

浙江大学

程 序 设 计 专 题

大 程 序 报 告



大程名称： 基于 LibGraphics 图形库的简单迷宫游戏

小组成员：

姓名： 马琦 学号： 3210104896 电话： 18090654381

指导老师： 王建江

2022~2023 春夏学期 2023 年 6 月 20 日

目 录

| | | |
|----------|------------------------|-----------|
| 1 | 大程序简介 | 3 |
| 1.1 | 选题背景及意义 | 3 |
| 1.2 | 目标要求 | 3 |
| 1.3 | 术语说明 | 3 |
| 2 | 需求分析 | 4 |
| 2.1 | 功能需求 | 4 |
| 2.2 | 数据需求 | 4 |
| 2.4 | 性能需求 | 4 |
| 3 | 程序开发设计..... | 4 |
| 3.1 | 总体架构设计 | 5 |
| 3.2 | 功能模块设计 | 6 |
| 3.3 | 数据结构设计 | 7 |
| 3.4 | 函数设计描述 | 9 |
| 3.5 | 源代码文件组织设计 | 14 |
| 4 | 部署运行和使用说明 | 15 |
| 4.1 | 编译安装 | 15 |
| 4.2 | 运行测试 | 15 |
| 4.3 | 使用操作 | 16 |
| 5 | 团队合作 | 20 |
| 5.1 | 任务分工 | 20 |
| 5.2 | 开发计划 | 20 |
| 5.3 | 编码规范 | 20 |
| 5.4 | 合作总结 | 20 |
| 5.5 | 收获感言 | 20 |
| 6 | 参考文献资料..... | 20 |

基于 libgraphics 图形库的简单迷宫游戏大程序设计项目

1 大程序简介

1.1 选题背景及意义

迷宫是一种具有迷路和难以找到出口的结构,通常由墙壁、走廊和死路组成。其目的是为了测试人们的方向感和解谜能力,并提供一种有趣的挑战和娱乐方式。迷宫通常有不同的形状和大小,从简单的几何形状如正方形、圆形、三角形,到更复杂的形式如迷宫城堡、星形迷宫等。迷宫可以在室内或室外建造,使用各种材料和技术。当然在电脑上呈现迷宫也是非常的便捷与方便。

1.2 目标要求

设计一个迷宫游戏的地图编辑器, 并实现游戏的手动和程序求解。

1. 基本功能要求:

(1) 地图生成: 地图中至少需包含玩家、障碍物、起点、终点等元素。支持交互式手动编辑地图和自动随机生成地图。

(2) 迷宫求解: 手动求解, 使用键盘方向键走迷宫, 到达终点即胜利; 程序自动求解, 可视化显示最短路线、所有可行路线等, 支持单步执行、自动执行等。

(3) 支持菜单和工具栏(快捷键) 功能:

- ①文件:新建地图、打开地图、保存地图、退出等,
- ②地图编辑: 随机生成、手动编辑、元素选择等
- ③地图求解: 手动求解、程序求解、展示所有可行解等
- ④帮助: 使用说明

2. 较高功能要求:

- (1) 手动求解的提示功能: 提示下一步可走的格子
- (2) 程序求解可视化: 可视化最短路径

1.3 术语说明

键盘数字区方向键: 即键盘右下角上下左右键。

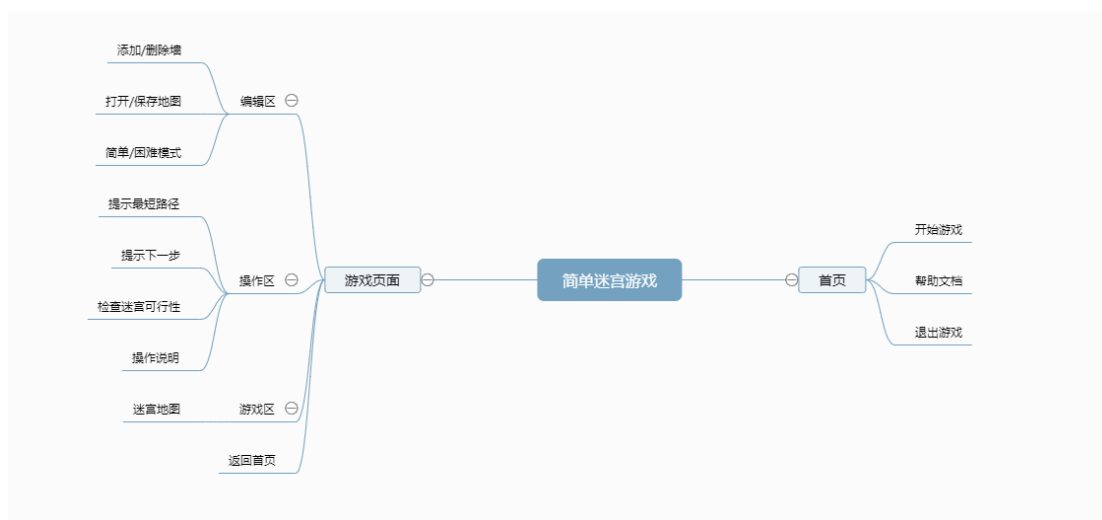
起点: 迷宫起点

终点: 迷宫终点

墙: 迷宫不可通行的部分

2 需求分析

2.1 功能需求



本程序是一个简单的迷宫游戏，其中主要实现的是游戏首页和游戏页面。

1. 首页

首页主要包含三个按钮开始、帮助和退出，点击开始按钮进入游戏页面开始游戏，点击帮助弹出帮助文档，点击退出则关闭游戏界面。

2. 游戏页面

游戏页面主要包含游戏区和操作区，玩家通过方向键控制游戏区的旗帜移动到到达终点则完成游戏。同时玩家可在操作区完成编辑迷宫、获得提示、返回首页等系列操作。

2.2 数据需求

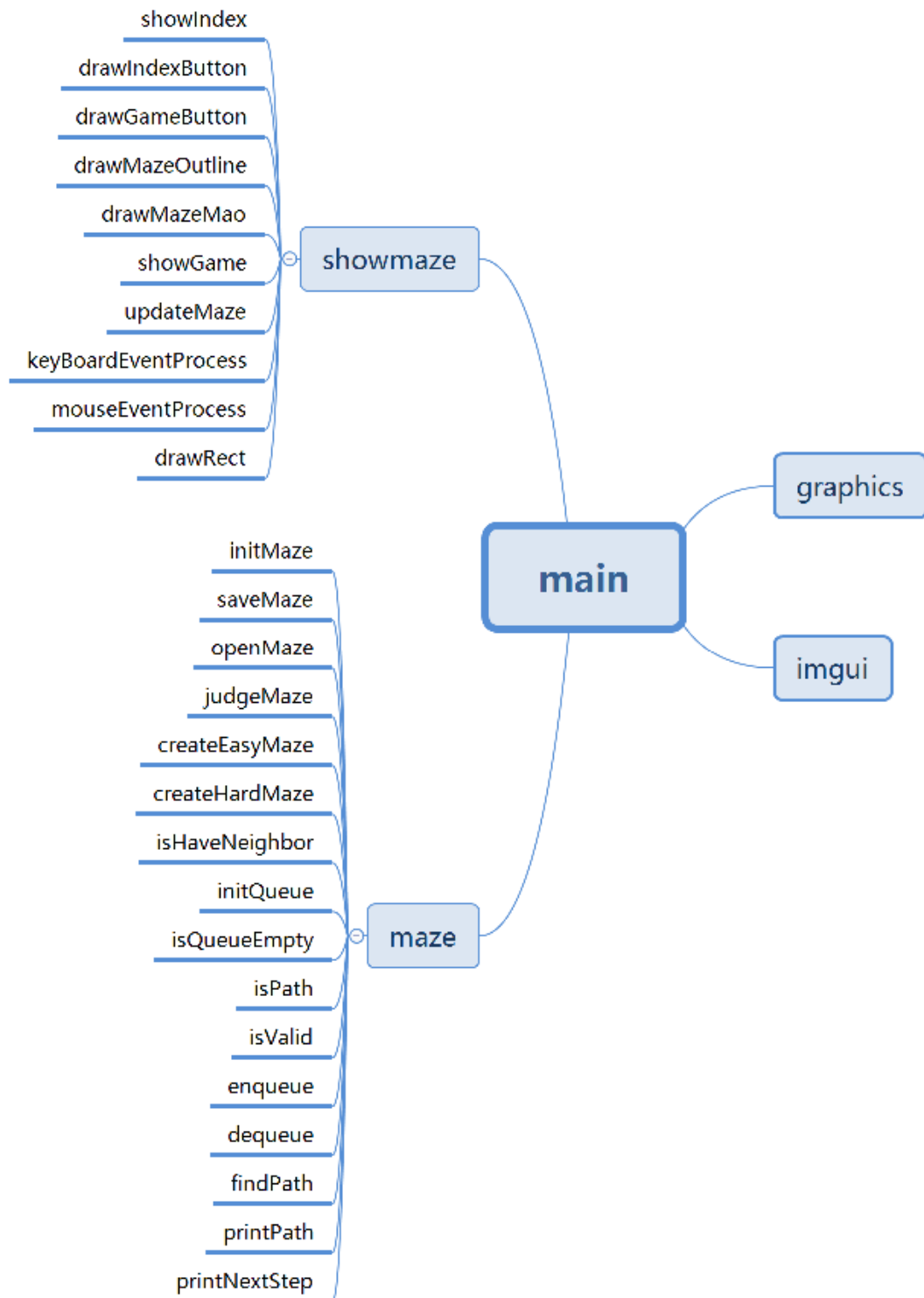
1. 本程序支持从文件中导入迷宫数据和迷宫数据的保存到文件。
2. 迷宫相关的起点、终点、路径、墙等通过二维数组进行保存。

2.3 性能需求

1. 因为有将二维的迷宫绘制到屏幕上的需求，因此有至少 $O(n^2)$ 的时间复杂度（ n 为迷宫的长和宽均为 15）

3 程序开发设计

3.1 总体架构设计



上图介绍了该程序的功能模块架构，主要包括绘图模块 graphics 和 imgui，迷宫生成和迷宫求解模块 maze，迷宫展示模块 showmaze。

3.2 功能模块设计

3.2.1 迷宫生成和求解模块 maze

该模块主要实现迷宫的生成，迷宫求解算法

```
1. #include "src/libgraphics/boolean.h"
2. #ifndef _H_MAZE
3. #define _H_MAZE
4. void createHardMaze(int xCur, int yCur);
5. void createEasyMaze(int xCur, int yCur);
6. int isHaveNeighbor(int xCur, int yCur);
7. void initMaze();
8. int judgeMaze();
9. void initQueue(struct Queue* queue);
10. bool isEmptyQueue(struct Queue* queue);
11. void enqueue(struct Queue* queue, int x, int y);
12. struct Node* dequeue(struct Queue* queue);
13. bool isPath(int x, int y);
14. bool isValid(int x, int y);
15. int findPath(int pathStartX, int pathStartY, int pathEndX, int pathEndY);
16. void printPath(int pathStartX, int pathStartY, int pathEndX, int pathEndY);
17. void printNextStep(int pathStartX, int pathStartY, int pathEndX, int pathEndY);
18. #endif
```

3.2.2 迷宫绘制模块 showmaze

该模块借助 graphics 库实现了迷宫的绘制

```
1. #ifndef _H_SHOWMAZE
2. #define _H_SHOWMAZE
3.
4. void DisplayClear();
5. void (*Show)(void);
6. void showIndex(void);
7. void drawIndexButton();
8. void drawGameButton();
9. void drawMazeOutline();
10. void drawMazeMap();
11. void showGame(void);
12. void updateMaze();
```

```
13. void keyboardEventProcess(int key, int event);
14. void mouseEventProcess(int x, int y, int button, int event);
15. void drawRect(double x, double y, double w, double h);
16. #endif
```

3.3 数据结构设计

3.3.1 parameter.h

```
1. #ifndef _H_PARAMETER
2. #define _H_PARAMETER
3. // 窗口大小
4. #define WindowWidth 12
5. #define WindowHeight 7
6.
7.
8. /**迷宫参数**/
9. #define mazeCountX 13
10. #define mazeCountY 13
11. int mazeMap[mazeCountX][mazeCountY];
12. int path[mazeCountX][mazeCountY]; // 存放路径
13. static int mazeLevel = 0; // 迷宫难度
14. // 0 - 简单模式
15. // 1 - 困难模式
16.
17. static int autoSolve = 0; // 提示类型
18. // 0 - 无提示
19. // 1 - 提示最短路径
20. // 2 - 提示下一步
21.
22.
23. // 首页相关参数
24. #define indexButtonWidth 1.5
25. #define indexButtonHeight 0.5
26. #define indexStartButtonX 5.0
27. #define indexStartButtonY 5.8
28. #define indexHelpButtonX 5.0
29. #define indexHelpButtonY 4.3
30. #define indexExitButtonX 5
31. #define indexExitButtonY 2.8
32. #define indexIconX 4.0
33. #define indexIconY 0.2
34.
```

```
35.
36. /***游戏界面参数***/
37. //  迷宫相关
38. #define mazeX 1.0
39. #define mazeY 6.5
40. #define mazeOutlineHeight 5.85
41. #define mazeOutlineWidth 5.85
42. #define mazeSize 0.45
43.
44.
45. //  操作区相关
46. #define gameAreaX 8
47. #define gameAreaY1 6
48. #define gameAreaY2 4.0
49. #define gameAreaY3 1.5
50. #define gameAreaWidth 2.0
51. #define gameAreaHeight1 0.5
52. #define gameAreaHeight2 5.0
53. #define gameButtonWidth 2.0
54. #define gameButtonHeight 0.5
55.
56.
57. /***位置参数***/
58. //  起点
59. int startX;
60. int startY;
61. //  当前位置
62. int currentX;
63. int currentY;
64. //  终点
65. int exitX;
66. int exitY;
67.
68. //  其他参数
69. int gameOver;    //  记录游戏结束与否, 1 则结束
70. int editBlock;   //  为 1 则生成墙块
71. int pageFlag;    //  1 为首页, 2 为游戏界面
72. #define CircleSize 0.05 //  弧线半径
73. //  节点
74. struct Node
75. {
76.     int x;
77.     int y;
78.     struct Node* next;
```



```
79. };  
80. // 队列  
81. struct Queue  
82. {  
83.     struct Node* front;  
84.     struct Node* rear;  
85. };  
86. #endif
```

3.4 函数设计描述

`void createHardMaze(int xCur, int yCur)`

功能描述：生成困难迷宫数组

参数描述：迷宫起始位置的横坐标、纵坐标

返回值描述：无返回值

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：仅业务代码

与 3.2 模块的对应关系：maze.h

`void createEasyMaze(int xCur, int yCur);`

功能描述：生成简单迷宫数组

参数描述：迷宫起始位置的横坐标、纵坐标

返回值描述：无返回值

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：仅业务代码

与 3.2 模块的对应关系：maze.h

`int isHaveNeighbor(int xCur, int yCur);`

功能描述：判断墙周围是否还有相邻的墙

参数描述：当前位置的横坐标、纵坐标

返回值描述：1 为有，0 为无

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：仅业务代码

与 3.2 模块的对应关系：maze.h

`void initMaze();`

功能描述：初始化迷宫数组

参数描述：无

返回值描述：无返回值

重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

```
int judgeMaze();
```

功能描述：判断迷宫是否可行
 参数描述：无
 返回值描述：1 迷宫可行，0 迷宫不可行
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

```
void initQueue(struct Queue* queue);
```

功能描述：初始化队列
 参数描述：队列指针
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

```
bool isEmpty(struct Queue* queue);
```

功能描述：判断队列是否为空
 参数描述：队列指针
 返回值描述：1 空，0 非空
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

```
void enqueue(struct Queue* queue, int x, int y);
```

功能描述：入列
 参数描述：队列指针，当前道路位置坐标
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

```
struct Node* dequeue(struct Queue* queue);
```

功能描述：出列
 参数描述：队列指针

返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

`bool isPath(int x, int y);`

功能描述：判断当前道路是否是路径
 参数描述：当前位置的横坐标、纵坐标
 返回值描述：1 代表是路径，0 代表是墙
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

`bool isValid(int x, int y);`

功能描述：判断当前位置是否有效
 参数描述：当前的横坐标、纵坐标
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

`int findPath(int pathStartX, int pathStartY, int pathEndX, int pathEndY);`

功能描述：找到当前位置到目标点的路径
 参数描述：路径的起始位置的横纵坐标
 返回值描述：1 为通路，0 为非通路
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

`void printPath(int pathStartX, int pathStartY, int pathEndX, int pathEndY);`

功能描述：将迷宫最短路径填入迷宫数组
 参数描述：路径起始位置的横坐标、纵坐标
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

`void printNextStep(int pathStartX, int pathStartY, int pathEndX, int pathEndY);`

功能描述：将下一步的提示填进迷宫数组

参数描述：路径起始位置的横纵坐标
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：maze.h

`void showIndex(void);`

功能描述：绘制迷宫的首页
 参数描述：无
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：showmaze.h

`void drawIndexButton();`

功能描述：绘制首页的按钮
 参数描述：无
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：showmaze.h

`void drawGameButton();`

功能描述：绘制游戏界面的按钮
 参数描述：无
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：showmaze.h

`void drawMazeOutline();`

功能描述：绘制游戏区边框
 参数描述：无
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：showmaze.h

`void drawMazeMap();`

功能描述：绘制迷宫
 参数描述：无
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：showmaze.h

`void showGame(void);`

功能描述：绘制游戏区
 参数描述：无
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：showmaze.h

`void updateMaze();`

功能描述：更新迷宫
 参数描述：无
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：showmaze.h

`void keyboardEventProcess(int key, int event);`

功能描述：读取按键
 参数描述：按键及对应事件
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：showmaze.h

`void mouseEventProcess(int x, int y, int button, int event);`

功能描述：获取鼠标按键
 参数描述：按键及事件
 返回值描述：无返回值
 重要局部变量定义：无
 重要局部变量用途描述：无
 函数算法描述：仅业务代码
 与 3.2 模块的对应关系：showmaze.h

```
void drawRect(double x, double y, double w, double h);
```

功能描述：绘制矩形

参数描述：绘制位置的横纵坐标和矩形的长和宽

返回值描述：无返回值

重要局部变量定义：无

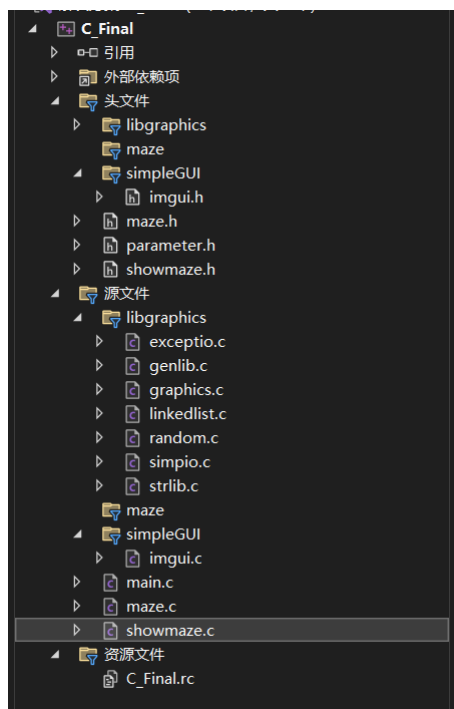
重要局部变量用途描述：无

函数算法描述：仅业务代码

与 3.2 模块的对应关系：showmaze.h

3.5 源代码文件组织设计

3.5.1 文件目录结构



3.5.2 文件函数结构

.c 文件与.h 文件如上文的结构设计和函数设计展示，故不在此赘述。

3.5.3 多文件构成机制

本项目的文件之间采用 `#include` 文件所对应的.h 文件来连接各个.c 文件。`extern` 声明绝大部分都放在.h 文件中，而真实存放位置是其对应的.c 文件；引用.h 文件就可以访问到对应的变量。

4 部署运行和使用说明

4.1 编译安装

解压所提供的压缩包，打开其中的 C_Final.sln 文件，在 Visual Studio 中编译运行即可。

| 名称 | 修改日期 | 类型 | 大小 |
|-------------------------|-----------------|-------------------------|-------|
| .vs | 2023/6/14 12:41 | 文件夹 | |
| Release | 2023/6/25 18:44 | 文件夹 | |
| src | 2023/6/14 17:57 | 文件夹 | |
| C_Final.apr | 2023/6/14 18:10 | APS 文件 | 2 KB |
| C_Final.rc | 2023/6/14 18:13 | Resource Script | 3 KB |
| C_Final.sln | 2023/6/14 12:41 | Visual Studio Soluti... | 2 KB |
| C_Final.vcxproj | 2023/6/17 23:00 | VC++ Project | 8 KB |
| C_Final.vcxproj.filters | 2023/6/17 23:00 | VC++ Project Filter... | 5 KB |
| C_Final.vcxproj.user | 2023/6/14 12:41 | Per-User Project O... | 1 KB |
| main.c | 2023/6/25 18:35 | C 文件 | 1 KB |
| maze.c | 2023/6/25 20:05 | C 文件 | 12 KB |
| maze.h | 2023/6/25 20:21 | C/C++ Header | 1 KB |
| parameter.h | 2023/6/25 18:51 | C/C++ Header | 2 KB |
| showmaze.c | 2023/6/25 20:05 | C 文件 | 13 KB |
| showmaze.h | 2023/6/25 20:07 | C/C++ Header | 1 KB |

4.2 运行测试

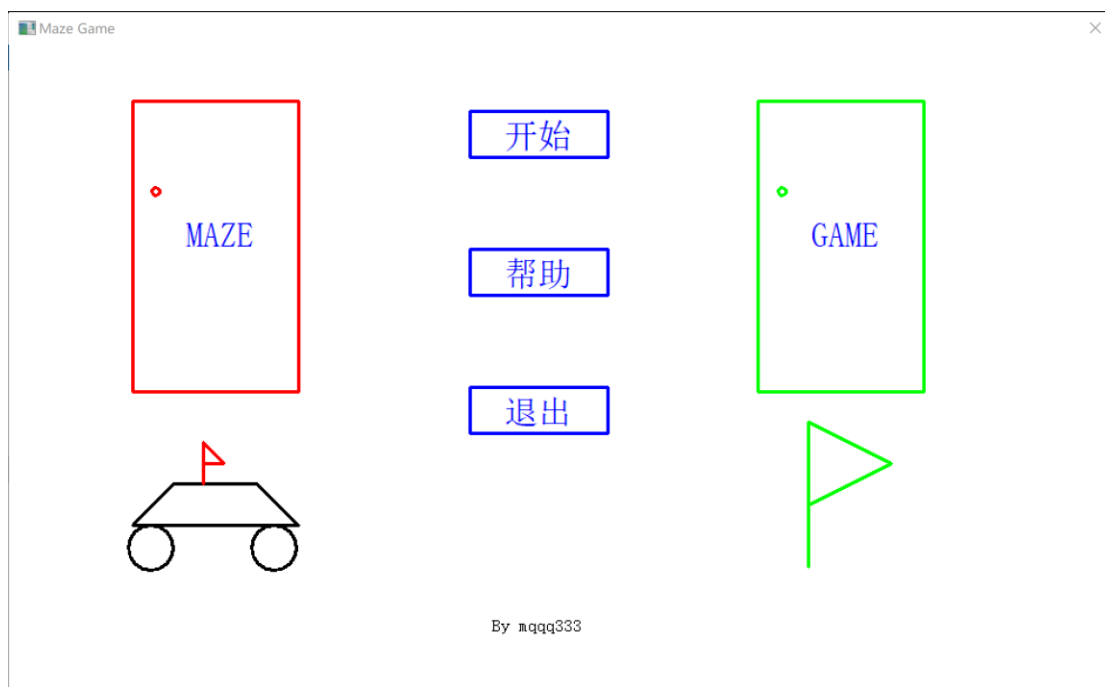
打开解压后的文件夹，点击其中的 Release

| | | | |
|-------------------------|-----------------|-------------------------|-------|
| .vs | 2023/6/14 12:41 | 文件夹 | |
| Release | 2023/6/25 18:44 | 文件夹 | |
| src | 2023/6/14 17:57 | 文件夹 | |
| C_Final.apr | 2023/6/14 18:10 | APS 文件 | 2 KB |
| C_Final.rc | 2023/6/14 18:13 | Resource Script | 3 KB |
| C_Final.sln | 2023/6/14 12:41 | Visual Studio Soluti... | 2 KB |
| C_Final.vcxproj | 2023/6/17 23:00 | VC++ Project | 8 KB |
| C_Final.vcxproj.filters | 2023/6/17 23:00 | VC++ Project Filter... | 5 KB |
| C_Final.vcxproj.user | 2023/6/14 12:41 | Per-User Project O... | 1 KB |
| main.c | 2023/6/25 18:35 | C 文件 | 1 KB |
| maze.c | 2023/6/25 20:05 | C 文件 | 12 KB |
| maze.h | 2023/6/25 20:21 | C/C++ Header | 1 KB |
| parameter.h | 2023/6/25 18:51 | C/C++ Header | 2 KB |
| showmaze.c | 2023/6/25 20:05 | C 文件 | 13 KB |
| showmaze.h | 2023/6/25 20:07 | C/C++ Header | 1 KB |

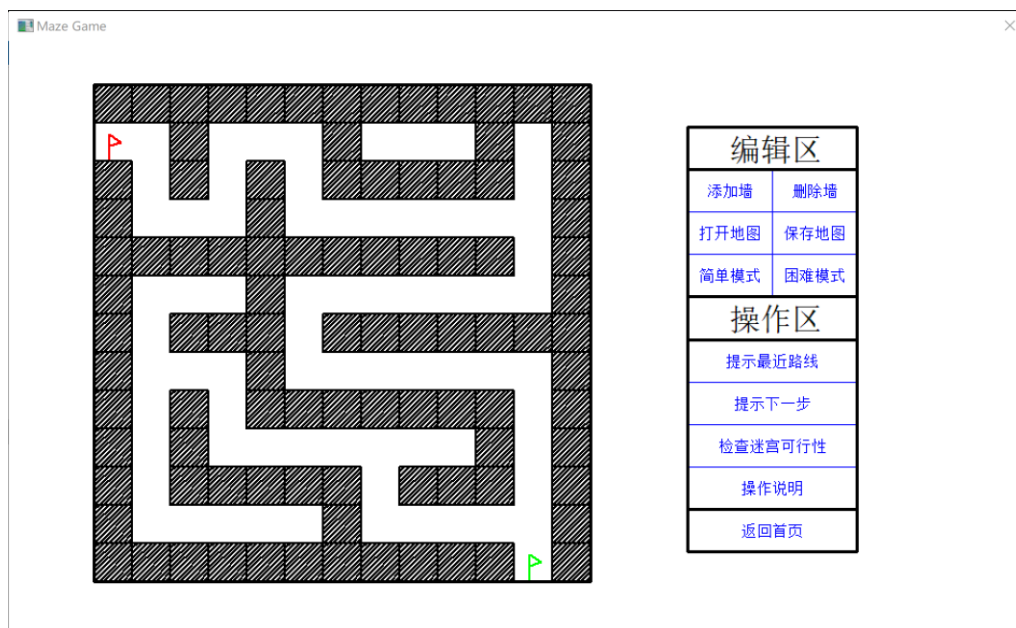
运行 Release 文件夹中的 C_Final.exe 文件进行测试

4.3 用户使用手册

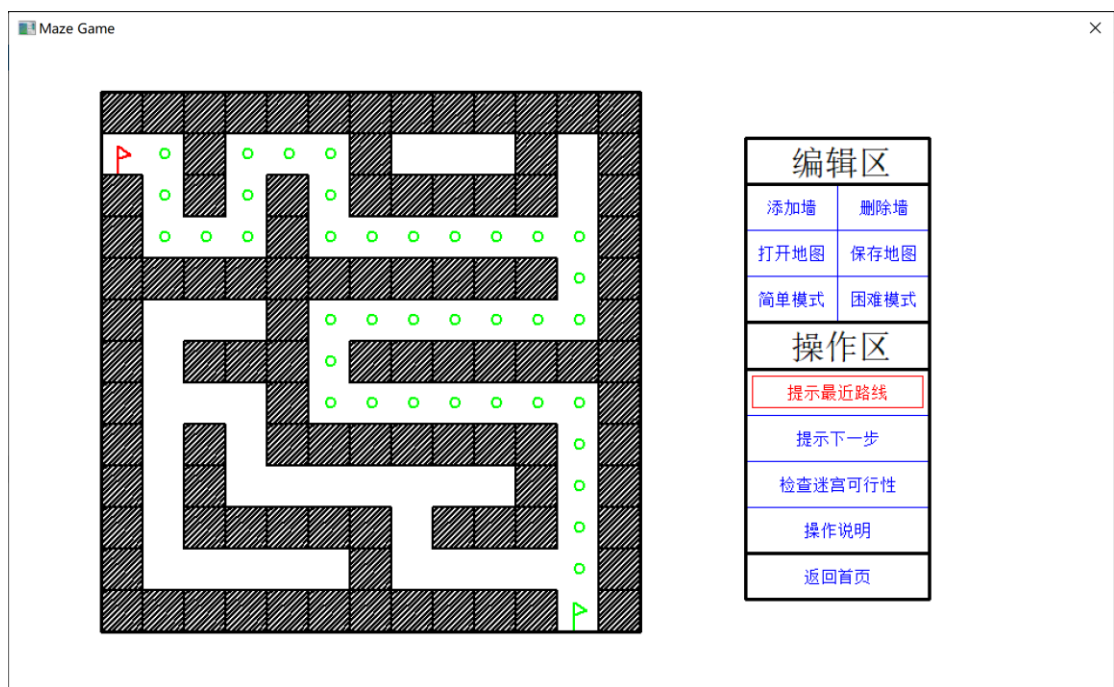
运行上述 C_Final.exe 文件后进入如下界面，点击开始即可开始游戏，点击帮助可查看帮助文档，点击退出可退出程序。



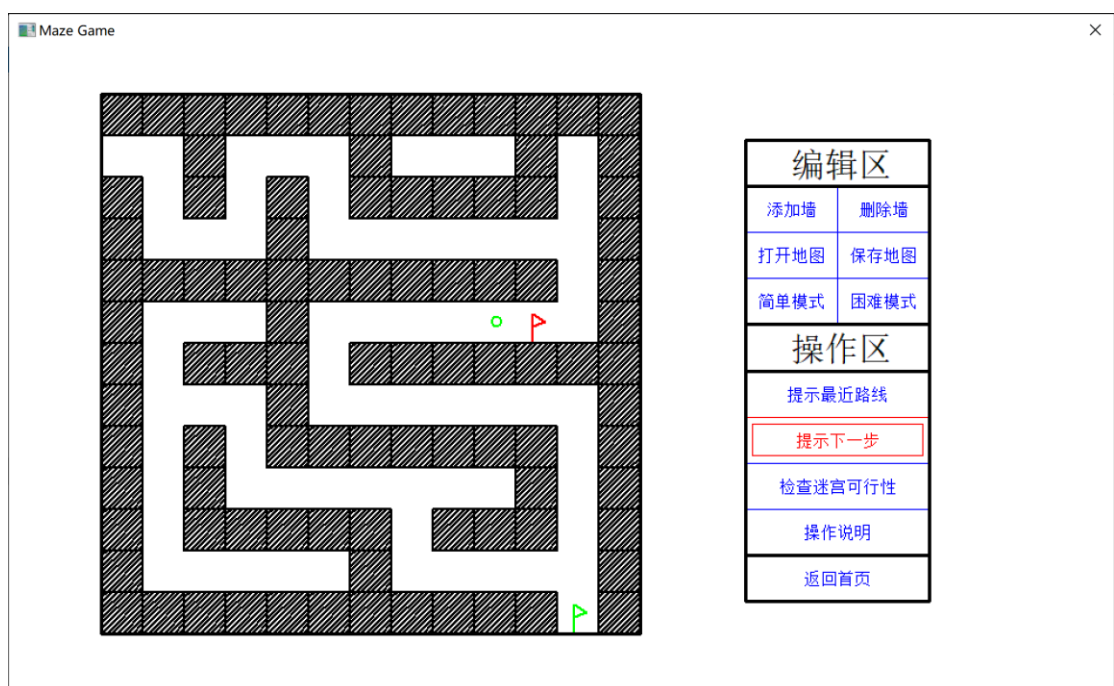
点击开始后进入游戏界面，可点击右边的添加墙和删除墙自行编辑地图，也可以打开和保存地图，可点击简单模式和困难模式生成新的地图。



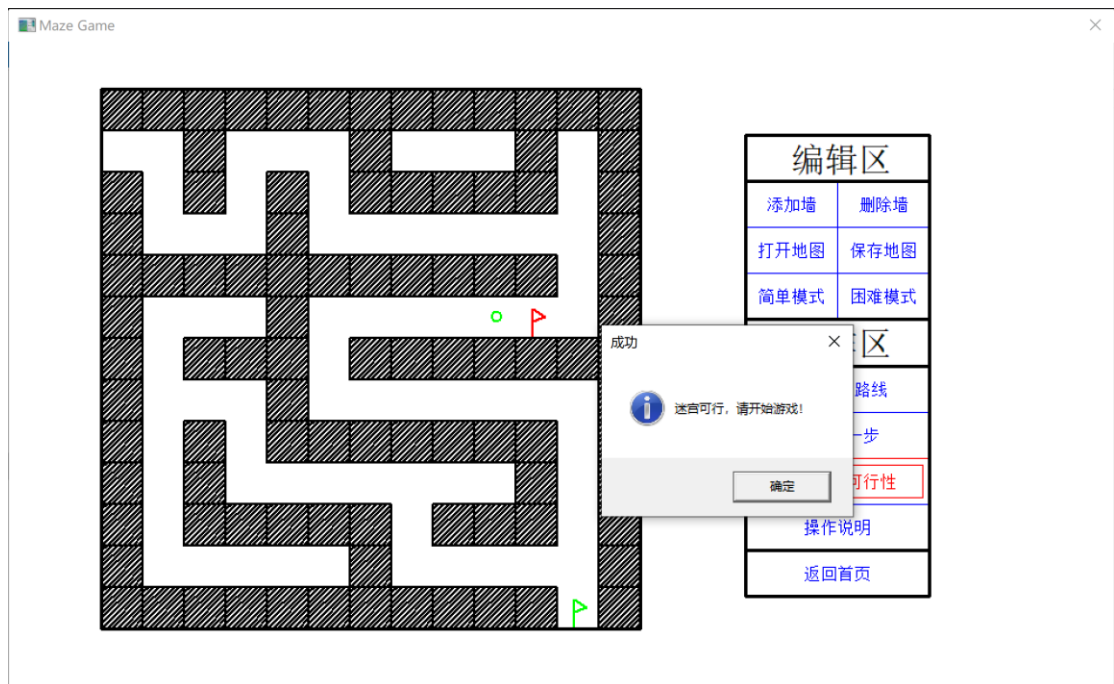
再点击提示最近路线，可提示由当前位置到终点的最近路径。



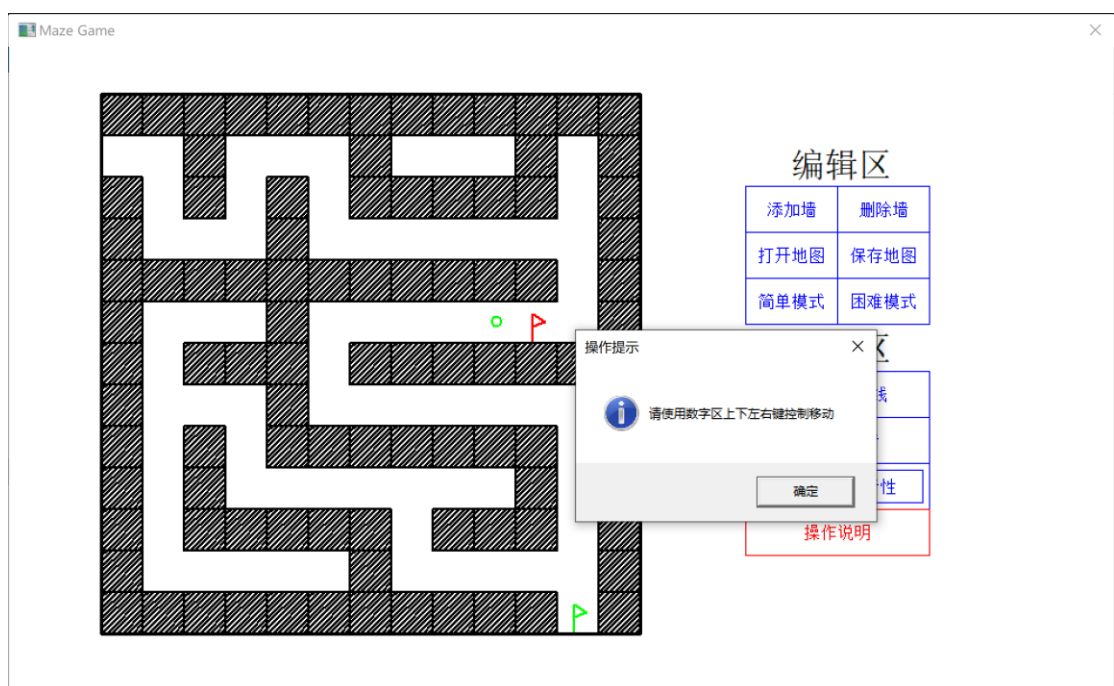
点击提示下一步，可提示下一步的位置



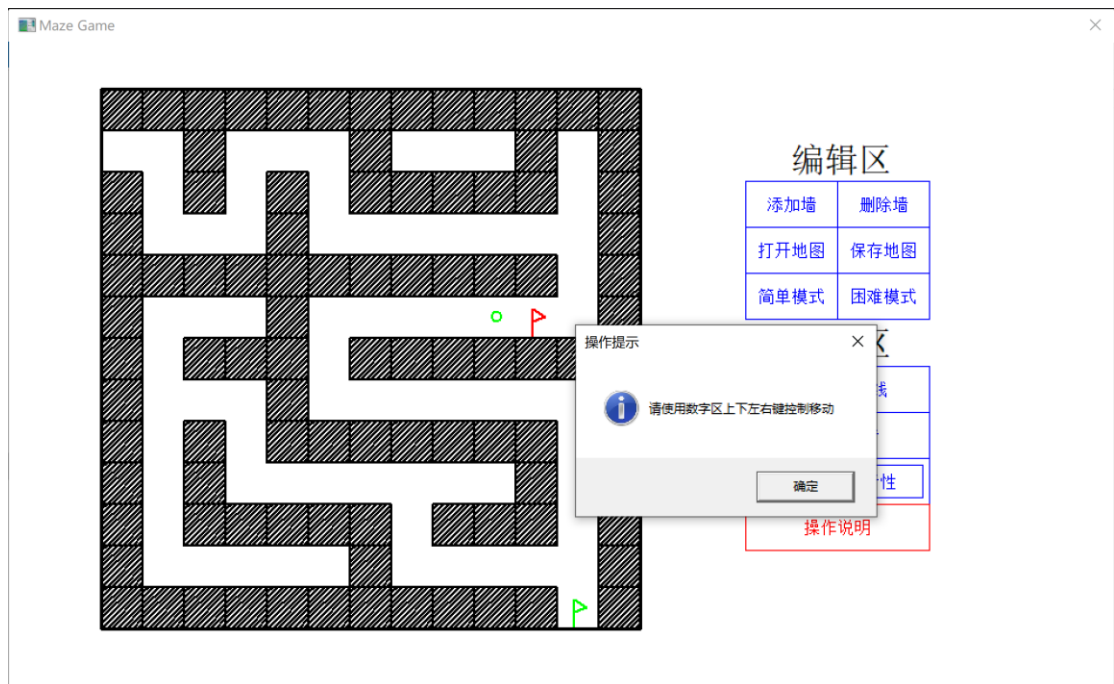
点击查看迷宫可行性可判断迷宫是否存在通路



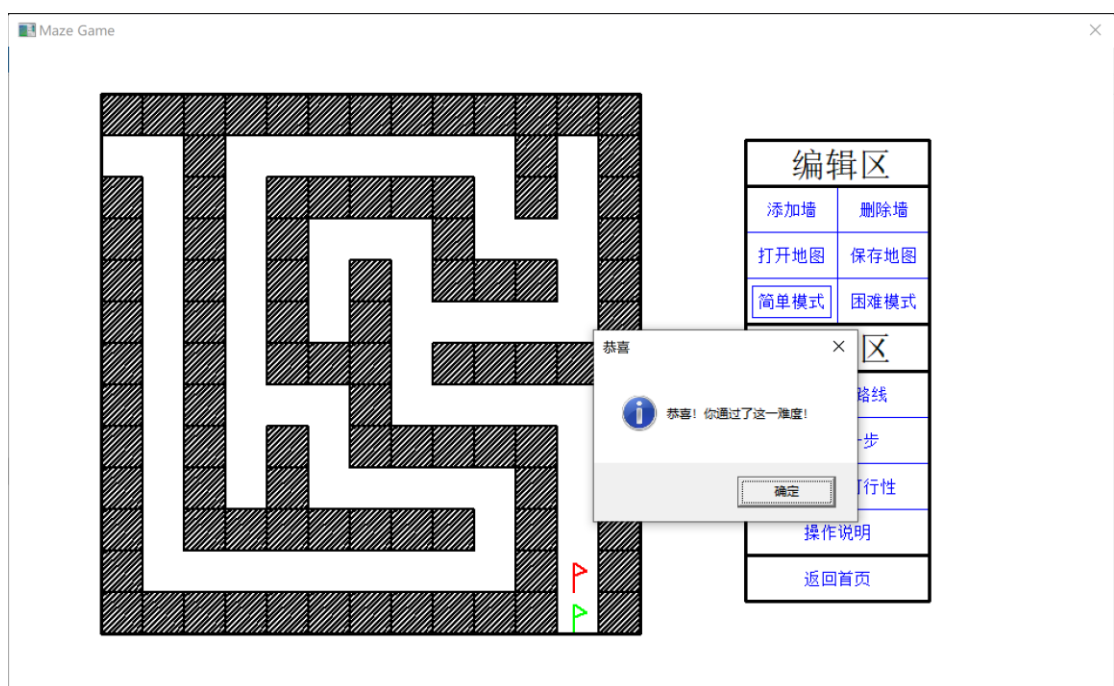
点击操作说明可提示按键说明。



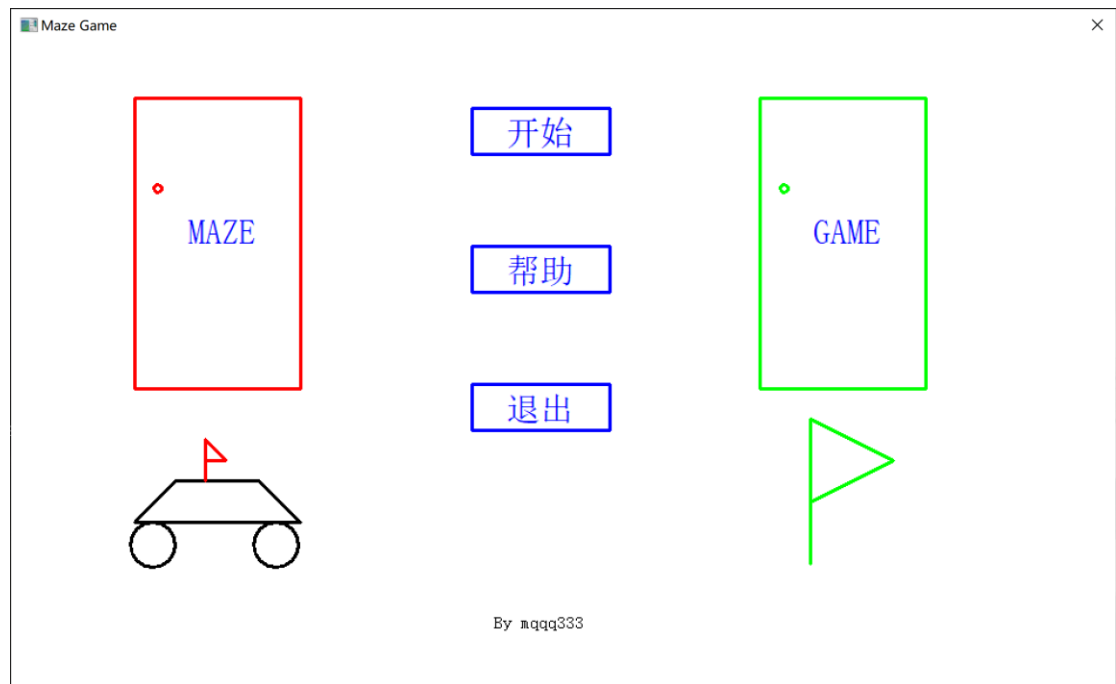
点击返回首页，可返回首页。



游戏胜利将获得成功提示。



点击返回首页将再次返回首页



5 团队合作

5.1 开发计划

- 第一步，构思整体架构，分配各个模块
- 第二步，解决迷宫的生成
- 第三步，解决 graphics、imgui 图形库的使用
- 第四步，开始编写各模块
- 第五步，后续完善与修饰

5.2 编码规范

- 1.使用 TAB 缩进
- 2.配有应有的注释
- 3.变量采用驼峰命名法
- 4.无任何括号的省略

5.3 任务分工

- 1.迷宫的生成
- 2.迷宫的绘制
- 3.迷宫的求解
- 4.界面的绘制

5.4 个人遇到的难点与解决方案

难点：迷宫生成和迷宫的求解。

解决法案：参考前人代码，理解后逐步完善形成自己的代码。

5.5 工作总结

5.5.1 开发亮点

个人实力较差，完成的比较仓促，应该没有突出的亮点。

5.5.2 开发挑战点

- 1.需要对多文件的组织和管理有一定了解
- 2.迷宫的生成算法和求解的理解

5.5.3 应用知识点

- 1.模块化程序设计
- 2.LibGraphics 的使用和 Win32 API 的使用
- 3.指针的使用
- 4.链表的使用
- 5.文件的读取与保存

5.6 收获感言

1. 在写代码前应该理清自己的思路，知道自己要做什么东西，理清各部分之间的联系便于程序的整体实现。

2.模块化程序设计：把可以单独编写的模块提出来写，减少重复代码编写，降低整个程序的结构复杂度，提高程序的可调式和可修改性。

3.规范的代码标准，让自己的代码更容易让自己和他人看懂。

这个项目是自己写的第一个较大的项目，不同于以前在命令行上运行的代码，这个项目有了界面，也有人与键盘和鼠标的交互，整个开发过程算是一段虽然艰难确实收获满满的经历。

6 参考文献资料

1. 《C Primer Plus》

2. 王建江老师课程 PPT