

---

# 项目说明文档

## 数据结构课程设计

### ——勇闯迷宫游戏

作者姓名： 安江涛

学 号： 1952560

指导教师： 张颖

学院、专业： 软件学院 软件工程

同济大学

Tongji University

---

## 目 录

1 分析.....	1
1.1 项目简介 .....	1
2 设计.....	1
2.1 数据结构设计.....	1
2.2 类结构设计.....	1
2.3 成员与操作设计.....	1
3 实现.....	3
3.1 系统初始化.....	3
3.2 寻找路径 .....	4
3.3 输出路径 .....	5
3.4 总体系统的实现.....	6
3.4.1 总体系统流程图 .....	6
3.4.2 总体系统核心代码 .....	7
3.4.3 总体系统截屏示例 .....	8

---

# 1 分析

## 1.1 项目简介

迷宫只有两个门，一个门叫入口，另一个门叫出口。一个骑士骑马从入口进入迷宫，迷宫设置很多障碍，骑士需要在迷宫中寻找通路以到达出口。

迷宫问题的求解过程可以采用回溯法即在一定的约束条件下试探地搜索前进，若前进中受阻，则及时回头纠正错误另择通路继续搜索的方法。从入口出发，按某一方向向前探索，若能走通，即某处可达，则到达新点，否则探索下一个方向；若所有的方向均没有通路，则沿原路返回前一点，换下一个方向再继续试探，直到所有可能的道路都探索到，或找到一条通路，或无路可走又返回入口点。在求解过程中，为了保证在达到某一个点后不能向前继续行走时，能正确返回前一个以便从下一个方向向前试探，则需要试探过程中保存所能够达到的每个点的下标以及该点前进的方向，当找到出口时试探过程就结束了。

# 2 设计

## 2.1 数据结构设计

显然，一般来说迷宫问题都可以用 DFS 搜索或是 BFS 搜索来实现，本题采用 DFS 搜索来实现，地图使用邻接矩阵来储存。

## 2.2 类结构设计

本项目只有一个类，迷宫类（**maze**），实现迷宫初始化、路径搜索以及输出路径等功能，其中使用到的 **vector** 是手写，包含在了头文件里。

## 2.3 成员与操作设计

迷宫类（**maze**）

---

```
1. class maze
2. {
3. public:
4.     maze() :_row(0), _col(0) {
5.         _entrancex = _entrancey = -1;
6.         _exitx = _exity = -1;
7.         _pathnum = 0;
8.     }
9.     void set();
10.    void setMaze();
11.    bool setEntrance();
12.    bool setExit();
13.    void mapPrintf();//输出迷宫地图
14.    void pathPrint();//输出路径
15.    void dfs(int x, int y);
16.    void init() {
17.        for (int i = 0; i < _row; i++) {
18.            for (int j = 0; j < _col; j++) {
19.                _done[i].push_back(false);
20.            }
21.        }
22.    }
23.
24. private:
25.     int _row, _col;//行和列,从0开始
26.     int _entrancex, _entrancey;
27.     int _exitx, _exity;
28.     std::string _maze[maxn];
29.     Vector<bool> _done[maxn];
30.     Vector<std::pair<int, int> > _path[10];
31.     Vector<std::pair<int, int> > _tpath;
32.     int _pathnum;
33.     int dir[4][2] = { {1,0},{-1,0},{0,1},{0,-1} };
34. };
```

---

## 3 实现

### 3.1 系统初始化

```
bool maze::setEntrance() {
    int row = -1, col = -1;
    std::cout << "Please enter the entrance of the maze:";
    std::cin >> row >> col;
    std::cout << '\n';
    if (row < 0 || row >= _row || col < 0 || col >= _col)
        return false;
    _entrancex = row; _entrancey = col;
    return true;
}

bool maze::setExit() {
    int row = -1, col = -1;
    std::cout << "Please enter the exit of the maze: ";
    std::cin >> row >> col;
    std::cout << '\n';
    if (row < 0 || row >= _row || col < 0 || col >= _col)
        return false;
    _exitx = row; _exity = col;
    return true;
}

void maze::setMaze() {
    if (_row == 0 || _col == 0) {
        std::cout << "Please enter the rows and columns of the maze: ";
        std::cin >> _row >> _col;
    }
    std::cout << '\n';
    std::cout << "Please enter the structure of the maze: " << '\n';
    for (int i = 0; i < _row; i++)
        for (int j = 0; j < _col; j++) {
            char ch;
            std::cin >> ch;
            _maze[i].push_back(ch);
        }
    std::cout << '\n';
}
```

---

## 3.2 寻找路径

```
1. void init() {
2.     for (int i = 0; i < _row; i++) {
3.         for (int j = 0; j < _col; j++) {
4.             _done[i].push_back(false);
5.         }
6.     }
7. }
8. void maze::dfs(int x, int y) {
9.     if (x == _exitx && y == _exity) {
10.        for (int i = 0; i < _tpath.size(); i++) {
11.            _path[_pathnum].push_back(_tpath[i]);
12.        }
13.        _pathnum++;
14.        return;
15.    }
16.    if (x >= _row || x < 0)
17.        return;
18.    if (y >= _col || y < 0)
19.        return;
20.    if (_maze[x][y] == '#')
21.        return;
22.    if (_done[x][y])
23.        return;
24.    _done[x][y] = true;
25.    for (int i = 0; i <= 3; i++) {
26.        int newx = x + dir[i][0];
27.        int newy = y + dir[i][1];
28.        _tpath.push_back(std::make_pair(newx, newy));
29.        dfs(newx, newy);
30.        _tpath.pop_back();
31.    }
32.    _done[x][y] = false;
33. }
```

---

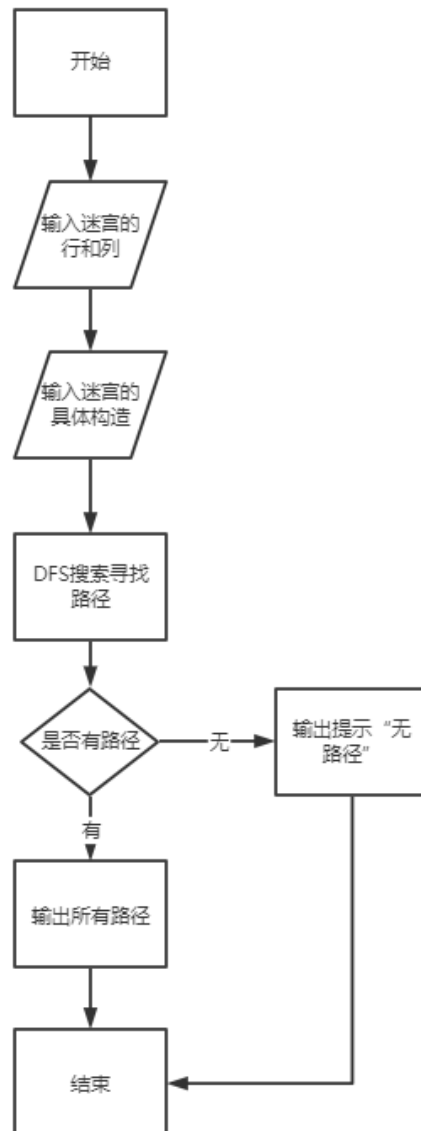
### 3.3 输出路径

```
void maze::mapPrintf() {
    std::cout << "Maze map: " << '\n' << '\n';
    std::cout << "  ";
    for (int i = 0; i < _col; i++) {
        std::cout << i << "Column" << " ";
    }
    std::cout << '\n';
    for (int i = 0; i < _row; i++) {
        std::cout << i << "Row";
        std::cout << " ";
        for (int j = 0; j <= _col; j++) {
            std::cout << _maze[i][j] << " ";
        }
        std::cout << '\n';
    }
    std::cout << '\n';
}
```

---

### 3.3 总体系统的实现

#### 3.3.1 总体系统流程图





---

### 3.3.2 总体系统核心代码

```
int main()
{
    maze mymaze;

    mymaze.set();

    mymaze.mapPrintf();

    mymaze.pathPrint();
}

void maze::set() {
    this->setMaze();

    this->setEntrance();

    this->setExit();

    this->_tpath.push_back(std::make_pair(_entrancex, _entrancey));

    this->init();

    this->dfs(_entrancex, _entrancey);
}
```

### 3.3.3 总体系统截屏示例

```
root@iZbp180dvecytxjnhuzrdnZ:~/css_design/subject# ./p3.out
Please enter the rows and columns of the maze: 7 7

Please enter the structure of the maze :
# # # # # # #
# 0 # 0 0 0 #
# 0 0 0 # # #
# 0 0 0 # 0 #
# 0 # 0 0 0 #
# 0 # 0 # 0 #
# # # # # #

Please enter the entrance of the maze:1 1

Please enter the exit of the maze: 5 5

Maze map :

    0Column  1Column  2Column  3Column  4Column  5Column  6Column
0Row #      #      #      #      #      #      #
1Row #      0      #      0      0      0      #
2Row #      0      0      0      #      #      #
3Row #      0      0      0      #      0      #
4Row #      0      #      0      0      0      #
5Row #      0      #      0      #      0      #
6Row #      #      #      #      #      #      #

Maze path :

Path 1 :
<1,1>--><2,1>--><3,1>--><3,2>--><2,2>--><2,3>--><3,3>--><4,3>--><4,4>--><4,5>--><5,5>
Path 2 :
<1,1>--><2,1>--><3,1>--><3,2>--><3,3>--><4,3>--><4,4>--><4,5>--><5,5>
Path 3 :
<1,1>--><2,1>--><2,2>--><3,2>--><3,3>--><4,3>--><4,4>--><4,5>--><5,5>
Path 4 :
<1,1>--><2,1>--><2,2>--><2,3>--><3,3>--><4,3>--><4,4>--><4,5>--><5,5>
root@iZbp180dvecytxjnhuzrdnZ:~/css_design/subject#
```

```
root@iZbp180dvecytxjnhuzrdnZ:~/css_design/subject# ./p3.out
Please enter the rows and columns of the maze: 7 7
```

```
Please enter the structure of the maze:
```

```
# # # # # # #
# 0 # 0 0 0 #
# 0 0 0 # # #
# 0 0 0 # 0 #
# 0 # 0 0 # #
# 0 # 0 # 0 #
# # # # # # #
```

```
Please enter the entrance of the maze:1 1
```

```
Please enter the exit of the maze: 5 5
```

```
Maze map:
```

	0Column	1Column	2Column	3Column	4Column	5Column	6Column
0Row #	#	#	#	#	#	#	#
1Row #	0	#	0	0	0	#	#
2Row #	0	0	0	#	#	#	#
3Row #	0	0	0	#	0	#	#
4Row #	0	#	0	0	#	#	#
5Row #	0	#	0	#	0	#	#
6Row #	#	#	#	#	#	#	#

```
No path!
```

```
root@iZbp180dvecytxjnhuzrdnZ:~/css_design/subject#
```