

1 画出地图方便测试

```
#define CHEAT
```

每次步骤都会打印出当前的地图和玩家所在的位置，方便进行测试。其中公主的标志是"P"，怪兽的标志是"M"，玩家的标志是"1"，起点（终点）的标志为"L"。且当遇到公主后，之后的玩家的标志将变为"P"。游戏中实际上对于玩家而言是不能看到地图的，把#define CHEAT注释掉，就不打印地图了。

2 地图的读取

```
fin.open("xxx.txt");
```

在工程目录下有"map1.txt" 和 "map2.txt"两个地图文件供测试使用，一个是3层3×3的地图，一个是4层4×4的地图。

```
#define MAP 1 //读取"map1.txt"地图
#define MAP 2 //读取"map2.txt"地图
#define MAP 0 //随机生成地图（层数2-5，每层房间数2^2-5^2）
```

该工程可以支持任意m层n×n的地图，具有较好的扩展性。如果需要后期自行设计更多的地图，可以参考以下的文件格式：

首先，为了方便进行读取，每行的数字之间要空格。第一行有5个数字，分别表示地图的层数，每层房间数（n×n），起点的位置，怪兽的位置，公主的位置（由于房间是存储在容器中，位置的编号从0开始，从第1层，每层从左上开始到右下进行蛇形递增，接着下一层继续）。

之后的文件n×n行按编号从小到大顺序记录每个房间的门的的信息。按门总数、北门、南门、西门、东门、上门、下门的真值顺序记录。

比如说，有一个房间总共有2个门，分别是北门和上门，则其在文件中的记录为：2（门数） 1（北门） 0（南门） 0（西门） 0（东门） 1（上门） 0（下门）。

以下表格能够更加清晰地进行说明：

门类	文件记录
无门，怪兽位置	0 0 0 0 0 0 0
满门，6个门	1 1 1 1 1 1 1
一般情况，南门、东门、上门	3 0 1 0 1 1 0
一般情况，上门、下门	2 0 0 0 0 1 1

3 用户界面的颜色

实验为用户界面设计了两种前景色、背景色的搭配。使用下面语句：

```
#define COLOR
/*
用户界面的配色为"color f4";
如果想要默认的用户界面，注释掉该宏定义语句。
*/
```

4 玩家输入

- 开始时，按任意按钮进入游戏

```
Welcome to the Adventure!
Press Any Key to Begin!
```

- 之后移动的时候输入相应的方向键

```
"w":走北
"s":走南
"a":走西
"d":走东
"u":走上
"n":走下
/*
移动过程中，任何时候想要退出当前游戏，输入"q"即可。
*/
```

- 游戏结束

每次游戏结束以后，输入"y"或者"no"决定是否重新开始游戏。