

**综合实践报告**

**《面向对象程序设计》**

|  |  |  |
| --- | --- | --- |
| 课程目标2：熟练运用面向对象程序设计思想和技术，针对有一定复杂程度的工程问题，提出合理的设计方案，能够搜集、整理对方案有支撑作用的资料，并对设计方案进行分析、设计、编码、测试和调试；同时对软件测试结果，用文字、图表进行展示，分析运行和测试结果并给出合理的评价。（60%） | 课程目标4：熟练运用设计文稿、文档报告等方法，并具备一定的沟通和交流能力，准确表达个人观点，能够流畅地向教师和同学展示和讲解系统方案以及系统效果。（40%） | **总成绩/签名** |
|  |
|  |  |

**课程名称 面向对象程序设计综合实践**

**院系名称 计算机科学与工程学院**

**学 号 12103990306**

**姓 名 林吉**

**任课教师 刘恒洋**

**时 间 2022.6**

## 功能说明

1. **系统功能：包括prim算法生成迷宫**

**递归分割生成迷宫**

**从文件读取迷宫**

**深度优先遍历**

**广度优先遍历**



1. prim算法

initializemaze1();

初始化迷宫

基于prim（）算法连接通路

Getmaze()返回迷宫，最终自动生成迷宫

形成一条曲折的单通路

1. 递归分割算法

初始化二维数组后

Division分割迷宫

Connect方法连接通路

最终形成多路径通路



1. 文件操作手动生成迷宫

迷宫文件有三种

Manuel1,25\*25

Manuel2,51\*51

Manuel3,101\*101



1. 深度优先遍历

遍历每一条可到达终点的路

以列表来装每个点

遍历所有路

1. 广度优先遍历

遍历每一条可到达终点的路点

最终可形成一条最短的路径

## 1.2 算法说明

1. prim算法生成迷宫

首先有一二维数组，边长为奇数

把x和y均为奇数的点变成路点

以1，1为起始点

若周围有未联通的路点

则随机打通

再已另外一个点为起始点

一直到没有空白路点为止



1. 递归分割算法生成迷宫

首先初始化二维数组

周围一圈均是墙

任意偶数位置横竖把路变成墙

分块后在每个块之间奇数位置打出一个通路点

每次随机打四个点

之后在往四个分块切割

直到边长为1为止



1. dfs深度优先遍历迷宫

传迷宫二维数组，设置起点与终点

遍历每个点

通路则往下走

死路则返回

直到找到终点



1. bfs广度优先遍历迷宫

多条路同时走

先到达终点的直接返回

故能快速高效找到最短路径



## 1.3 功能设计

## 1.总体来说迷宫分为生成迷宫和寻路

生成迷宫又分为两种算法

Prim算法和递归分割算法

手动生成迷宫有三个文件

以不同难度来读取

寻路算法又分为两个模块

深度优先遍历和广度优先遍历

2自动生成单路径迷宫

其底层原理是prim算法

由于该算法在实现时随机性比较强

故会生成一个非常曲折的唯一通路

3自动生成多路径迷宫

其底层原理时递归分割算法

在连接通路时

把三条路变成四条路

故能实现多条通路到达终点

两个点之前通路并不唯一

4深度优先遍历寻路

遍历多路径

每条路都去遍历

但有些路会重复遍历

故效率上会低于广度优先遍历

用列表来存每条可到达的通路

比较每条通路的大小

最终寻找出最短路径

## 1.3.1 系统静态模型

createMaze类

|  |
| --- |
| createMaze |
| +Maze: int  +maze1：int[][]  +maze2;：int[][]  -*road* = 1;  -*wall* = 0;  +side：int  +option :int |
| +getMaze() : int[][]  +initializeMaze1(int side,int start\_x,int start\_y): void  +prim(int x,int y) : void  + IsHaveNeighbor(**int x,int y**) : Boolean  + save() : void  +read() : void  + readfile(int side) : int[][]  + createMaze2() : void  + connect(int start\_x,int start\_Y,int endx,int endy) : void  + division(int x,int y,int height,int width) : void |

Point类

|  |
| --- |
| Point |
| +val: int  - *x : int*  - y : int  +state : Boolean[] |
| +getx() : int  +getY() : int |

Dfsfindpath类

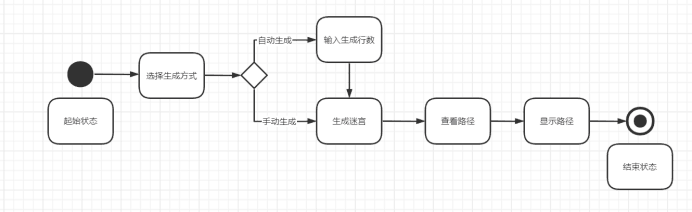
|  |
| --- |
| Dfsfindpath |
| res : ArrayList<ArrayList<Point>>  line : ArrayList<Point>  -maze : int[][]  +maze : int[][]  +side : int  +v : int[][]  -startX : int  -startY : int  -endX : int  -endY : int |
| + dfs(int x,int y) : void  +getMaze() : int[][]  +Printmin() : void |

Bfsfindpath类

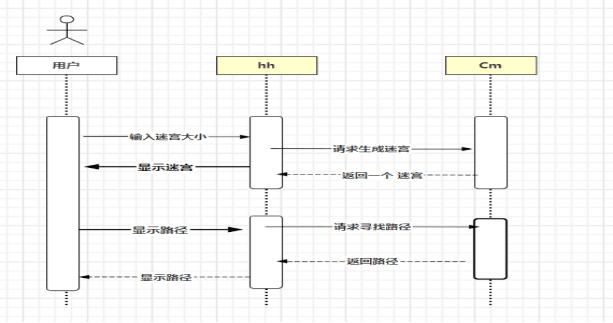
|  |
| --- |
| bfsfindpath |
| -maze : Point[]  -queue : Queue<Point>  -side: int  +map : int[][]  -pathrecord : Point[][]  -startX : int  -startY : int  -endX : int  -endY : int |
| -PointState() : void  +bfs() : void  +find() : void |

1.3.2 **系**统动态模型

**状态图**



**时序图**



1.4 系统测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 1.1 | | 版本号 | | 1.1 |
| 测试环境 | Win10 64位，eclipse | | | | |
| 用例名称 | 测试prim算法生成的迷宫 | | | | |
| 前提条件 | 大小为21 | | | | |
| 测试步骤 | 输入21 | | | | |
| 输入数据 | 21 | | | | |
| 预期输出 |  | | | | |
| 实际输出 |  | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 林吉 | 设计日期 | | 2022.6.17 | |
| 测试人 | 林吉 | 测试日期 | | | 2022.6.17 |
| 再测试人 | 林吉 | 再测试日期 | | | 2022.6.17 |
| 问题修改摘要 |  | | | | |
| 修改人 | 林吉 | | 修改日期 | | 2022.6.17 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 1.2 | | 版本号 | | 1.1 |
| 测试环境 | Win10 64位，eclipse | | | | |
| 用例名称 | 测试深度优先遍历算法生成的迷宫 | | | | |
| 前提条件 | 大小为21，起点为1，1终点为19，19 | | | | |
| 测试步骤 | 输入21，1，1，19，19 | | | | |
| 输入数据 | 输入21，1，1，19，19 | | | | |
| 预期输出 |  | | | | |
| 实际输出 |  | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 林吉 | 设计日期 | | 2022.6.17 | |
| 测试人 | 林吉 | 测试日期 | | | 2022.6.17 |
| 再测试人 | 林吉 | 再测试日期 | | | 2022.6.17 |
| 问题修改摘要 |  | | | | |
| 修改人 | 林吉 | | 修改日期 | | 2022.6.17 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 1.3 | | 版本号 | | 1.1 |
| 测试环境 | Win10 64位，eclipse | | | | |
| 用例名称 | 测试深度优先遍历算法生成的迷宫 | | | | |
| 前提条件 | 大小为31，起点为1，1终点为29，29 | | | | |
| 测试步骤 | 输入31，1，1，19，29 | | | | |
| 输入数据 | 输入31，1，1，29，29 | | | | |
| 预期输出 |  | | | | |
| 实际输出 |  | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 林吉 | 设计日期 | | 2022.6.17 | |
| 测试人 | 林吉 | 测试日期 | | | 2022.6.17 |
| 再测试人 | 林吉 | 再测试日期 | | | 2022.6.17 |
| 问题修改摘要 |  | | | | |
| 修改人 | 林吉 | | 修改日期 | | 2022.6.17 |

1.5 课程设计总结

这次为期两周的课程设计终于接近尾声，在同学和老师的帮助我下完成了这次的课程设计的题目，而我也在这次的课程设计中获益匪浅。

我选的题目时电脑迷宫鼠，其核心算法不仅难，而且有好几个，如深度优先遍历，广度优先遍历，所以我光看算法就看了好几天，但经过几天的学习，我已经能较为熟练地贯彻面向对象思想并把它使用到实践中

面向对象的思想是编程界的基本指导思想之一，它能够让程序设计人员采用一种接近真实世界的方式来设计程序，而不仅仅是考虑需要处理的数据和功能实现。这样我们就可以将程序中所需操作的对象抽象成类，并按照我们的需要来为它设置属性和方法，程序中的数据交换可以抽象为对象之间的交流，这样的思想隐藏了大量的细节，使得我们可以迅速的构建出程序的框架，实现快速开发。基于对象的程序扩展性也很强，具有十分优秀的可维护性。

最后，虽然程序已经完成，符合预期的要求，但还远远不能满足，我的程序在许多方面还有值得改进的地方，bug会在意想不到的时候突然出来，扩展性不好，不能很快的被人读懂，性能方面也不够高效，本来这个题目我们还可以增广出更多有用的功能，但是由于时间所迫，我们没有编写，存在以上种种不足，还需我们继续学习弥补