

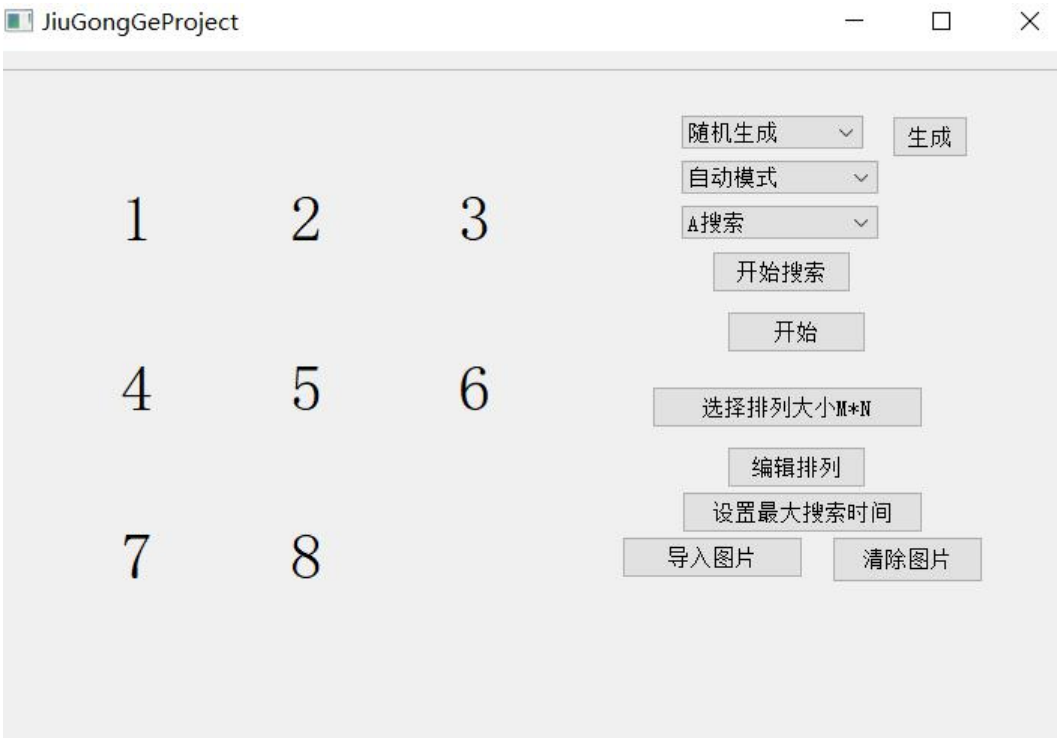
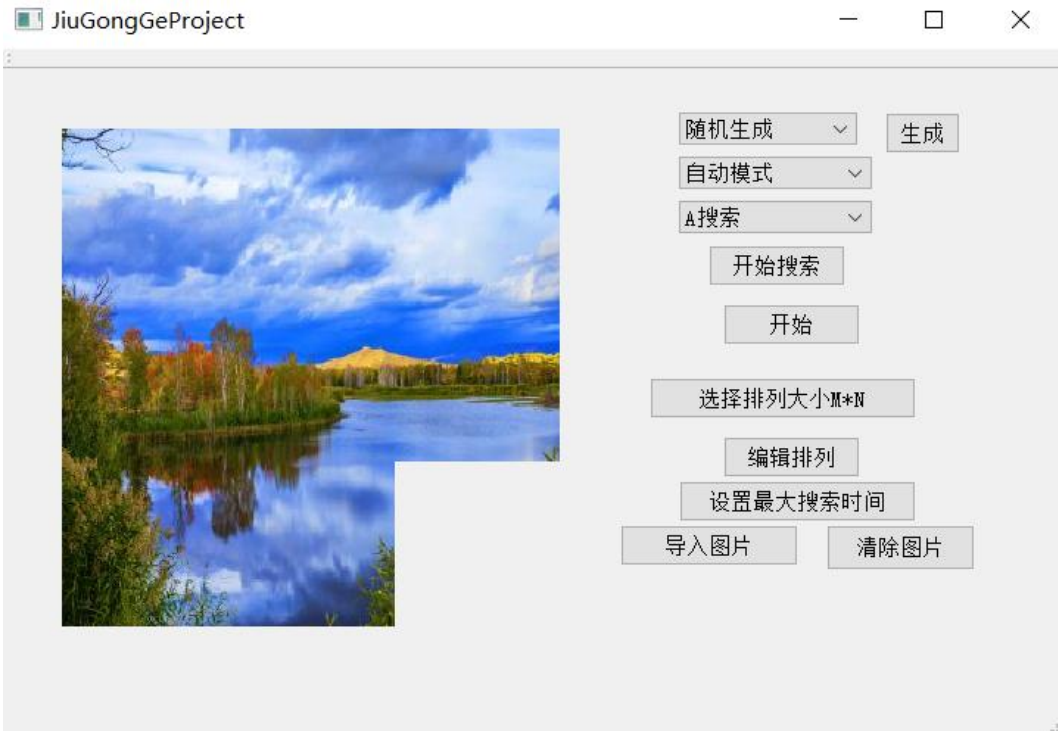
# 人智第一次大作业

李明 自 85 2018011546

重排九宫问题：

## 一、界面设计

### 1. 界面概况

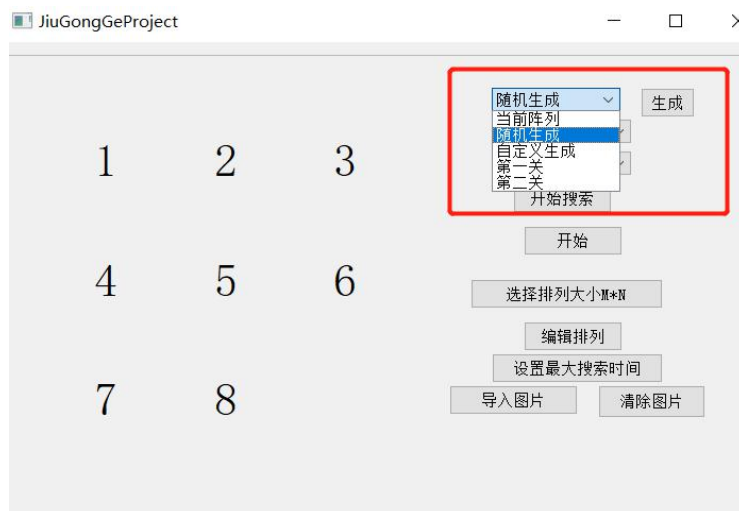


## 2. 界面说明

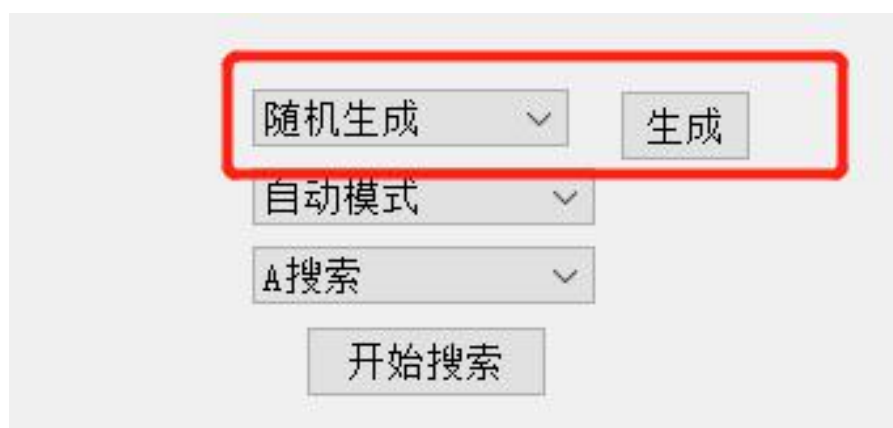
界面实现的功能有：选择生成地图的方式，移动图片的方式，搜索模式，设置地图的大小，自定义编辑地图，设置最大的搜索时间，导入图片和删除图片  
具体说明如下：

### a. 选择生成地图的方式

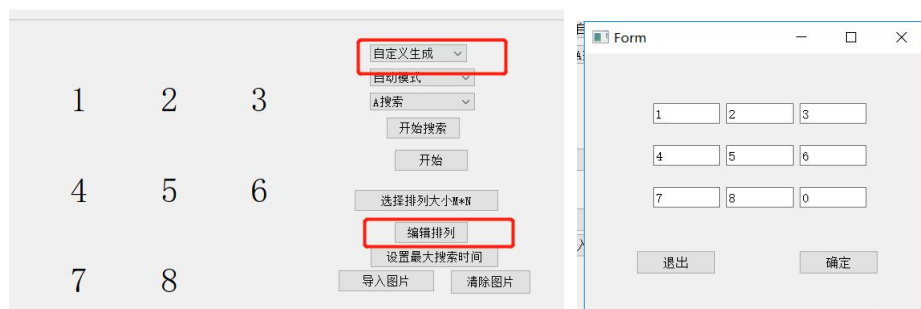
上图中最上面的一个选择框是选择生成的方式，包括随机生成、自定义生成，关卡生成。



**随机生成。**这种状态下右侧会有生成按钮，点击生成时，会出现一个新的地图。大小与当前地图相同。



**自定义生成。**这种状态下，生成按钮消失，同时可以点击编辑排列会出现一个新的窗口进行自定义编辑。



关卡生成。我事先准备了两个关卡可以进行挑战，分别为 3\*3 的和 4\*4 的。

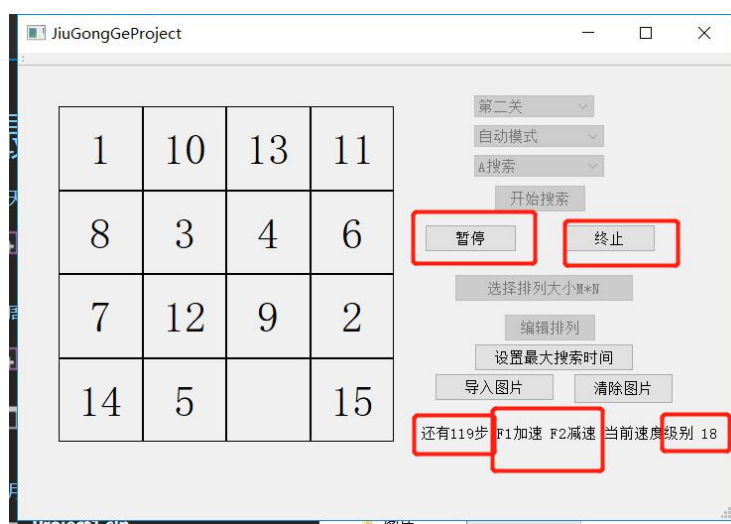


#### b. 自动模式选择

为了方便用户进行游戏，我设置了自动模式和手动模式还原。



**自动模式。**用户需要先进行搜索，然后点击开始，即可进入到自动模式状态。自动状态下，可以点击暂停和终止，下方会显示剩余的步数，当前的速度级别。通过 F1 或 F2 改变速度



**手动模式。**选择手动模式后，需要点击开始进行还原，下方会显示用时。控制是通过键盘的上下左右键进行。



### c. 搜索模式

共有四种模式，A 搜索，深度搜索，宽度搜索和万能搜索。其中万能搜索是仿照人的还原方式，用时很短，能够适用特别大的地图。具体方式请看下面的算法设计介绍。当搜索用时耗尽时，会给出提示，如果成功时，会弹出成功窗口；若当前的地图与最终的地图是一个反向的关系，即有两个块最终会反过来，这时会弹出信息，说明搜索结果是一个反向结果。



### d. 设置地图大小

按照下面的提示输入即可

A dialog box titled "Form" with a standard Windows window frame. It contains two input fields: "行数" (Number of rows) and "列数" (Number of columns). Below these fields, it says "范围3-20". At the bottom, there are two buttons: "取消" (Cancel) and "确定" (OK).

e. 自定义编辑地图

需要注意在自定义状态下才能点击，否则会弹出信息。

The main interface for map editing. It features several controls: a dropdown menu set to "自定义生成" (Custom generation), a "手动模式" (Manual mode) dropdown, an "A搜索" (A search) dropdown, a "开始搜索" (Start search) button, a "开始" (Start) button, a "选择排列大小M\*N" (Select arrangement size M\*N) button, a button labeled "编辑排列" (Edit arrangement) which is highlighted with a red rectangle, a "设置最大搜索时间" (Set maximum search time) button, and two buttons at the bottom: "导入图片" (Import image) and "清除图片" (Clear image).

f. 设置最大的搜索时间

按照提示设置即可，如果搜索超时，会退出搜索，返回失败信息。

A dialog box titled "Dialog" with a standard Windows window frame. It contains a label "设置搜索的最大时间" (Set maximum search time) followed by an input field and the unit "毫秒" (milliseconds). Below this, it says "时间范围为100-300,000". At the bottom, there are two buttons: "OK" and "Cancel".

#### g. 导入图片和删除图片

点击会产生文件框，可以自己进行选择图片或者点击清除回到数字状态。有一点需要注意，当格数很大（>10）时，图片分割时会产生缝隙，但是不会影响搜索结果。



## 二、算法设计

在重排九宫问题中，使用了四种方法进行还原。

### 1. A 算法

A 算法参考了老师在上课时讲到的 A 算法的过程。利用 close 表和 open 来存储访问的节点。

#### A 算法具体过程如下：

- 将根节点放到 open 表中。
- 从 open 表中选择 f 值最小的节点进行扩展，并将这个节点从 open 表中删除，并放入 close 表中。
- 如果遇到的扩展的节点已经访问过，那么不再进行扩展。
- 直到找到最终节点或者内存或时间耗尽退出程序，返回失败或成功以及成功时搜索的路径。

#### A 算法的评价函数设计：

我采用了三部分来描述评价函数：

第一部分是搜索的深度，第二部分是当前状态与最终状态的每个位置的不同的垂直水平距离之和，第三部分是每个点的右方和下方挨着的点是否在最终的状态下挨着。这三部分的权重是通过不断地尝试获得的，选取了其中效果比较好的权重作为最终的权重。

#### A 算法使用的数据结构：

A 算法使用优先级队列来实现 open 表，用 stl 中的 map 存储已经访问的节点，这样可以快速找到一个节点是否已经被扩展。

#### A 算法的适用说明：

对于设计的 A 算法，3\*3 的地图可以在 1s 内完成，大约有 95% 的 3\*4, 4\*3, 4\*4 的地图可以在 10s 内完成，更大的地图的时间会很长，一般在 10s 内不能完成。

## 2. 深度搜索算法

深度搜索算法是 A 算法的退化版本，主要区别在于将 open 表换了，换成了优先弹出最后进来的节点，搜索会比 A 算法慢很多，其余的与 A 算法相同。

## 3. 宽度搜索算法

深度搜索算法也是 A 算法的退化版本，与 A 算法的区别为 open 表采用了后进后出的结构，其余与 A 算法相同。搜索效果一般不如 A 算法。

## 4. 仿人还原方法

这种方法主要是参考了人类在玩这个游戏的策略，即每次只关注某个点，从上到下、从左到右进行还原。在这个还原过程中，虽然不保证结果是最优的，但是可以很快的找到巨大的地图的路径，经过试验，可以在 1s 内完成 19\*19 的搜索。

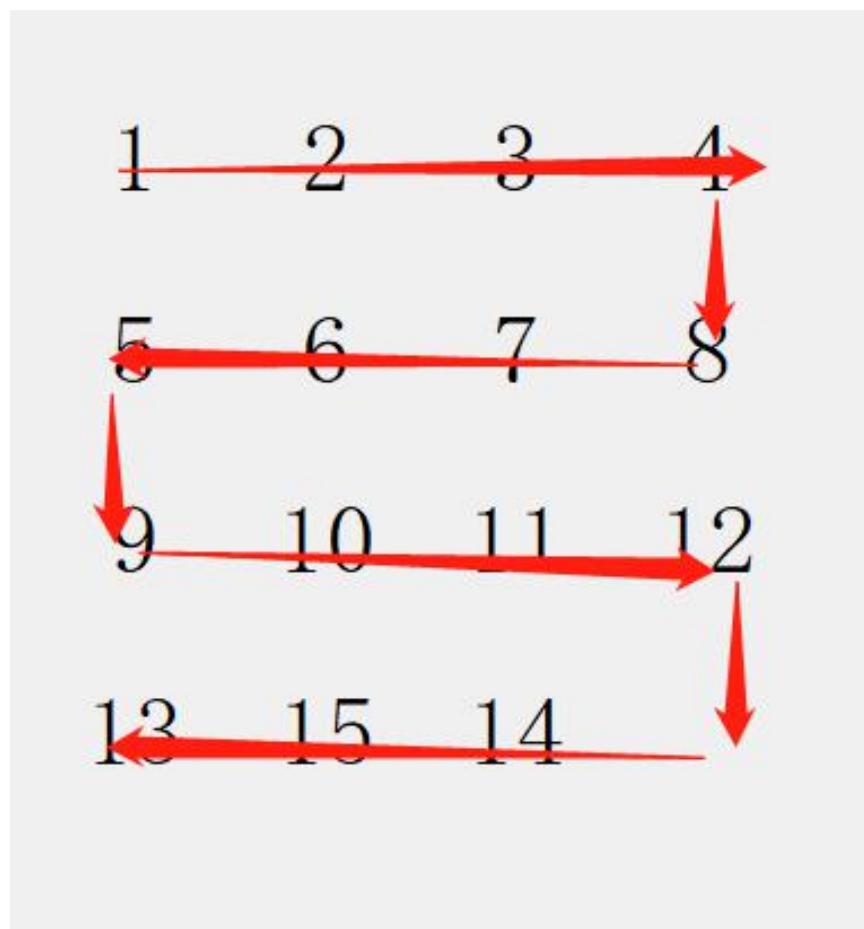
### 算法实现过程：

算法的实现是从左上角开始，每次还原一个点，还原完一行之后还原下一行。需要注意的是每行的最后一个需要特殊考虑，以及最后两行需要特殊考虑。实现过程中，逻辑十分复杂，需要将每种情况都考虑到，代码工作量很大，而且很容易出差错。

## 5. 关于还原状态的证明

证明如下：

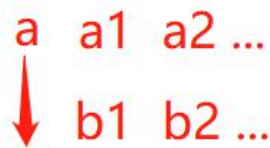
首先定一个一种遍历顺序如下：





按照箭头方向进行遍历, 会有奇数和偶数个逆序对, 这样会将所有的地图分为两类, 下面进行详细证明。

- a) 进行左右移动时, 显然不改变逆序对个数
- b) 进行上下移动时, 如下图所示, 假设在  $a$  所在的行从左到右遍历, 那么  $b$  的行为从右到左, 那么当上下移动时, 改变的逆序对只有  $a$  与  $a_1, a_2 \dots, b_1, b_2 \dots$ , 共有  $2k$  个, 那么可知逆序对每次会改变偶数个, 不改变逆序对的奇偶性。



由  $a$ 、 $b$  可知结论正确。

在程序中判断过程中, 我就是先采用了这样的方法进行判断, 然后再进行还原。

### 三、遇到的问题

在这个大作业中, 遇到的问题有下面的这些。

1. 由于采用的 C++, 需要自己管理内存, 需要自己进行回收, 在开始时, 我的 open 表和 close 表中的元素都是需要的时候进行申请, 但是运行完了之后发现内存没有释放, 最终我才用了一种自己在程序开始时申请了一大块内存的方式, 每次需要用到新内存时, 直接从这块内存中进行分配。最后释放内存时只需要释放一次即可。
2. 采用的 QT 显示图片时, 会有严重的错位, 后来发现是 QT 中 QLayout 与 QWidget 之间并不是大小相等, 而是 QLayout 的大小会比 QWidget 小 20, 加上了偏差后, 显示图片好了很多。
3. 仿人还原算法的实现十分的复杂, 这个程序改了很多很多次, 从代码量上看, 这一个函数用了 1040 多行, 中间大量的分支, 而且就算考虑分函数进行写, 其中的逻辑也是很复杂, 最后能实现可以说是有种幸运在里面, 没有出现十分严重的 bug, 每次遇到问题时改某一块的东西, 来来回回修改, 最终实现了。
4. A 算法的启发函数。这个需要一遍一遍地运行数据进行验证, 最后才找到了一个比较可靠的权重, 能够应付大多数的  $4 \times 4$  地图, 再大的地图也还是会有搜索的问题, 此时就需要使用万能搜索算法进行还原。