# 程序设计II 大作业实验报告

姓名：吴骏东

学号：PB20111699

## I.实验题目与要求

本次实验主要内容是实现一个简单的数独软件，具体要求如下：

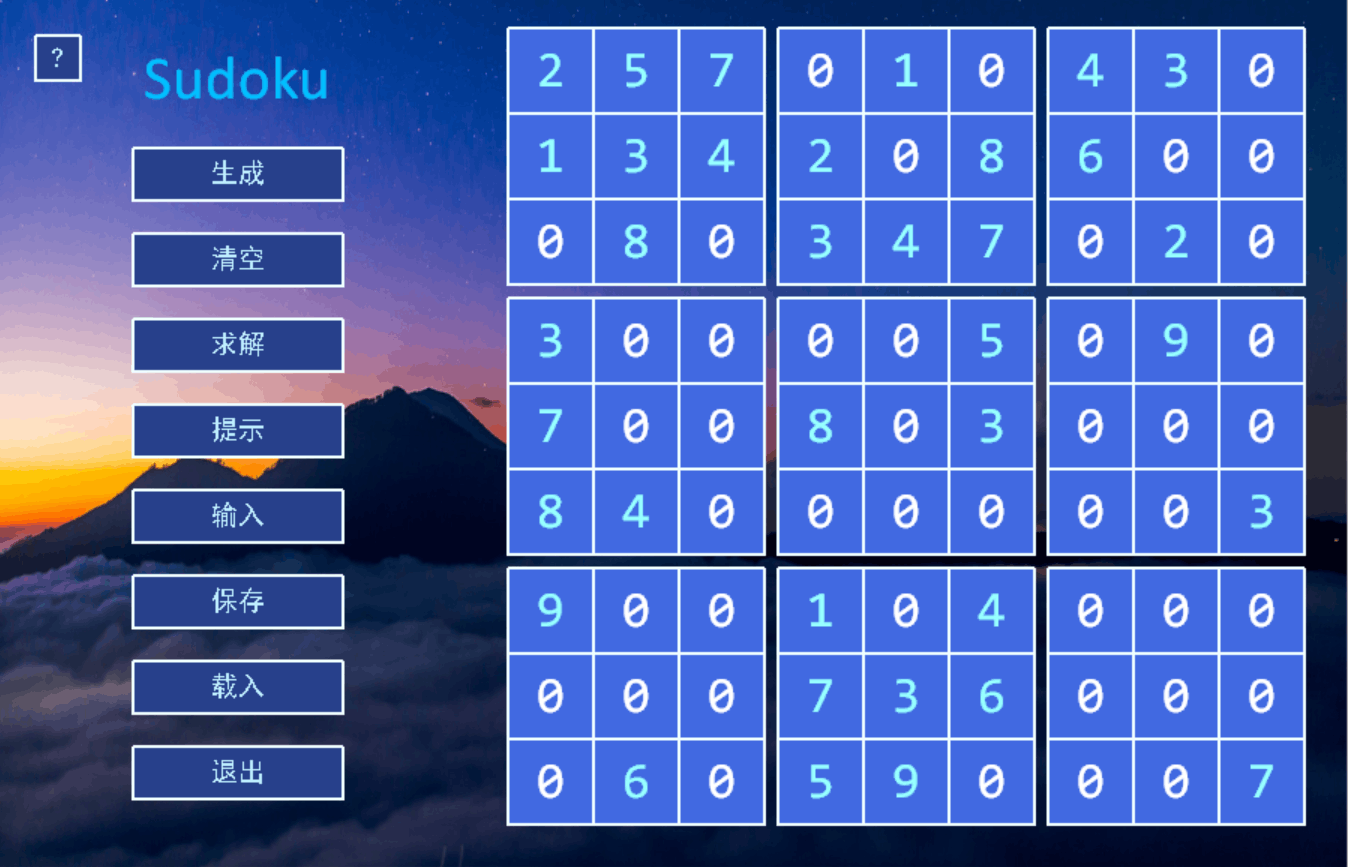
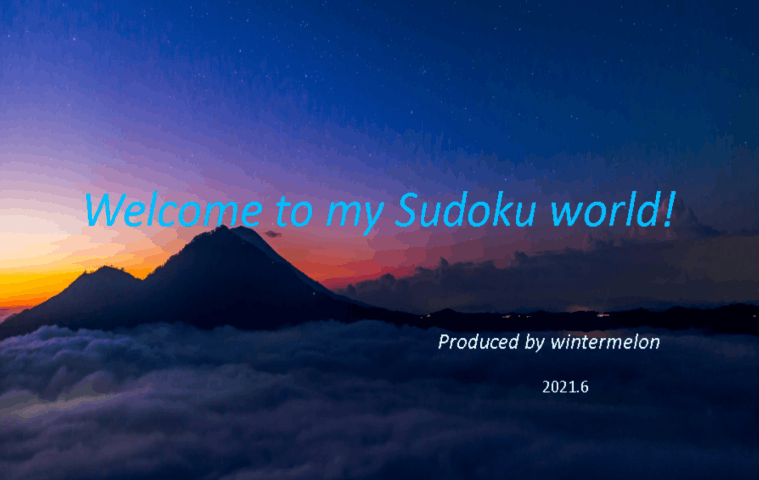
### 基本功能部分

下文提到的数独状态指9\*9的对角线数独内容；

下文使用的数独仅作为格式示例，并不作为内容实例。

#### 1.程序说明信息以及交互

程序运行后，会进入欢迎界面，等待三秒后进入主界面。



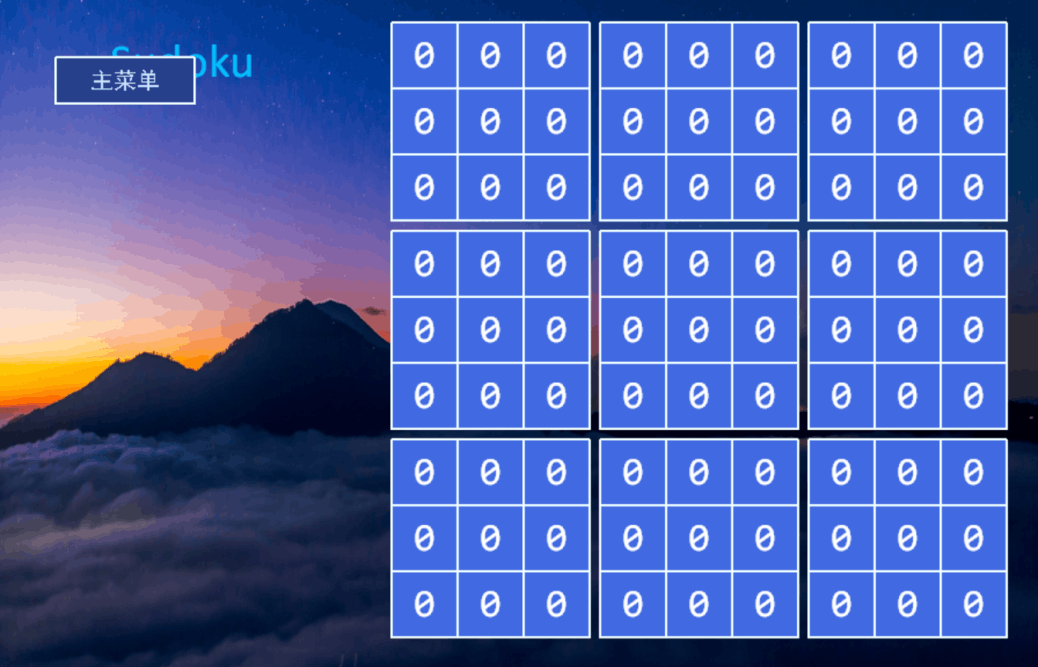
进入主界面后，程序会自动生成一个新的数独。其中白色数字代表可修改、可撤销数字；蓝色数字代表不可修改、不可撤销数字。为“0”的地方为空位，可以填入数字1~9。

#### 2.打印数独

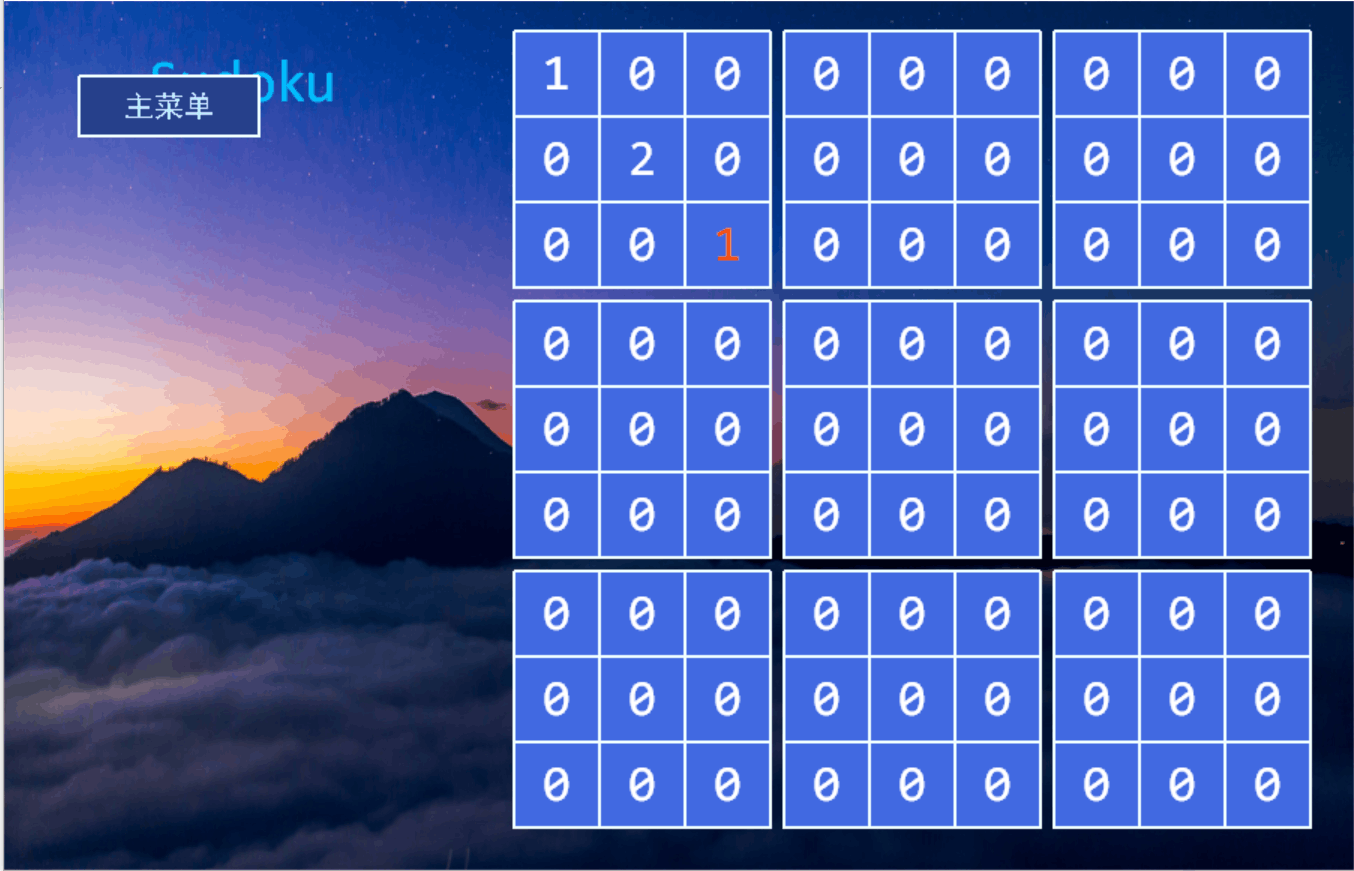
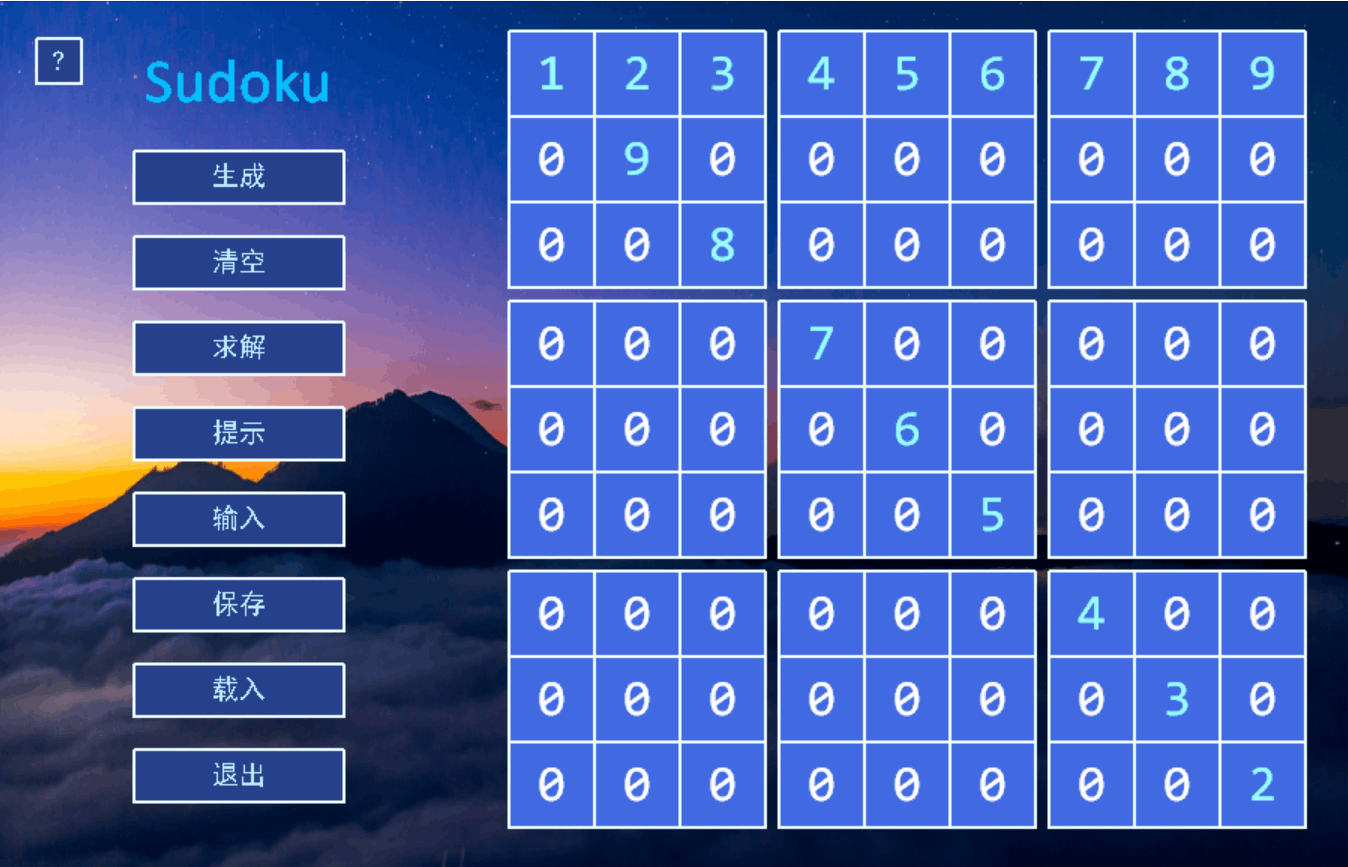
数独主界面会实时显示当前的数独内容。

#### 3.输入数独

程序支持用户自定义输入数独。在左侧按钮栏中左键单击[输入]按钮即可进入输入数独界面。



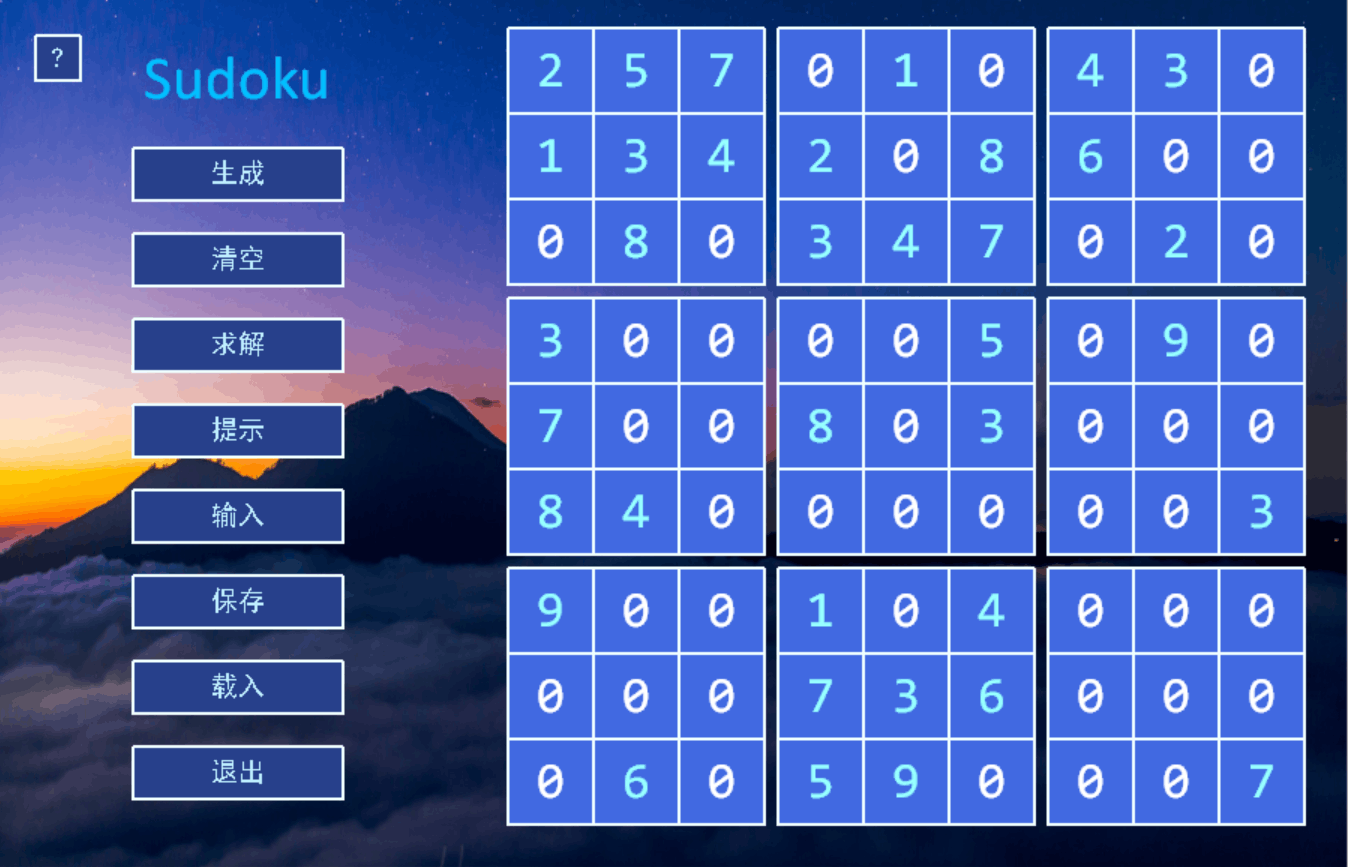
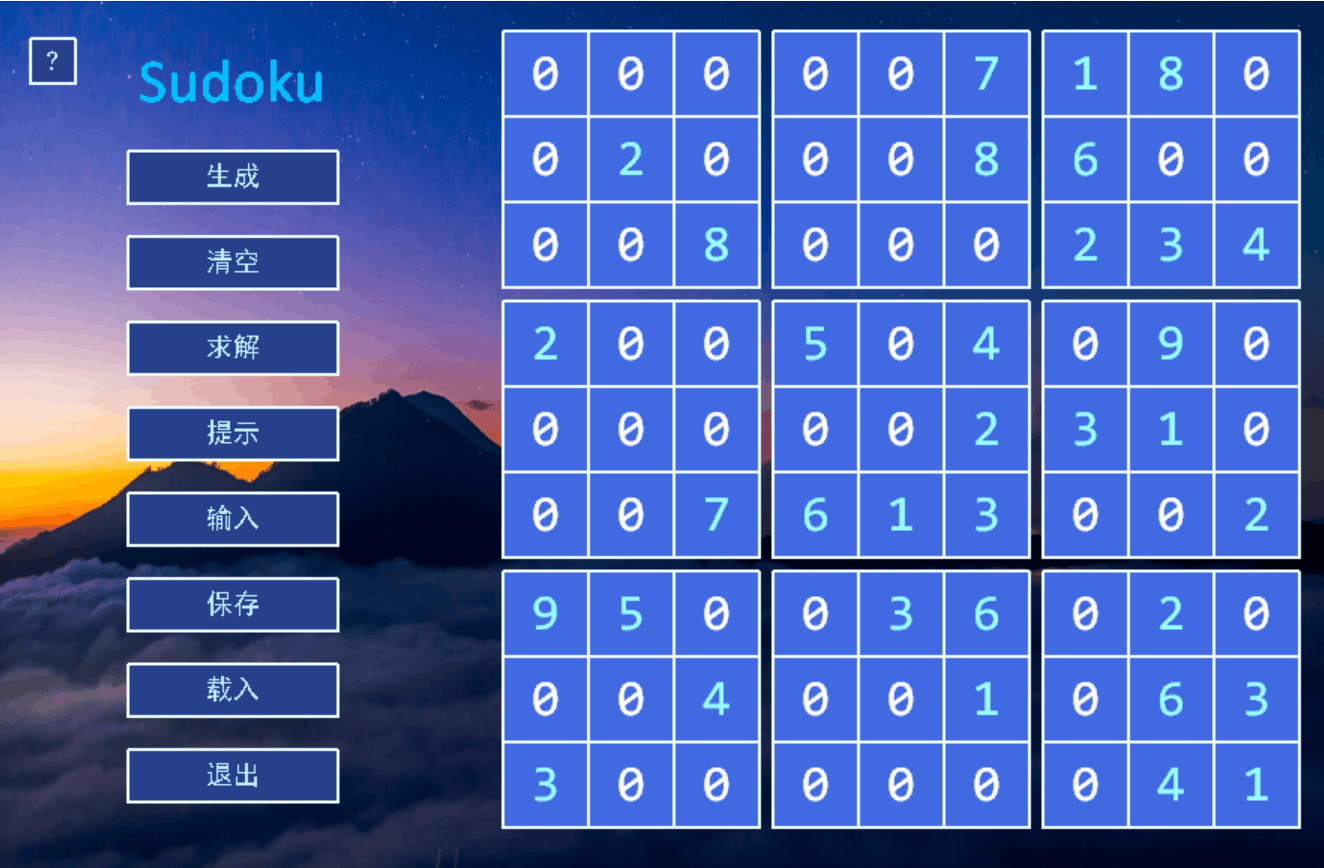
如果输入是合法的（输入不少于17个数字字符，且已填入位置的行列对角线九宫格不重复），在单击[主菜单]按钮后程序将自动存储并返回数独主界面。此时显示的数独即为用户输入的数独。



如果输入不合法，则单击[主菜单]按钮后数独状态不会改变，依然为输入前的状态。

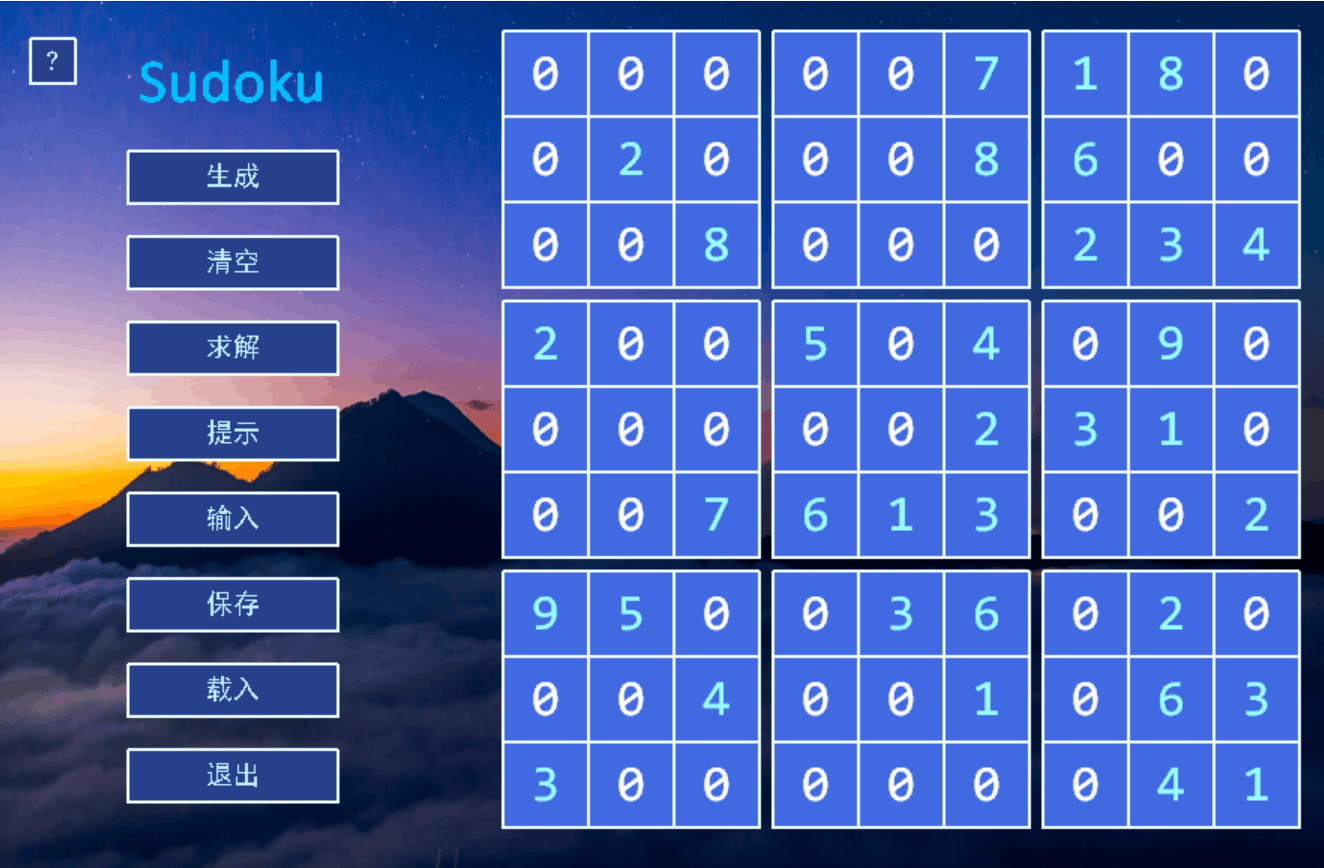
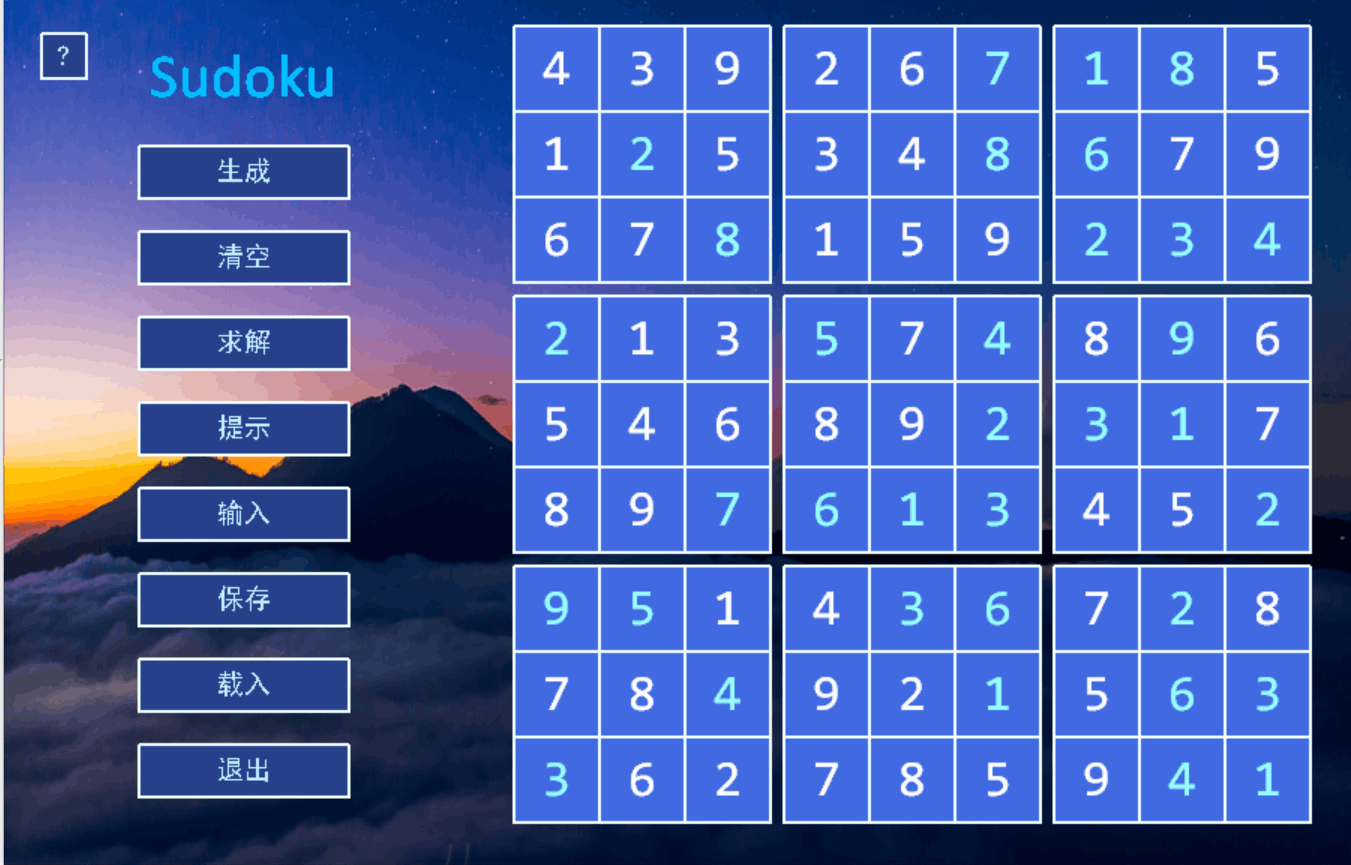
#### 4.生成数独

在单击[生成]按钮后，程序会自动生成一个新的合法的对角线数独。该数独将拥有至少17个非零数字，且一定有解。

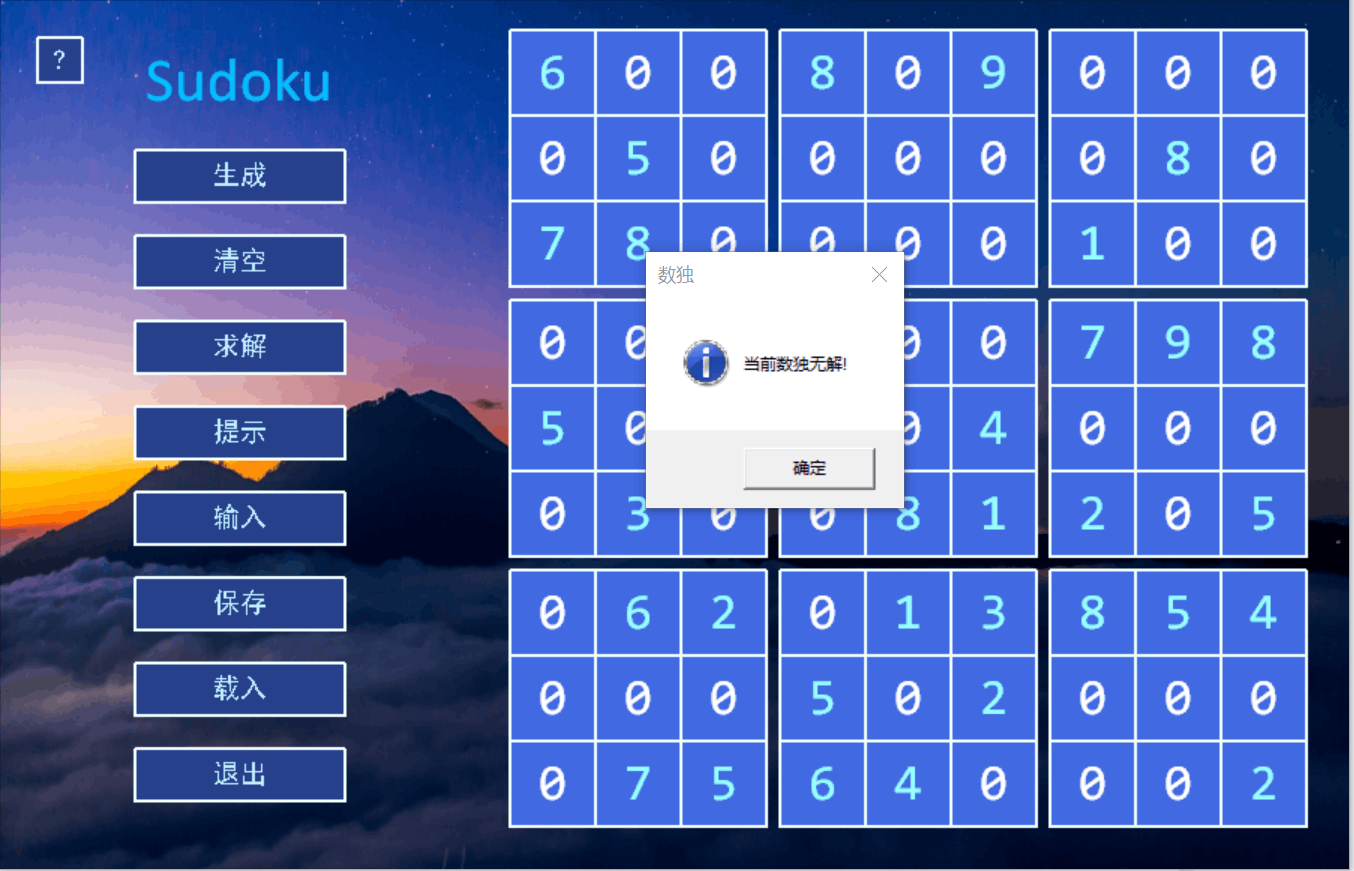
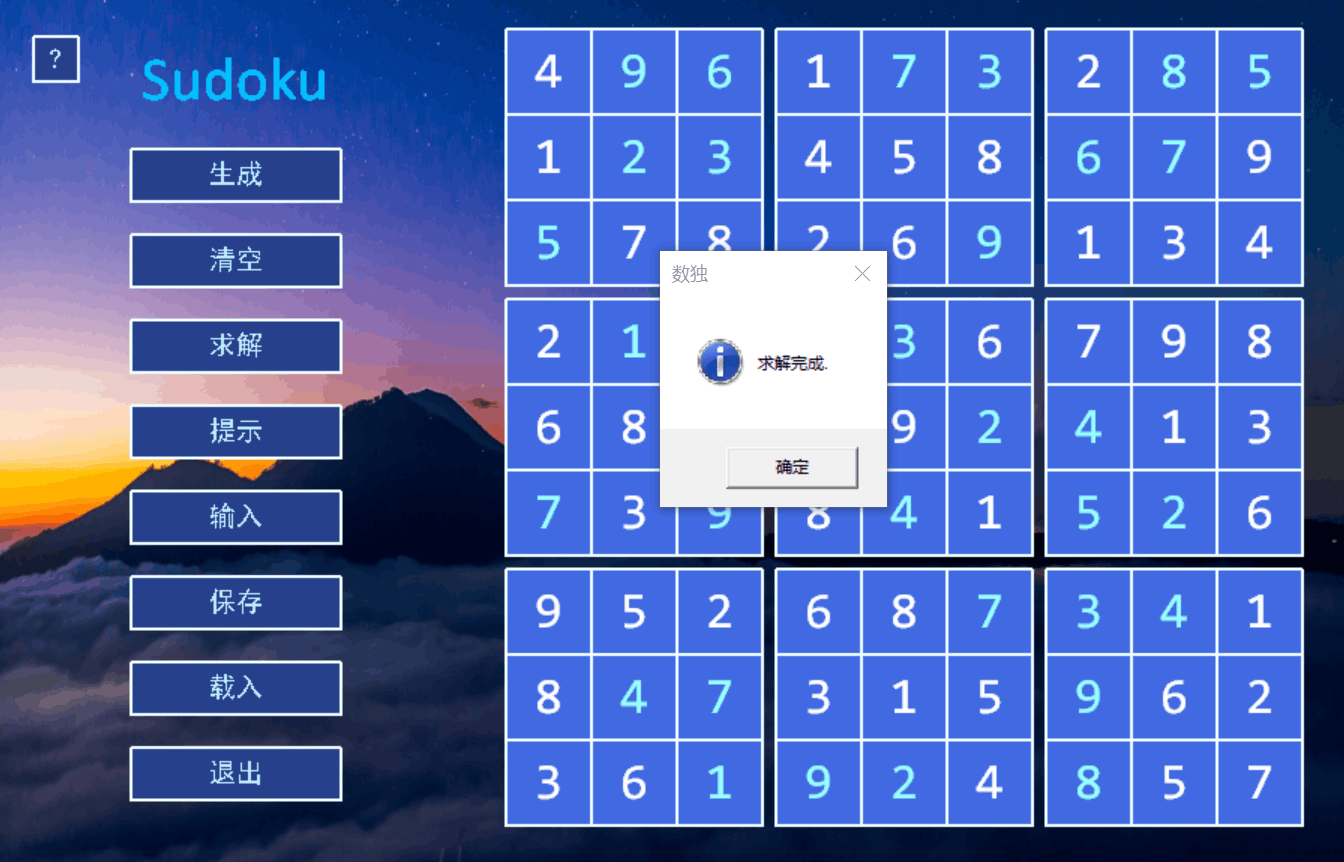


#### 5.输出数独的解

在单击[求解]按钮后，如果数独有解，则会将其中一个解显示在数独主界面上。

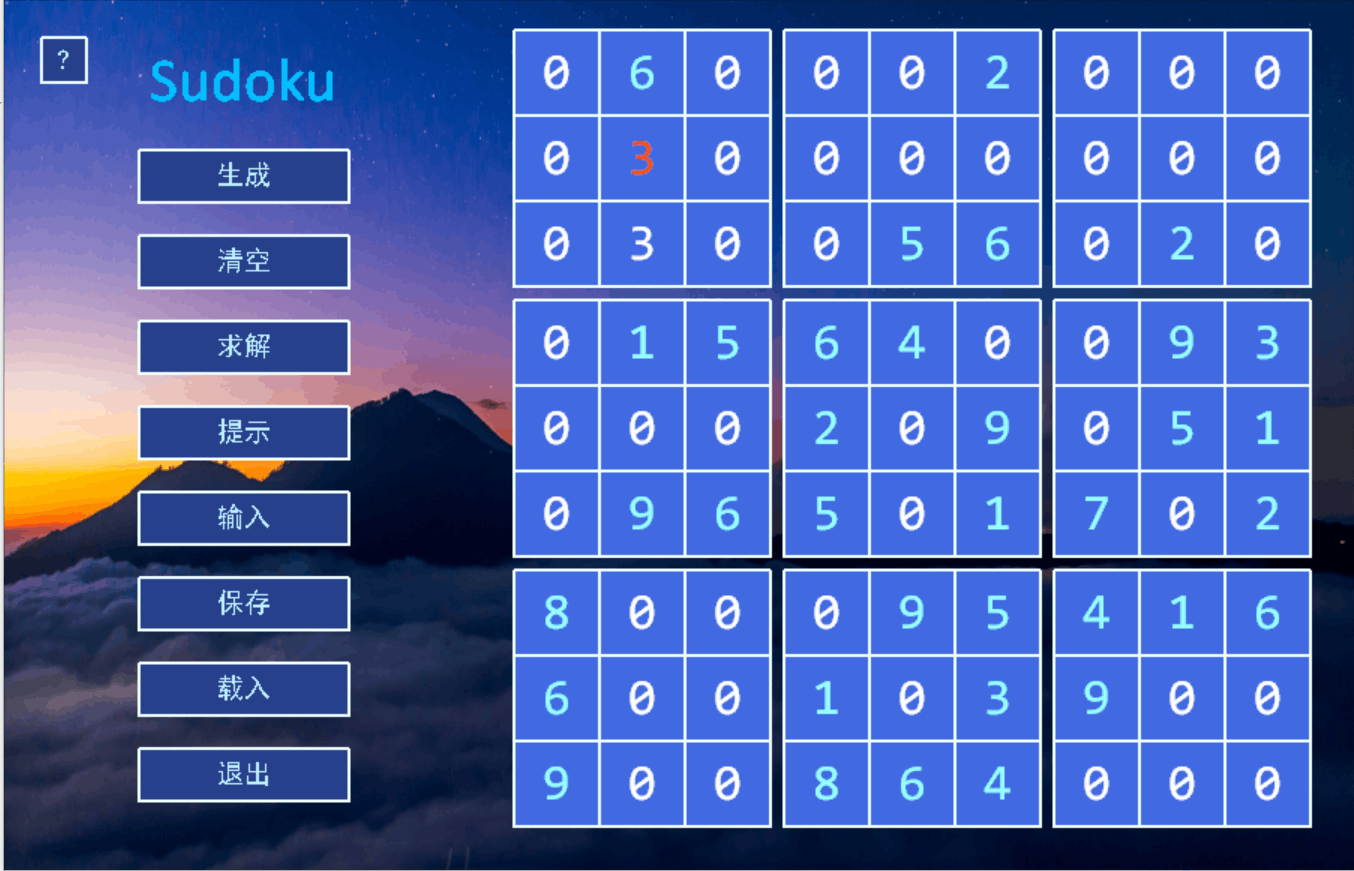
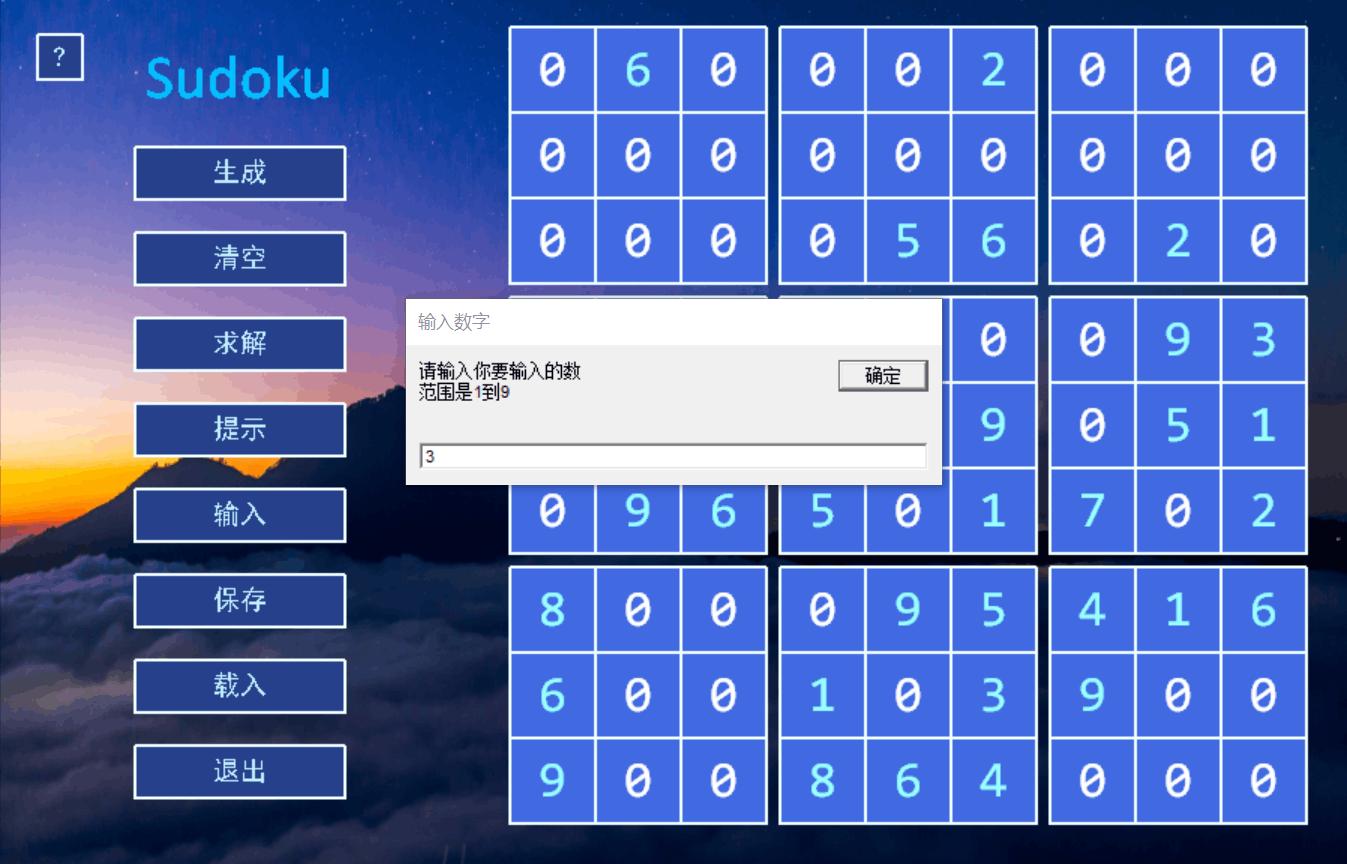
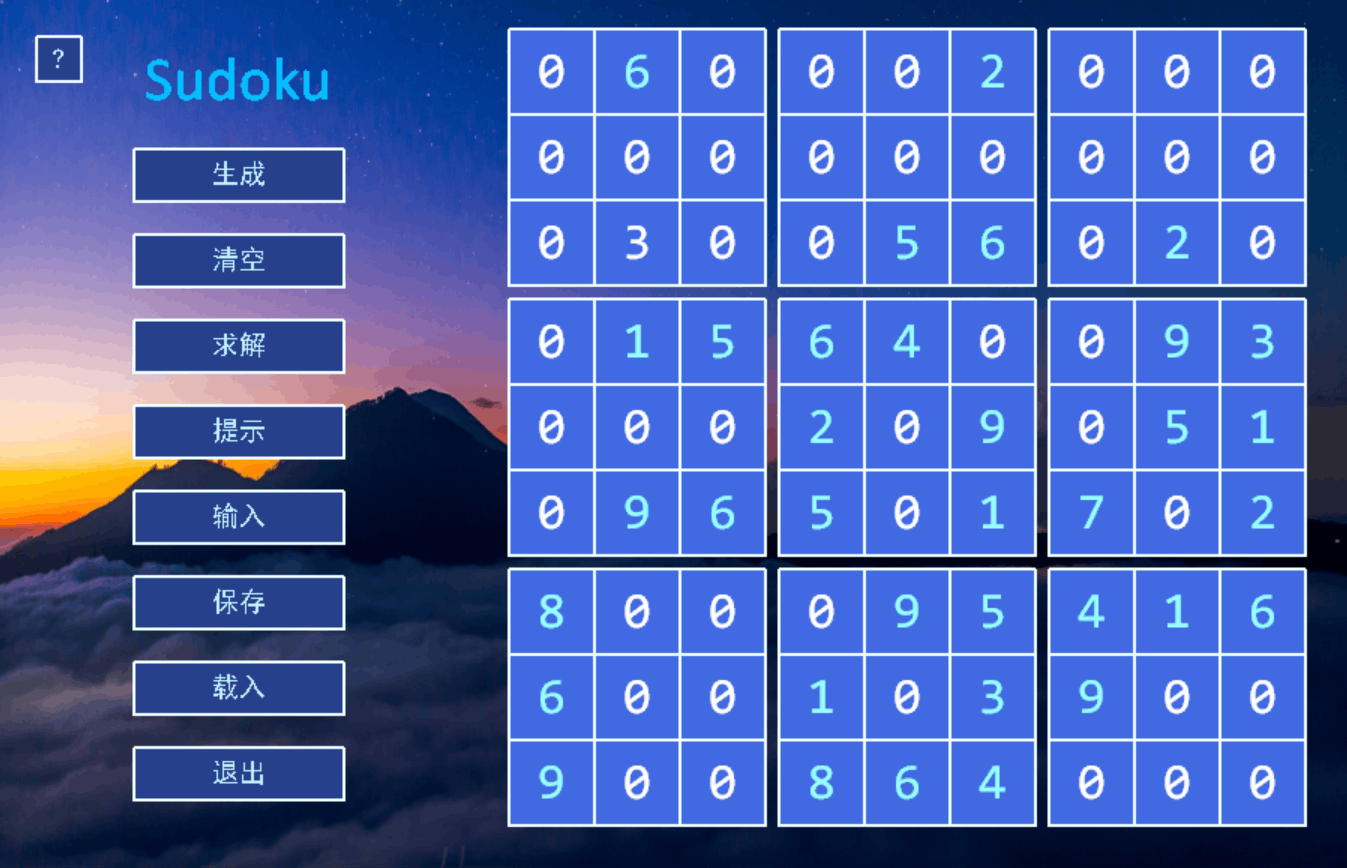


如果数独没有解，则会弹出提示框，并提示该数独无解；如果已经求解完成，则会弹出提示框，并提示求解完成。



#### 6.填入数字与合法性提示

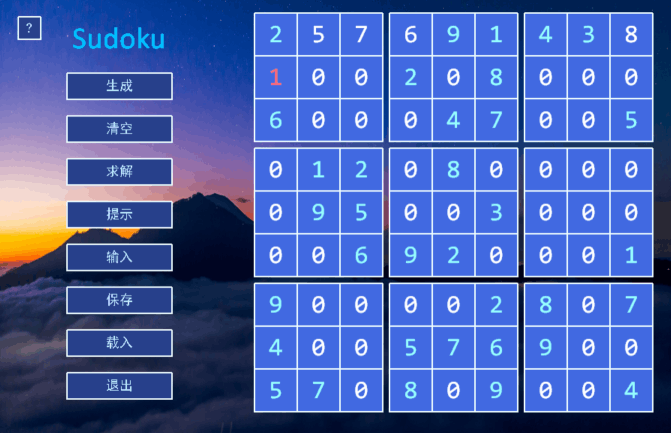
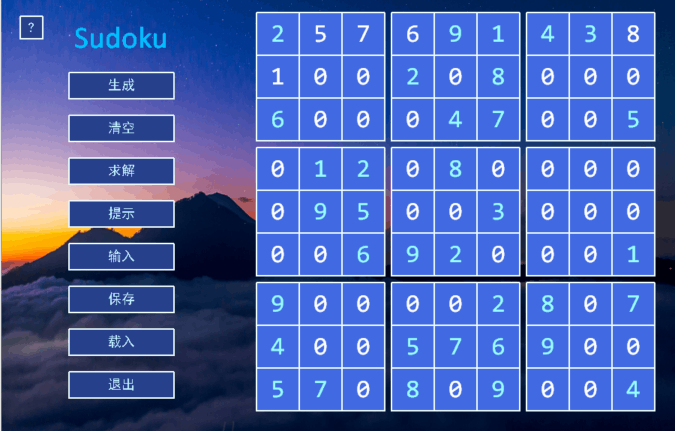
程序支持用户直接在主界面上更改数字。在左键单击白色数字后，程序将会弹出提示框，由玩家在键盘上输入想要填入的数字。如果输入合法，则该处将会显示为白色的输入数字；如果输入不合法，则该处将会显示为红色的输入数字。



#### 在右键单击白色数字后，程序会将该数字还原成0，视为撤销操作。

#### 7.提示功能

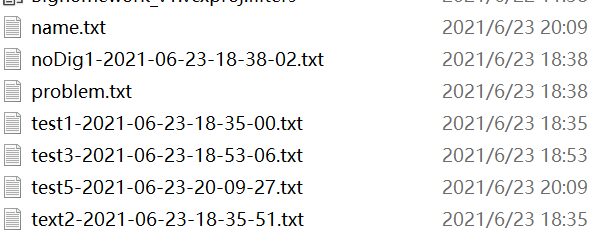
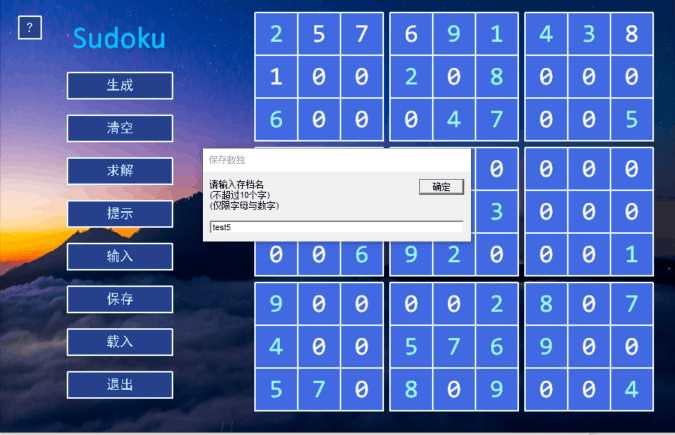
在单击[提示]按钮后，程序会选择一个当前没有填入数字的方格(显示为白色的数字0)，然后填入一个正确的数字。该数字将会先以洋红色闪烁一下，随后变为白色。



与求解功能类似，提示也会在无解、求解完成的状态下分别弹出相应的对话框。

#### 8.保存游戏状态

由于有些数独很难，可能一时半会儿解不出来，程序支持用户实时保存当前的数独状态。在单击[保存]按钮后，程序会弹出提示窗口，由玩家向其中输入用户名(目前支持英文字母+阿拉伯数字的组合)，随后自动保存该数独至相应的位置。



#### 9.加载游戏状态

除了保存数独，程序也支持载入已保存的数独。在单击[载入]按钮后，程序会进入存档加载界面。该界面支持最多20个存档的选择。在单击相应的存档按钮后，程序将会回到该存档存储的数独主界面。



#### 扩展功能部分

**1.图形化界面**

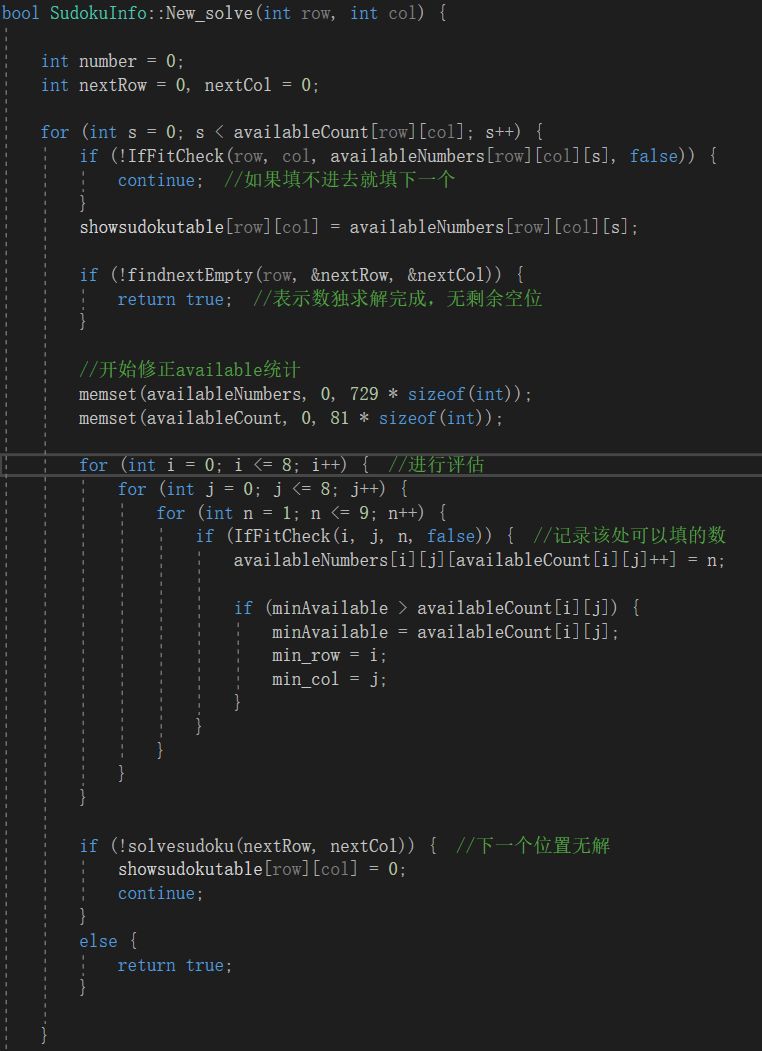
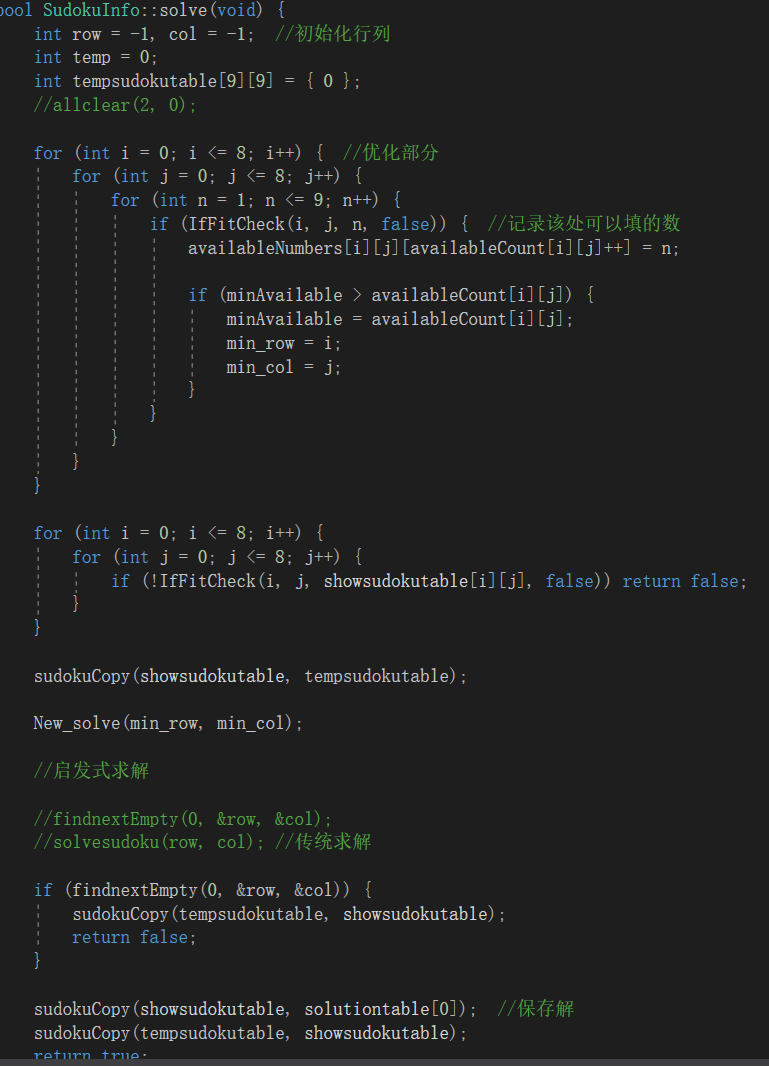
详细介绍如上。

**2.高级搜索技巧**

基于求解数独通用的迭代算法，本程序使用了两种不同的高级搜索算法。

A.启发式搜索

传统迭代算法通常是顺序求解，即找到从当前位置开始顺序向下，第一个未填入数字的方格，然后开始迭代。启发式搜索算法则对每个格子提供了一个状态评估。寻找当前的最优情况进行迭代。本实验中我选择了将方格中所能填入的数字的可能情况数目作为判断标准，每次求解都将会从可能状态数最小的格子开始。这样将有效减少迭代的层数，从而提高效率。



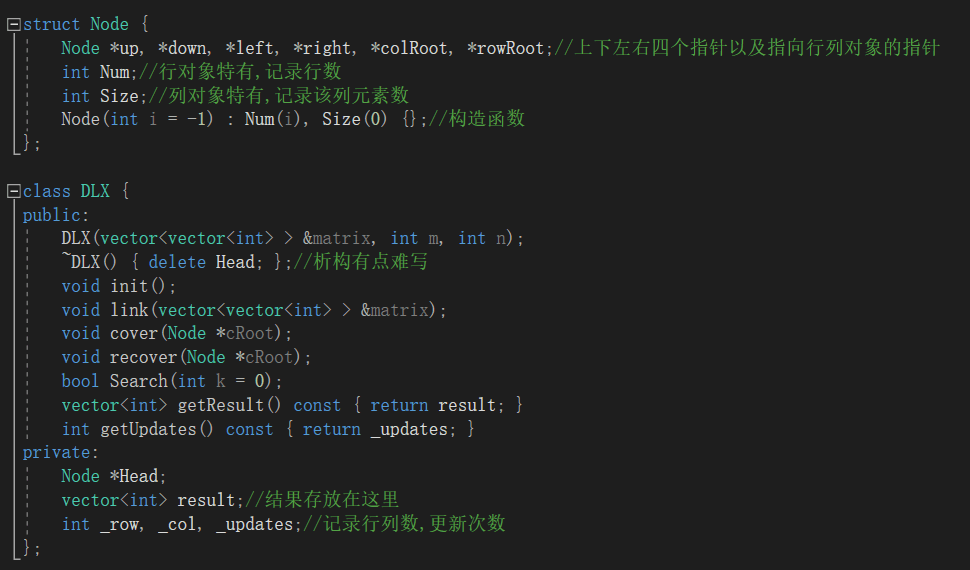
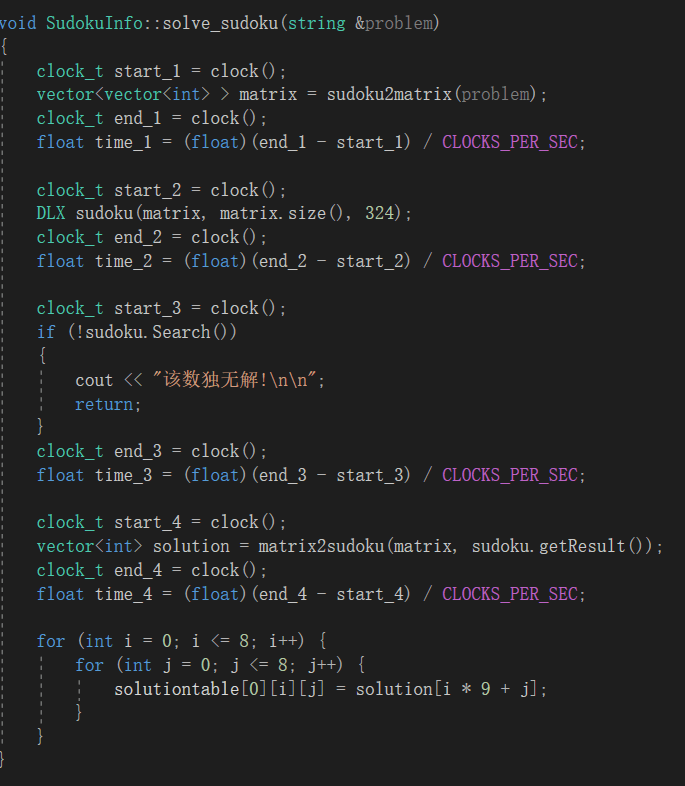
启发式搜索求解算法

求解模块封装函数

1. DLX算法

舞蹈链算法是求解覆盖问题的高效算法，在求解数独时，首先将标准数独方阵转化为0-1矩阵，随后进行求解。本实验中额外增加了数独模式切换开关，在选择“对角线数独”时程序将使用算法A求解；

双向十字链表结构体与DLX类



求解算法

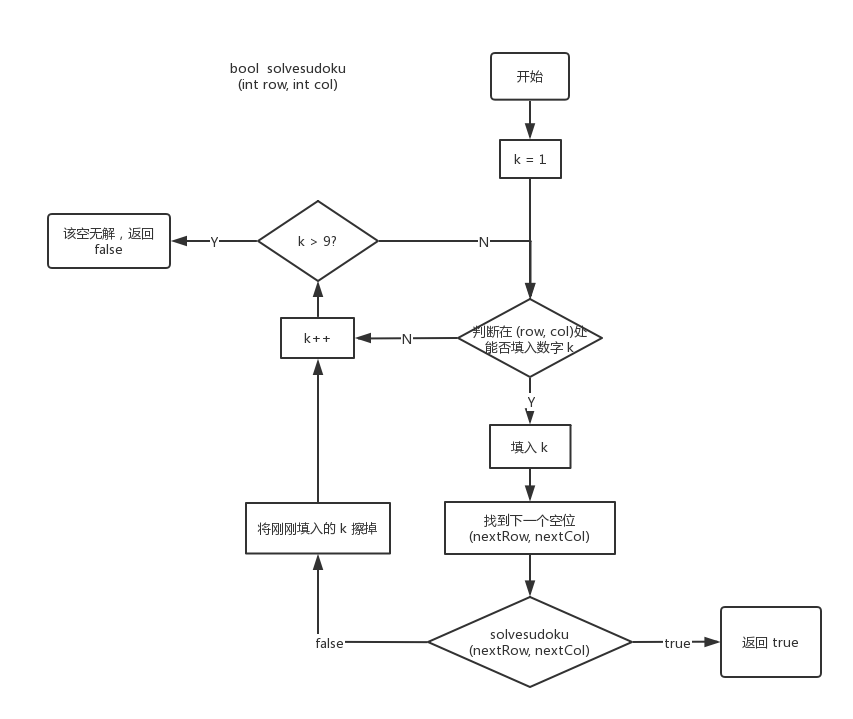
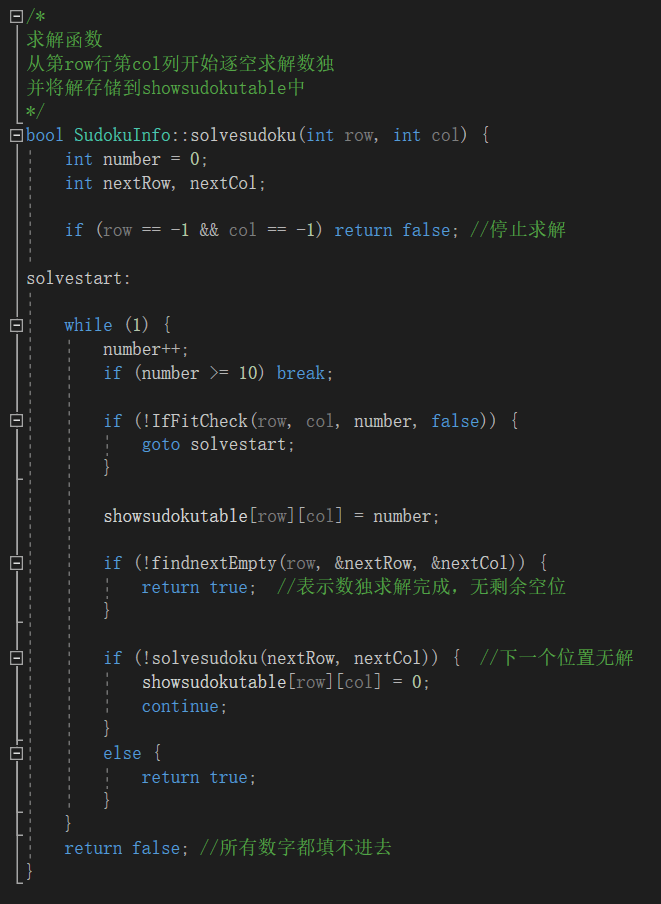
选择“普通数独”时程序将使用算法B求解。

**II.具体设计部分**

实验中部分功能点实现的设计思路如下：

**1.求解数独**

实验中采用了深度优先搜索的方法进行数独求解(高级算法在上文提过，此处仅涉及普通求解算法)。流程图及代码如下：

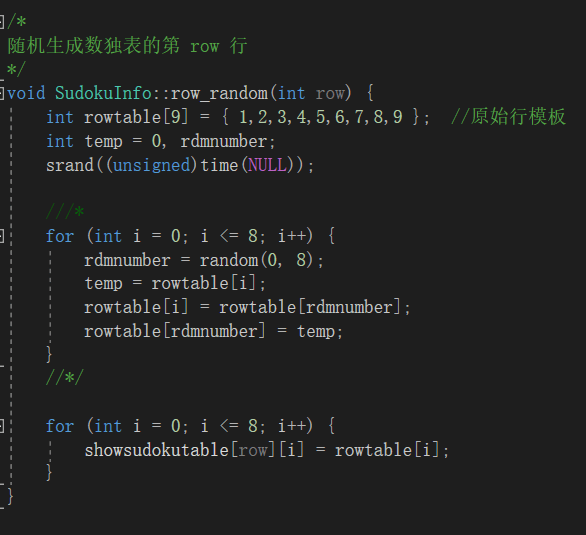
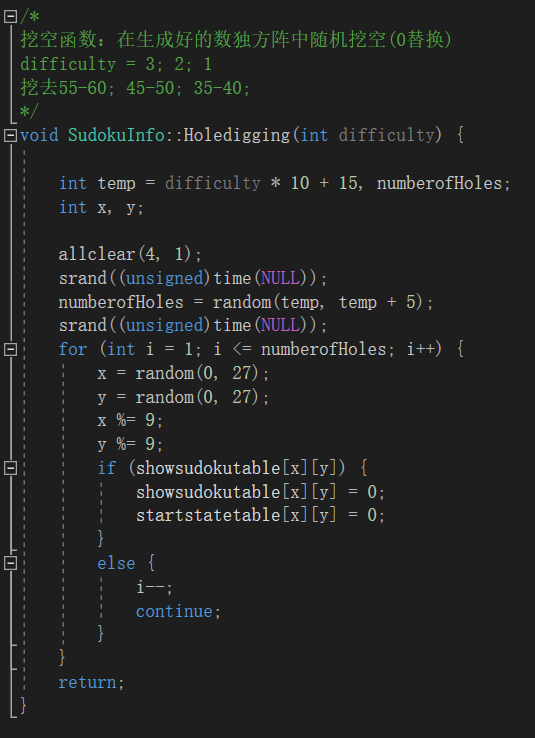
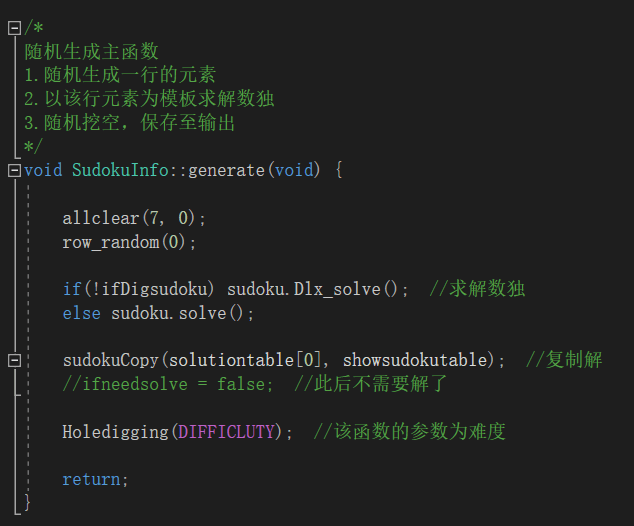


1. **生成数独**

实验中采用“先生成后挖空”的方式生成一个绝对有解数独。基本思路是：

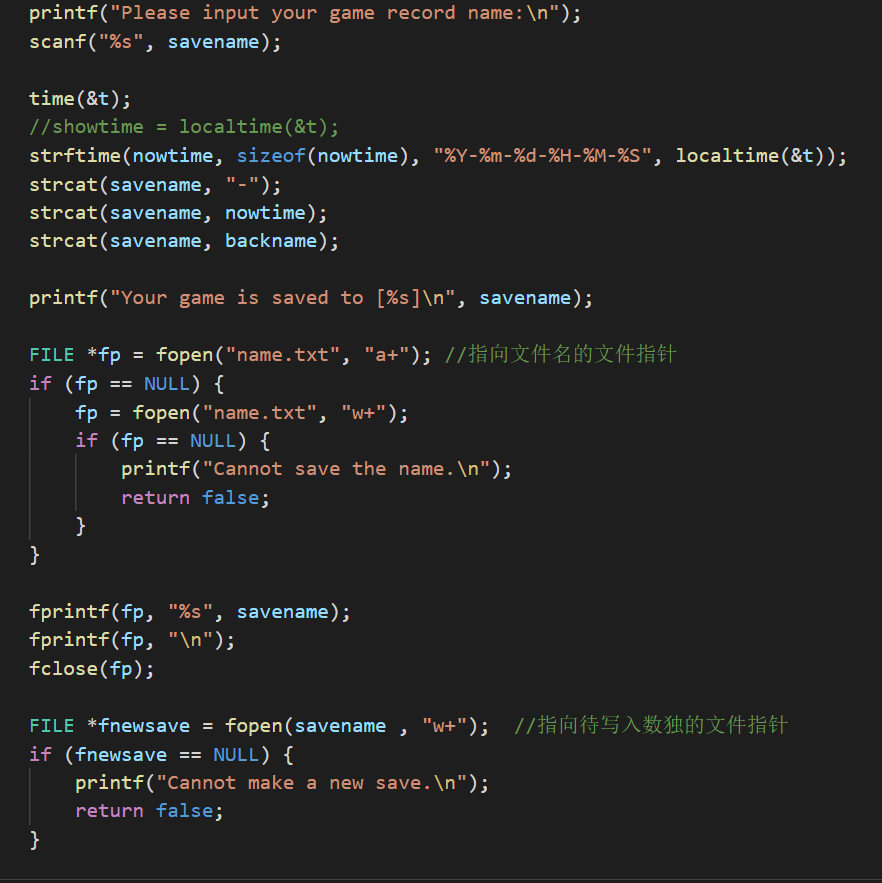
**随机生成一行乱序 1~9 的排列→以此为模板求解数独→随机挖空**

其中求解的部分调用的是上一功能介绍的函数。代码实现如下：

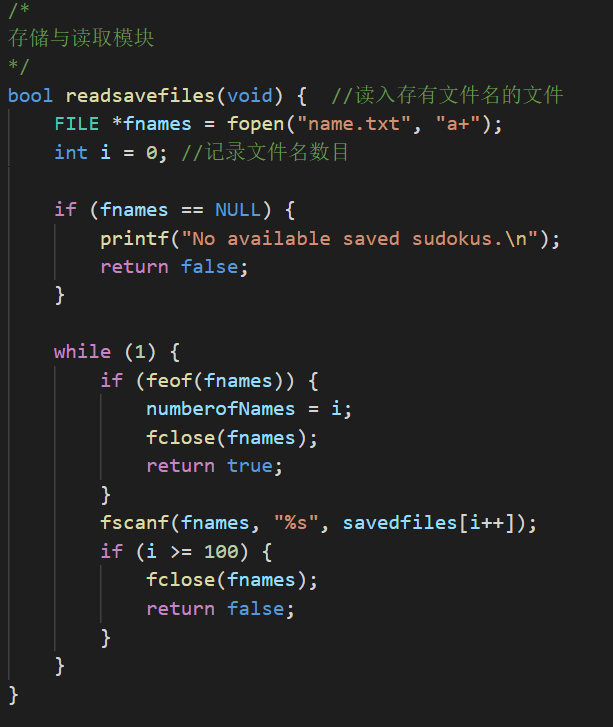
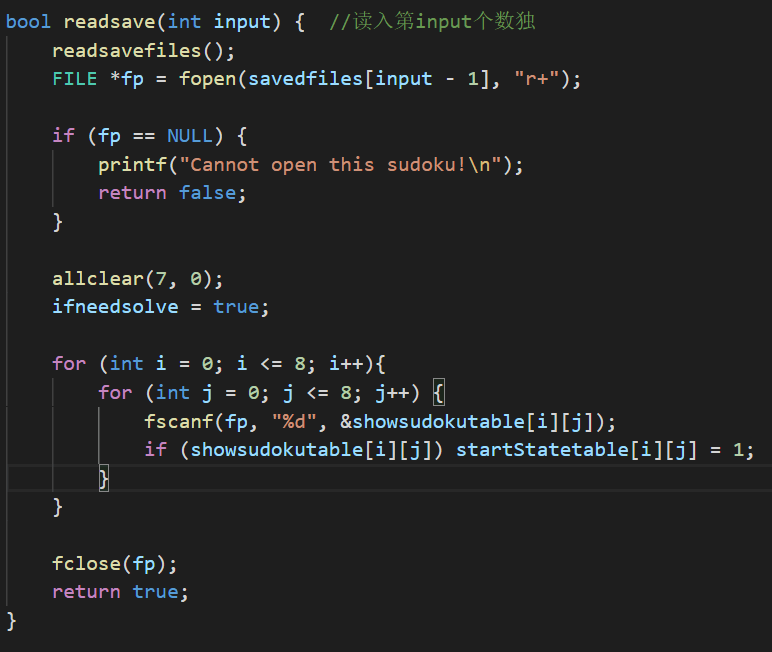


1. **存档与载入 (S/L)**

核心思路是文件读写。为了存储文件名，我特别增加了”name.txt”文件去存储每一个存档的名字。用户在输入存档名”savename”后会自动生成一个”savename-Year-Month-Day-Hour-Minute-Second.txt”的存档文件。具体代码实现如下：



载入存档时，程序会先去读取”name.txt”文件中的存档名，存储进入savefiles[]数组。之后再由用户选择读入。具体代码实现如下：



1. **提示**

基本思路是从当前数独的解中选择一个位置，填入现有的数独之中。每一次提示相当于求解当前的数独一次。

1. **图形化部分**

本次作业采用了EasyX图形库，基于C++在VS2017环境下实现的图形化。

具体函数介绍可参考EasyX文档：[EasyX 文档 - 基本说明](https://docs.easyx.cn/zh-cn/intro)

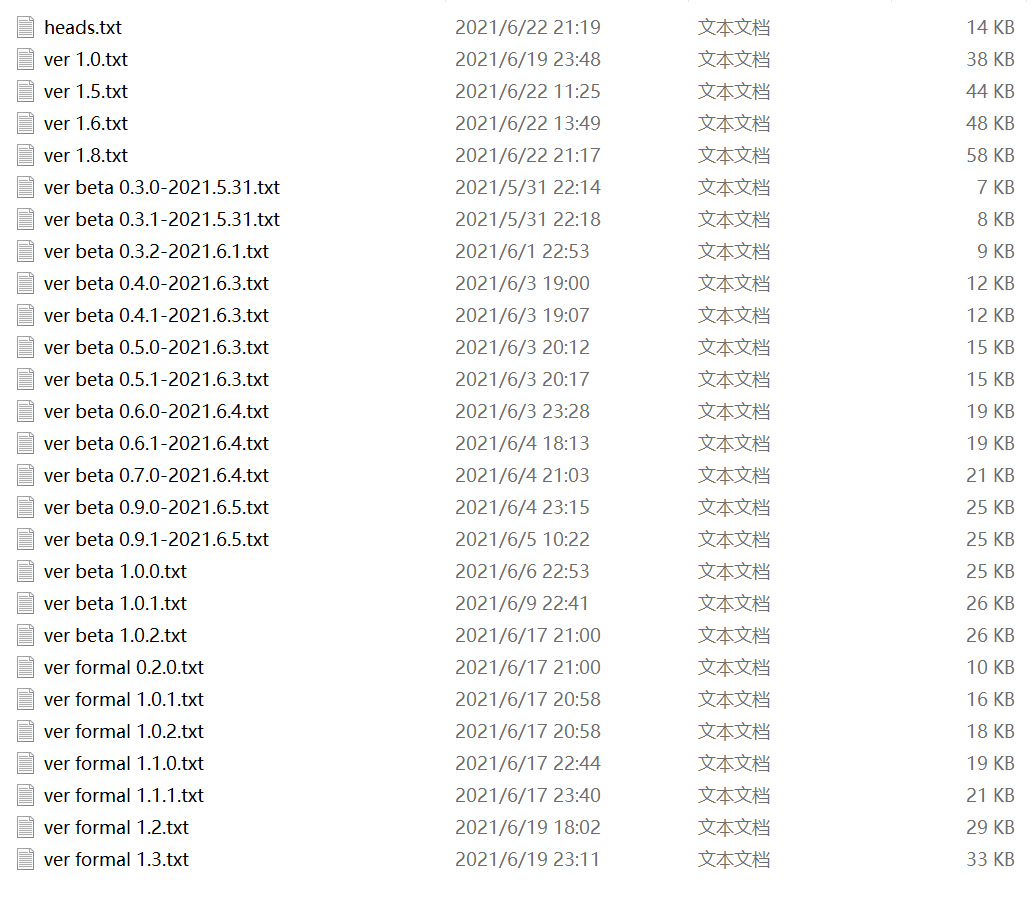
对于相关界面绘制，我采用了相对坐标法以便于后续删改；对于按钮功能，我采用了逐帧更新检测的方法进行判断。详细代码请参考代码文件。

1. **高级算法部分**

本实验中共有两种高级算法。其功能与代码介绍上文已做说明，此处不在赘述。

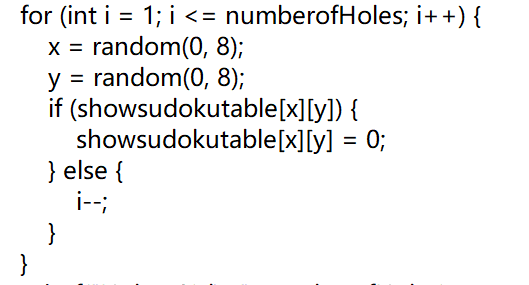
**III.实验过程**

整个实验历时两周，开发过程如下图所示。



其中，beta版本为命令行界面开发，formal版本为图形化界面开发。实验中遇到的主要bug简述如下：

[21060301]生成数独时出现死循环现象。

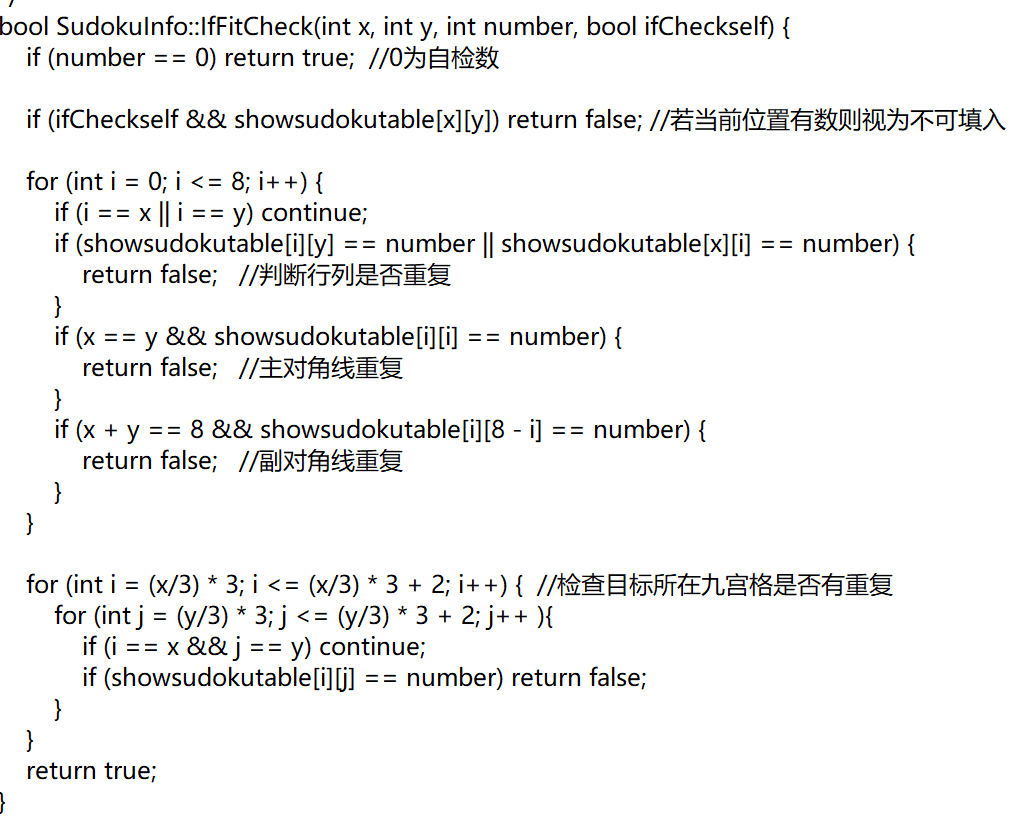
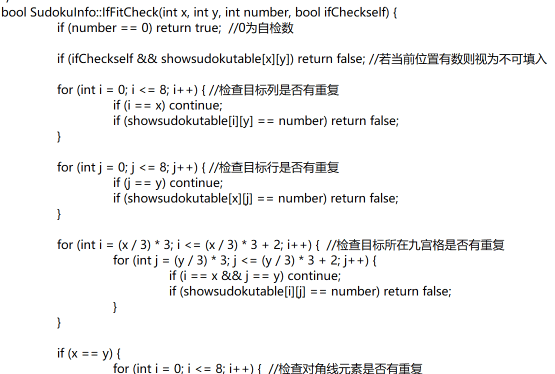
问题：对生成好的数独挖空时，程序采用了动态计数法。如果重复挖空会使计数变量自减，再次挖空。程序中使用了错误的循环判断方式，导致i一直自减，无法结束循环。

解决：进行如图所示进行更改。

[21061901]重构后生成数独无解

问题：发现在判断数字合法性时程序逻辑错误：对行列进行了同时判断，从而给出了错误的判断条件。

解决：行列分开进行判断。



[21062201]图形化载入功能异常，单击空白存档框时程序退出并报错[访问越界]

问题：在单击存档框后，由于未载入对应的存档文件名，此时的fp指向空白文件区域，从而访问异常。

解决：增加逻辑判定：如果该存档不存在则自动跳过存档打开步骤，直接返回主界面。

**IV.实验总结**

这一次的大作业是我第一次独立开发一个小型项目。相比于之前功能单一的程序来说，此次经历锻炼了我的逻辑思维能力，让我更加深入地去体会用户的使用逻辑与程序稳定的重要性。此外，我还=掌握了新的图形化开发技能与高级搜索算法，进一步提升了我的综合技能。