**勇闯迷宫项目文档**

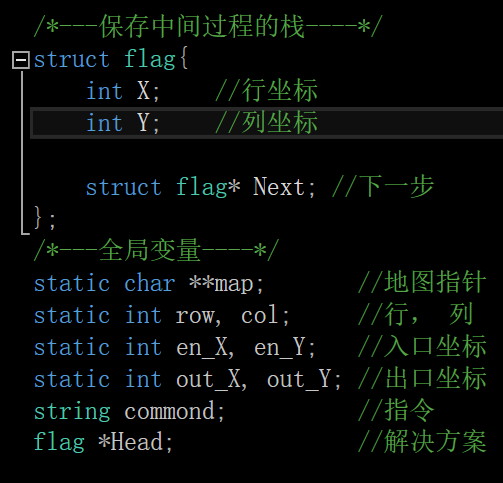
1. 项目简介：

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。

1. 项目功能要求:

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，则需要在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。

1. 具体实现：
2. 数据结构：



1.struct flag:{int X, Y; struct flag\* Next} //保存中间过程的栈

2.static char \*\*map //地图指针

3.static int row, col //行，列

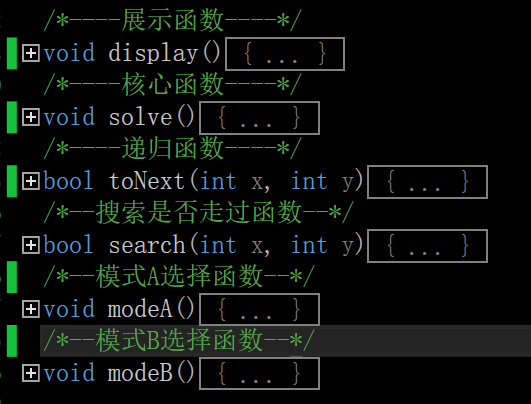
4.static int en\_X, en\_Y //入口坐标

5.static int out\_X, out\_Y //出口坐标

6.string commond //指令

7.flag \*Head //解决方案头指针

2.函数：



1.void display() //展示可行路径

2.void solve() //核心函数

3.bool toNext(int x, int y) //判断下点可行性

4.bool search(int x, int y) //规避重复路径

5.void modeA() //模式A选择函数

6.void modeB() //模式B选择函数

4．算法分析：

基本思想为深度搜索，由上、下、左、右的顺序进行搜索，对任一可行节点进行递归搜索，每一步保存路径。在递归失败之后层层退栈，继续尝试下一可行节点，直到找到出口。然后将路径栈逐步出栈即可找到可行路径。

1. 游戏特色：

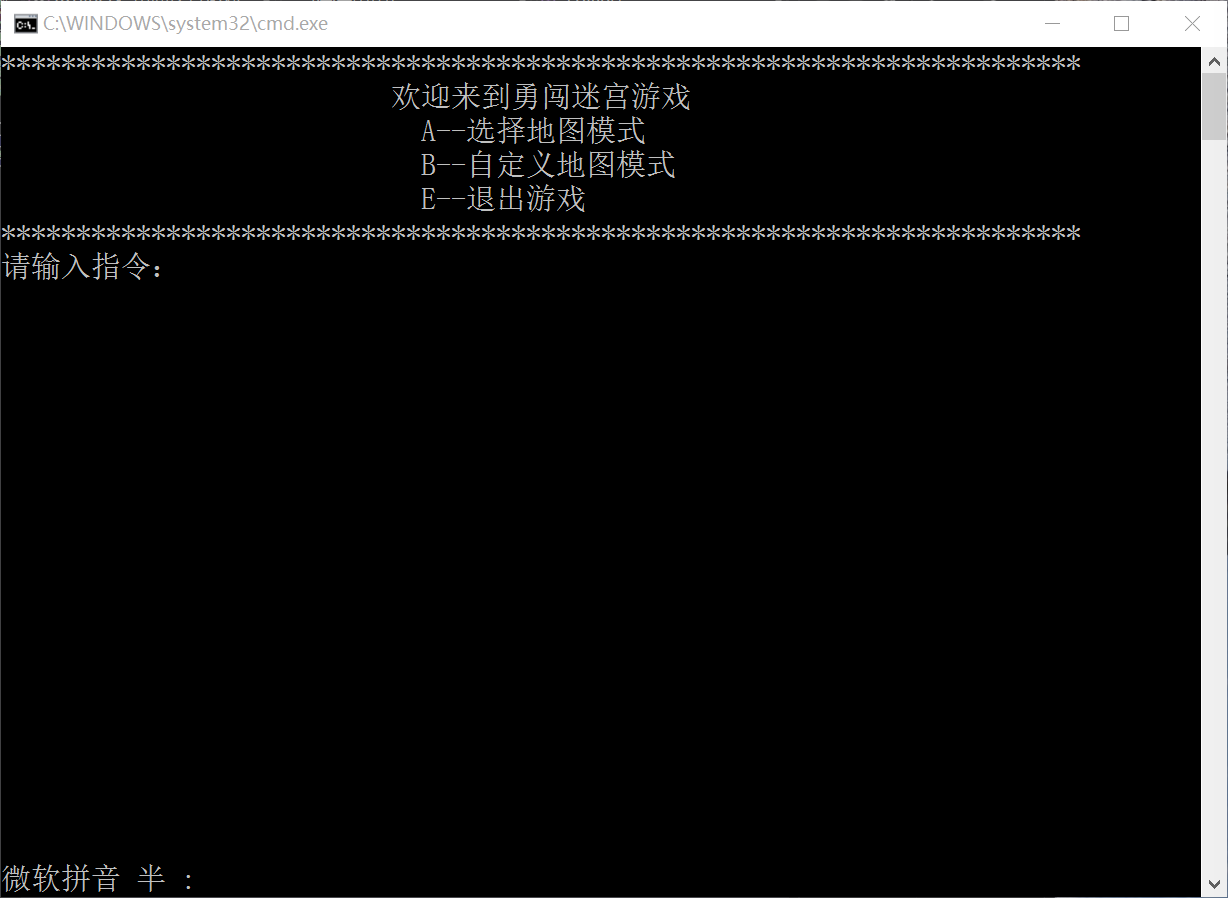
游戏提供了两种游戏模式

ModeA:由玩家选择所提供的地图进行游戏。其中地图写在文件中，由程序读取文件。文件写死，如果找不到文件，将无法使用这个模式。

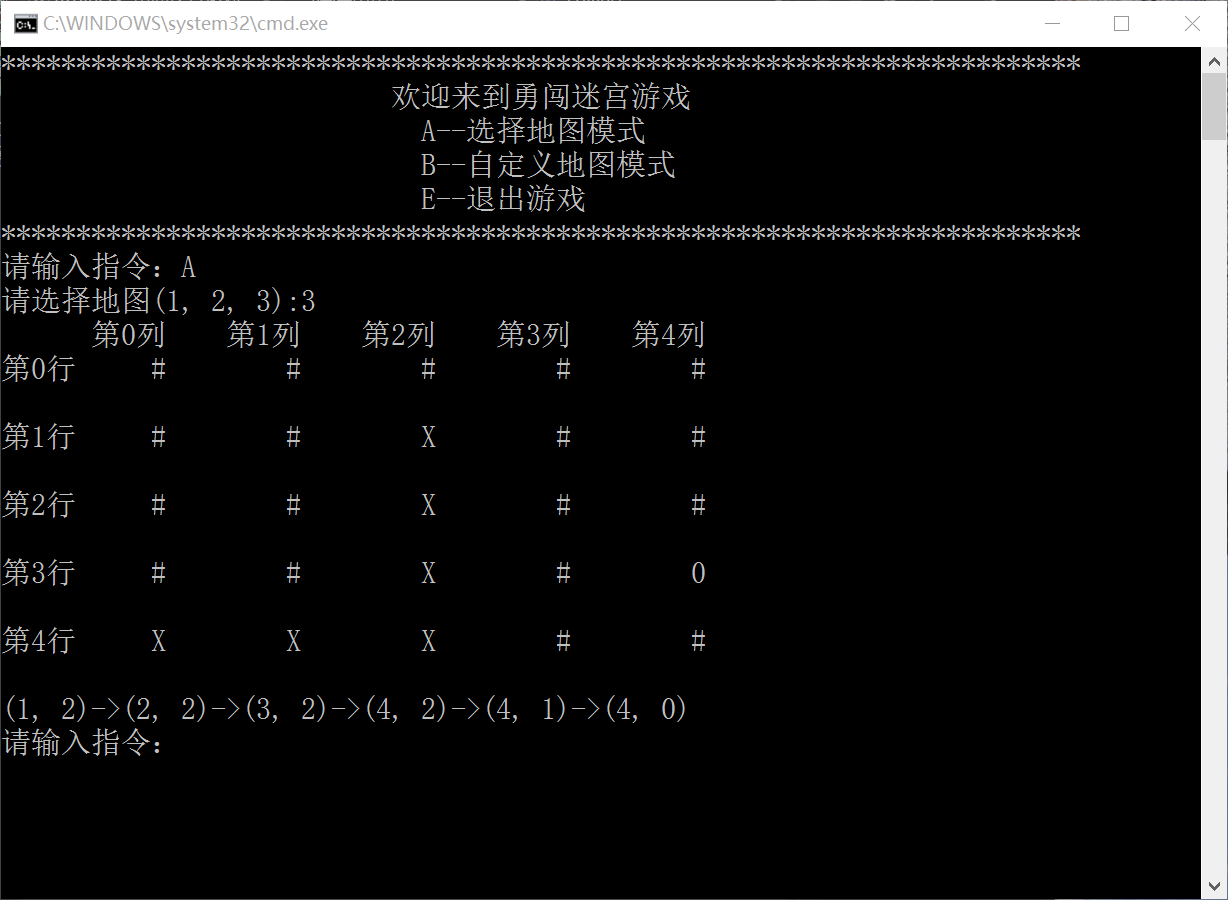
ModeB:由玩家自定义地图进行游戏。要求玩家键入行和列，入口出口等信息，然后自定义输入地图，最后对自定义的地图进行操作。

1. 程序截图：

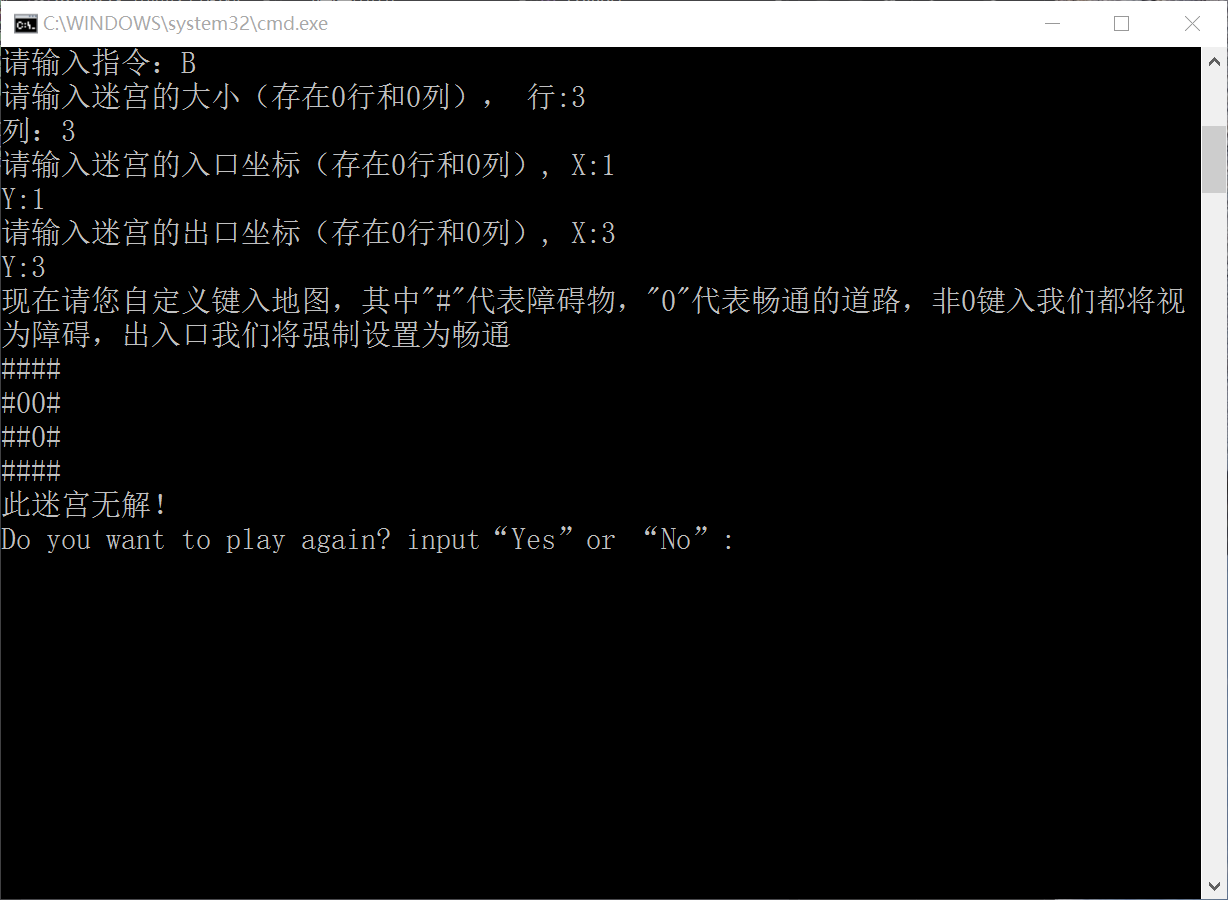
开始界面：

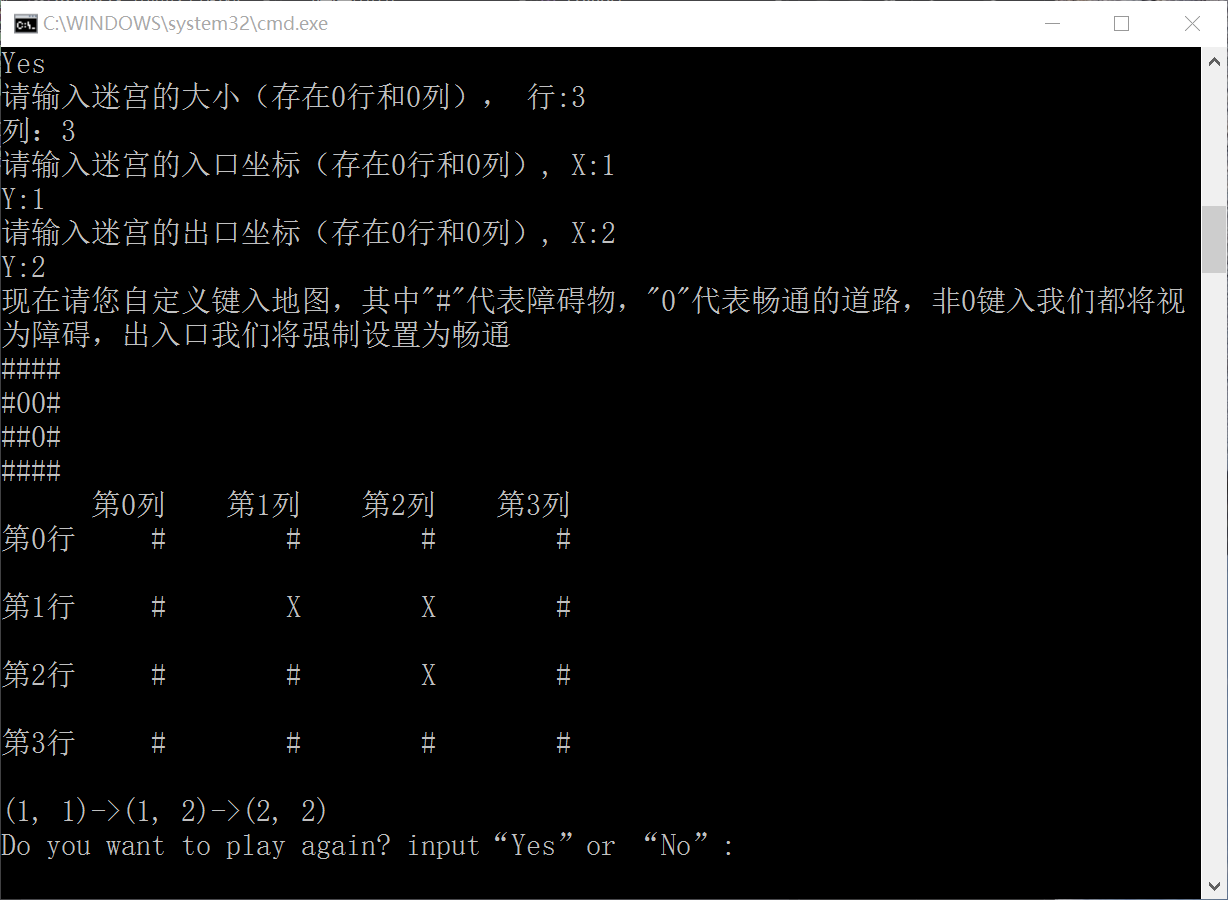


模式A:



模式B:





1. 体会

这个项目使用了深度搜索的算法，中间过程的保存是这道题的关键，即使用了回溯法。运用递归思想，对所有路径进行一次遍历，最后给出可行路径。