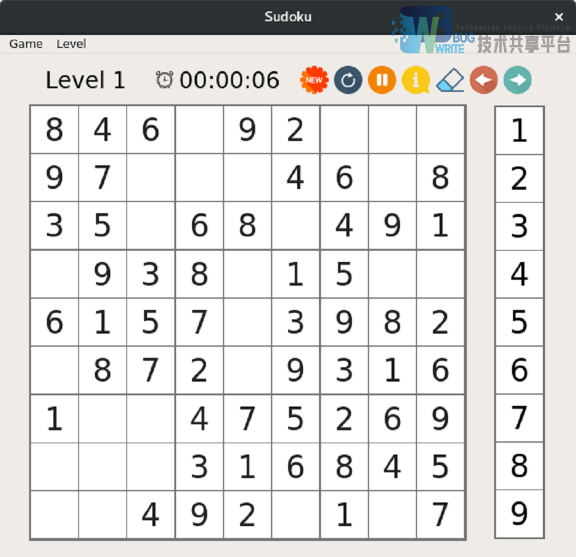
**实习二 基于舞蹈链（Dancing Links）的**

**数独游戏设计与实现**

**一、实习目的与要求**

**【问题描述】**

数独，是源自18世纪瑞士的一种数学游戏。是一种运用纸、笔进行演算的逻辑游戏。玩家需要根据9×9盘面上的已知数字，推理出所有剩余空格的数字，并满足每一行、每一列、每一个粗线宫（3\*3）内的数字均含1-9，不重复。本次实习要求设计并实现一款数独游戏软件（下图为9x9数独游戏示例）。



**【基本要求】**

1、采用舞蹈链（Dancing Link）算法实现数独游戏的求解；

2、采用控制台或GUI，实现语言不限；

3、设计并实现数独的基本功能，可根据个人情况对以下功能进行裁剪：

（1）新游戏：玩家可以开始一局新的游戏

（2）重玩：玩家可以重新开始本局游戏

（3）暂停：玩家可以暂停该局游戏（即暂停计时）

（4）提示：如果当前已经确定的数都是正确的，玩家将会得到一个未填空格的正确数字；如果当前已经确定的数和答案矛盾，导致整个数独无解，那么所有与答案矛盾的数字将会被粗体标出

（5）清除：清除当前选中格子的所有数字

（6）撤销：撤销前一步的操作，以及取消撤销（最多可支持 50 步撤销）

（7）同时可以通过菜单来实现多达 10 种难度的游戏选择，可以求解任意用户输入的数独问题。

**二、分析与设计**

1. **需求分析与类设计**

数独（Sudoku）是一种运用纸、笔进行演算的逻辑游戏。玩家需要根据9×9盘面上的已知数字，推理出所有剩余空格的数字，并满足每一行、每一列、每一个粗线宫内的数字均含1-9，不重复。 每一道合格的数独谜题都有且仅有唯一答案，推理方法也以此为基础，任何无解或多解的题目都是不合格的。

玩法介绍：设计了三种不同的游戏难度。完成9×9的数独游戏。包含重开、提交、检查当前空、提示当前空、显示答案、撤销、前进、标记、计时和暂停等功能。

游戏盘生成：先生成一个所有单元都是空的空数独，然后随机选择一个空单元，找到它的所有可能解，接着遍历空单元的每个可能解。将每个解填入，求的填入后数独的解的个数。 最后如果有可能解的填入之后数独解个数为1，则此填入此可能解之后的数独即为生成的数独，否则，随机选择一个可能解，重新进行上述步骤。

数独问题解决：提取出数独问题的约束条件，分别解决，然后把数独转换为矩阵。从数独的格子依次转换成行，插入到矩阵中。完成了数独问题到精确覆盖问题的转换。剩下的求解精确覆盖问题，用舞蹈链算法解决。

1. **算法设计与分析**

算法的实现来自：https://github.com/chenshuo/recipes/tree/master/sudoku

数独问题有四个约束条件：

1. 每个格子只能填一个数字；
2. 每行1-9的这9个数字都得填一遍（也就意味着每个数字只能填一遍）；
3. 每列1-9的这9个数字都得填一遍；
4. 每宫1-9的这9个数字都得填一遍。

可以将其转化为一个精确覆盖问题。

**完成约束条件1：每个格子只能填一个数字：**

我们把矩阵的每个列都定义成一个约束条件。

第1列定义成：（1，1）填了一个数字

第2列定义成：（1，2）填了一个数字

第9列定义成：（1，9）填了一个数字

第10列定义成：（2，1）填了一个数字

第81列定义成：（9，9）填了一个数字

**完成约束条件2：每行1-9的这9个数字都得填一遍：**

第N列（1≤N≤81）定义成：（X，Y）填了一个数字。N、X、Y之间的关系是：X = INT((N - 1)/ 9)+1; Y = ((N-1) Mod 9) + 1; N =(X - 1) × 9 + Y。

第82列定义成：在第1行填了数字1

第83列定义成：在第1行填了数字2

第90列定义成：在第1行填了数字9

第91列定义成：在第2行填了数字1

第99列定义成：在第2行填了数字9

第162列定义成：在第9行填了数字9

**完成约束条件3：每列1-9的这9个数字都得填一遍：**

第N列（82≤N≤162）定义成：在第X行填了数字Y。N、X、Y之间的关系是：X = INT(N - 81 - 1)/ 9)+ 1; Y = ((N - 81 - 1)Mod 9)+ 1; N = (X - 1)× 9 + Y + 81.

第163列定义成：在第1列填了数字1

第164列定义成：在第1列填了数字2

第171列定义成：在第1列填了数字9

第172列定义成：在第2列填了数字1

第180列定义成：在第2列填了数字9

第243列定义成：在第9列填了数字9

**完成了约束条件4：每宫1-9的这9个数字都得填一遍：**

第N列（163≤N≤243）定义成：在第X列填了数字Y。N、X、Y之间的关系是：X = INT((N - 162 - 1) / 9) + 1; Y =((N - 162 - 1)Mod 9) + 1; N = (X - 1)× 9 + Y + 162。

第244列定义成：在第1宫填了数字1

第245列定义成：在第1宫填了数字2

第252列定义成：在第1宫填了数字9

第253列定义成：在第2宫填了数字1

第261列定义成：在第2宫填了数字9

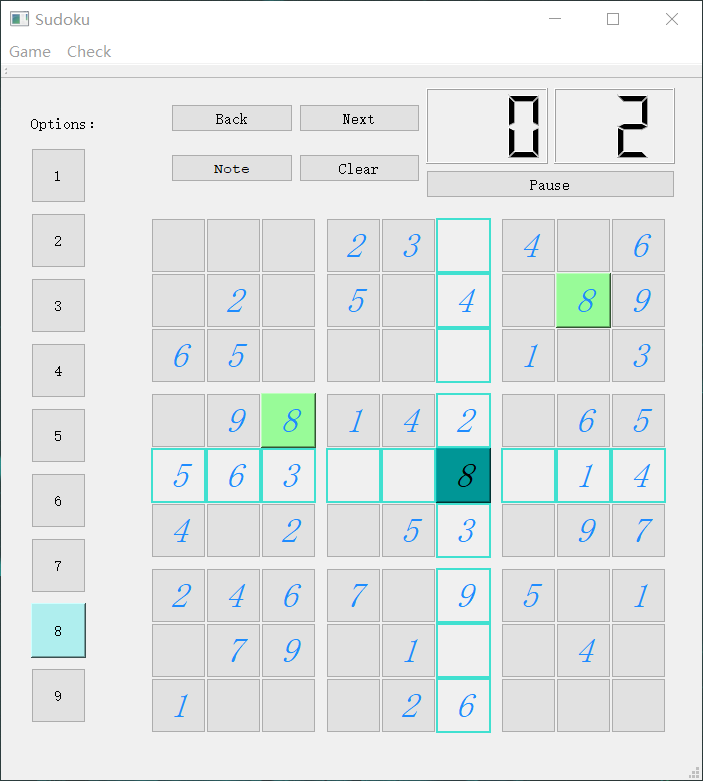
第324列定义成：在第9宫填了数字9

至此，用了324列完成了数独的四个约束条件，矩阵的列定义完成。那接下来，就是把数独转换为矩阵。数独问题中，每个格子分两种情况。有数字的格子、没数字的格子。从数独的格子依次转换成行（1行或者9行）插入到矩阵中。完成了数独问题到精确覆盖问题的转换。剩下的求解精确覆盖问题，就可以用舞蹈链算法解决了。

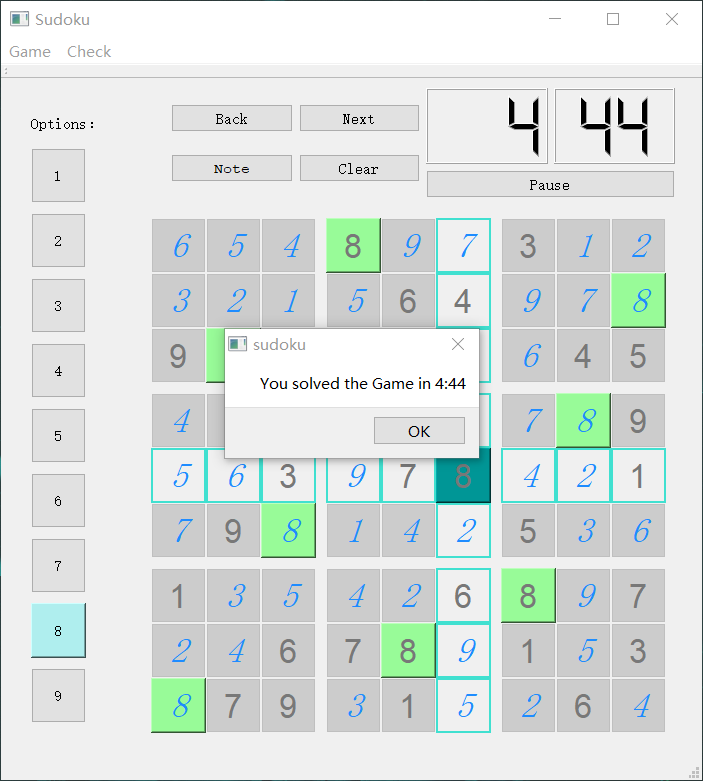
**3、测试与改进**

（1）功能测试

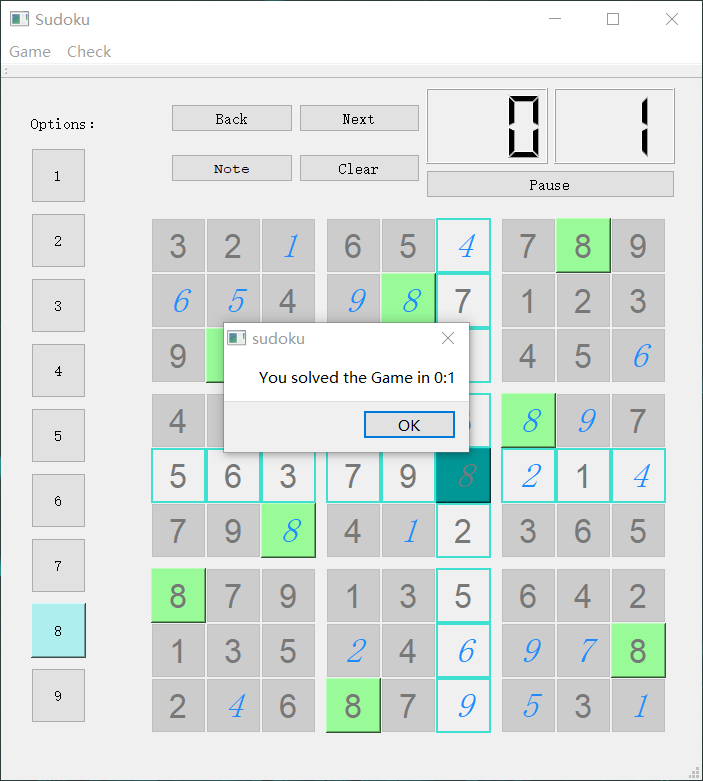
游戏主界面：



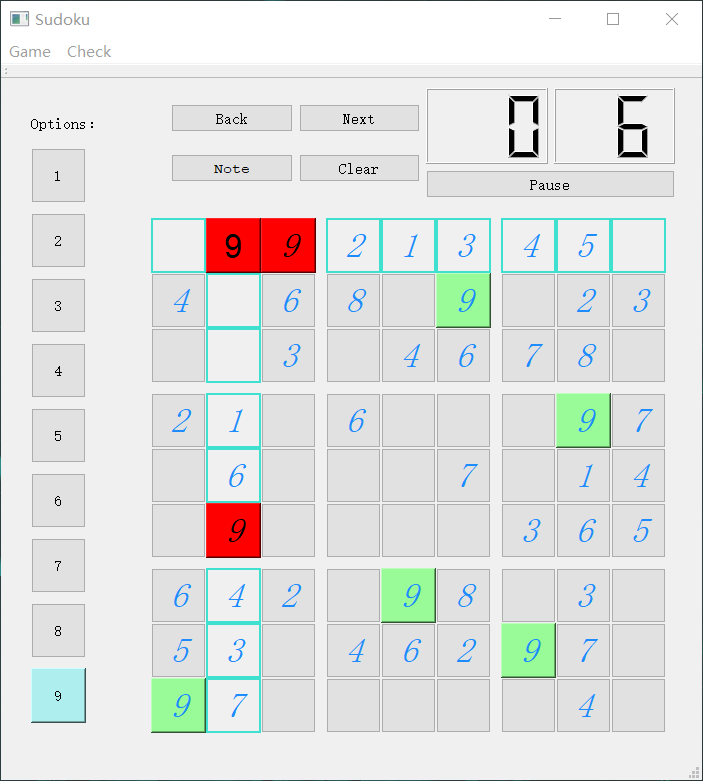
通关界面：



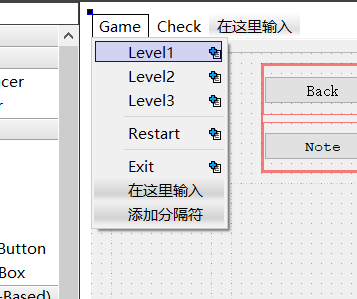
直接输出答案：



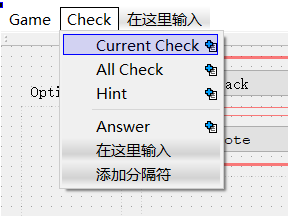
检查当前错误：



难度选择、当前游戏重开、退出：



检查当前格答案、提交所有答案、提示、显示所有答案：



（2）改进与优化

舞蹈链的实现使用了数组模拟的实现方式，除此之外还有指针直接实现的方法。相比较数组模拟调试更容易些，但是时空复杂度相比高了很多，可以对舞蹈链的实现进行优化。

**三、实习小结**

本次实习是实现基于舞蹈链的数独游戏。对舞蹈链的数据结构向数独问题上的迁移花费了较长的时间。相比上次的图像压缩更难上手解决。参考了较多的解决相关问题的博客文章。其次在界面设计上也花费较长时间，对QT界面有了更好的理解。实习中还存在着一些问题，没有及时解决，希望接下来的其他实习可以完成的更好。

-------------------------------

成绩评定：

教师签名：

批改日期：