+**

**+**

**+**

**+**



**-**

**-**

**-**

**-**

**-**

**-**

**-**

**-**

**-**

**-**

**-**

**-**

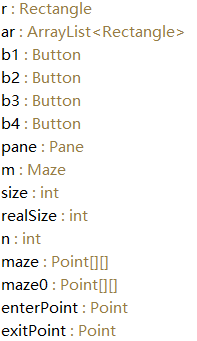
**-**

**-**

**-**

**-**

ShowMaze



**----------------**

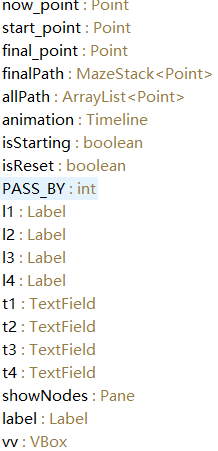
**-**

**-**

**-**

**-**

**-**



**++**

**+**

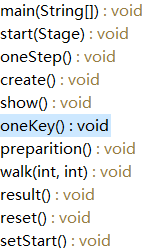
**+**

**+**

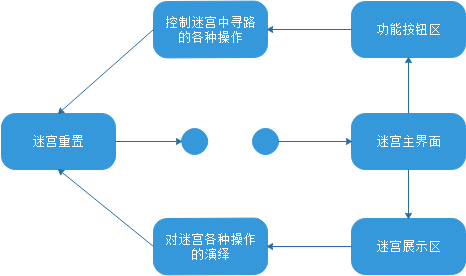
**++**

**+++**

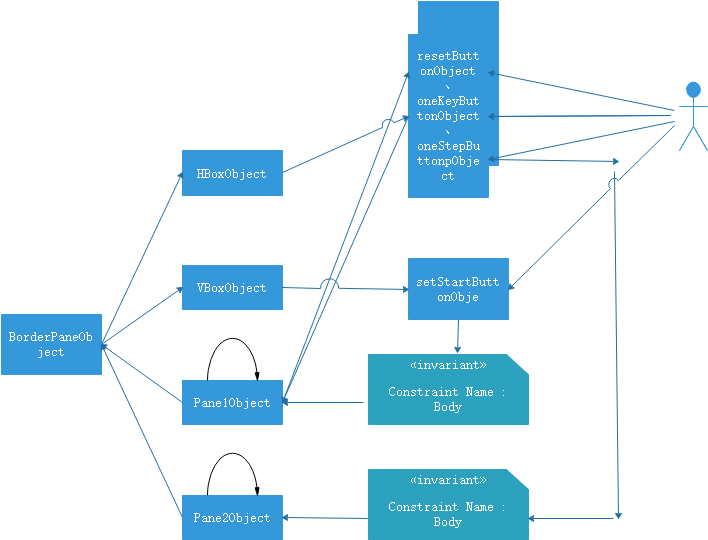
**+**



## 1.3.2 系统动态模型



**图3（系统状态图）**



**图4（协作图）**

1.4 系统测试

## 迷宫地图自动生成

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试环境 | Eclipse Oxygen.2 Release (4.7.2)、Java(TM) SE Runtime Environment (build 9.0.4+11)、Windows64位 | | | |
| 测试内容 | 自动生成迷宫测试 | | | |
| 测试结果 | 自动生成迷宫测试-成功 | | | |
| 问题描述 | 通过运行程序即产生随机迷宫，每次运行的迷宫都不一样，具有随机性 | | | |
| 设计人 | 邓小平 | 设计日期 | 2018.10.8 | |
| 测试人 | 邓小平 | 测试日期 | | 2018.10.8 |

## 迷宫地图一键寻路

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试环境 | Eclipse Oxygen.2 Release (4.7.2)、Java(TM) SE Runtime Environment (build 9.0.4+11)、Windows64位 | | | |
| 测试内容 | 迷宫一键寻路测试 | | | |
| 测试结果 | 迷宫一键寻路测试-成功 | | | |
| 问题描述 | 通过点击BorderPane对象下方的oneKey按钮来一键寻路 | | | |
| 设计人 | 邓小平 | 设计日期 | 2018.10.8 | |
| 测试人 | 邓小平 | 测试日期 | | 2018.10.8 |

## 迷宫地图单步寻路

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试环境 | Eclipse Oxygen.2 Release (4.7.2)、Java(TM) SE Runtime Environment (build 9.0.4+11)、Windows64位 | | | |
| 测试内容 | 迷宫单步寻路测试 | | | |
| 测试结果 | 迷宫单步寻路测试-成功 | | | |
| 问题描述 | 通过点击BorderPane对象下方的oneStep按钮来单步寻路，BorderPane对象左方的Pane对象中会显示路径节点坐标。 | | | |
| 设计人 | 邓小平 | 设计日期 | 2018.10.8 | |
| 测试人 | 邓小平 | 测试日期 | | 2018.10.8 |

## 迷宫地图重置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试环境 | Eclipse Oxygen.2 Release (4.7.2)、Java(TM) SE Runtime Environment (build 9.0.4+11)、Windows64位 | | | |
| 测试内容 | 迷宫地图重置测试 | | | |
| 测试结果 | 迷宫地图重置测试-成功 | | | |
| 问题描述 | 通过点击BorderPane对象下方的reset按钮来实现迷宫地图重置 | | | |
| 设计人 | 邓小平 | 设计日期 | 2018.10.8 | |
| 测试人 | 邓小平 | 测试日期 | | 2018.10.8 |

## 迷宫指定点寻路

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试环境 | Eclipse Oxygen.2 Release (4.7.2)、Java(TM) SE Runtime Environment (build 9.0.4+11)、Windows64位 | | | |
| 测试内容 | 迷宫指定点寻路测试 | | | |
| 测试结果 | 迷宫指定点寻路测试-成功 | | | |
| 问题描述 | 输入数据后，通过点击BorderPane对象右方的setStart按钮来实现迷宫指定起点和终点，然后通过点击BorderPane对象下方的oneKey按钮来一键寻路 | | | |
| 设计人 | 邓小平 | 设计日期 | 2018.10.8 | |
| 测试人 | 邓小平 | 测试日期 | | 2018.10.8 |

1.5 课程设计总结

通过此次课程设计，使我更加扎实地掌握了Java的相关知识，在设计过程中虽然遇到了一些问题，但经过一次有一次思考，一遍又一遍的检查终于找出了原因的所在，也暴露出了前期我在这方面的知识欠缺和经验不足。实践出真知，通过亲自动手制作，使我掌握的知识不再是纸上谈兵。过而能改，善莫大焉。在课程设计过程中，我们不断发现错误，不断改正，不断领悟，不断获取。同时，此次课程设计也让我明白了思路即出路，有什么不懂不明白的地方要及时请教或者上网查询，只要认真专研，动脑思考，动手实践，就没有弄不懂的知识。