

# 大作业——拼图游戏

## 1、简介

拼图游戏是一款常见的 DIY 游戏。游戏创作者可以任意选择一张图片，将其切分成  $M \times N$  个小块，去掉最后一块，再打乱顺序；游玩者则试图经过若干步移动恢复小块的顺序，每次移动可以将与空格相邻的小块移动到空格处。

例如，游戏创作者选择图 1 所示的图片；将其切分为  $3 \times 3$  个小块，并去掉最后一块，形成图 2 所示的样子；随机打乱顺序后，形成图 3 的样子；如果我们将 9 个小块按先行后列的顺序编号为 0-8，8 号小块被去掉，用-1 表示空格，则图 3 就可以对应为图 4 的矩阵。



图 1



图 2



图 3

3	7	0
2	1	5
4	6	-1

图 4

图 4 的状态有两种合法的移动方式，分别如图 5 和图 6 所示。其中，将图 4 中空格上方的小块下移形成了图 5，将图 4 中空格左侧的小块右移形成了图 6。游玩者的目的是通过若干步合法的移动，使得状态变为图 7 所示。

3	7	0
2	1	-1
4	6	5

图 5

3	7	0
2	1	5
4	-1	6

图 6

0	1	2
3	4	5
6	7	-1

图 7

让我们编写一个程序完成上述游戏功能吧。

## 2、功能要求

(1) 读图：输入一行，包含图片的完整路径名，读取该图片进行游戏，图片格式可参考：

[https://en.wikipedia.org/wiki/BMP\\_file\\_format](https://en.wikipedia.org/wiki/BMP_file_format)

(2) 设置难度：读图成功后，输入一行，包括两个整数，M 和 N，

用来设置难度；

(3) 随机变换：完成难度设置后，打乱图片方块，显示打乱后的状态；

(4) 可行性分析：判断并提示打乱后的状态是否能够复原；

(5) 开始游玩：

(5.1) 移动：通过输入命令 W、A、S、D 进行移动，对应的意义如下表：

输入命令	意义
W	将空格上方的小块向下移
A	将空格左侧的小块向右移
S	将空格下方的小块向上移
D	将空格右侧的小块向左移

(5.2) 每次移动后显示当前状态，并判断是否复原；如果复原，给出成功复原的提示；

(5.3) 重新游戏：输入命令 R，重新开始游戏，恢复成步骤 4 结束的状态重新游玩；

(5.4) 保存图像：输入命令 G，下一行输入完整路径名，保存当前状态的对应的图像；

(5.5) 存档：输入命令 I，下一行输入进度名称，保存游戏进度（游戏进度并不永久保存，只在程序运行期间有效）；

(5.6) 读档：输入命令 O，下一行输入进度名称，载入游戏进度；

(5.7) 自动还原：输入命令 F，自动完成拼图复原，以 W、A、S、

D 组合的形式输出复原步骤；

(5.8) 退出：输入命令 P，退出游戏，程序结束。

### 3、补充说明

图像分为  $M \times N$  块后，第  $i$ （从 0 开始）行第  $j$ （从 0 开始）列的小块的编号为  $i * N + j$ ；其中，第  $M-1$  行第  $N-1$  列原本编号为  $M * N - 1$  的块被去掉，变为编号为  $-1$  的空格。

注意，本次大作业允许自由设计合理的附加功能，会根据附加功能的合理性、难易度及完成度酌情加分。

### 4、提交要求

- (1) 源码（包括必要的工程文件）；
- (2) 可执行程序（可选交必要的动态链接库）；
- (3) 实验报告（**非常重要!**），应至少包括：编译运行环境说明、关键函数说明、附加功能说明。
- (4) 演示视频（**非常重要!**），应至少包括：编译过程、基本功能演示、附加功能演示。在保证演示效果的情况下，可以使用录屏软件或手机摄像，可以使用视频编辑剪辑软件进行适当加工。视频长度应控制在 5 分钟以内为宜。
- (5) 将上述四条放在一个压缩包内，将压缩包文件命名为“学号-姓名.zip”的形式提交。压缩包大小应控制在 50MB 以内为宜。