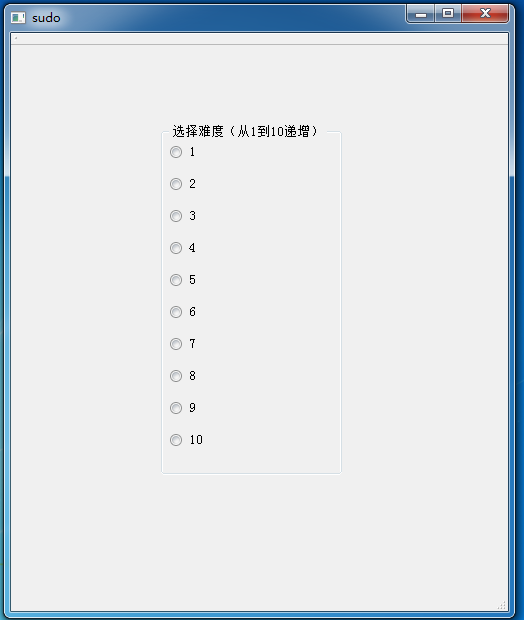
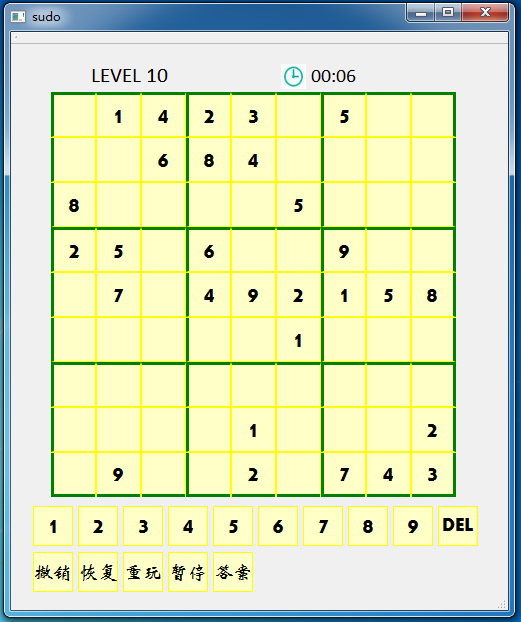
# Qt大作业——数独设计方案

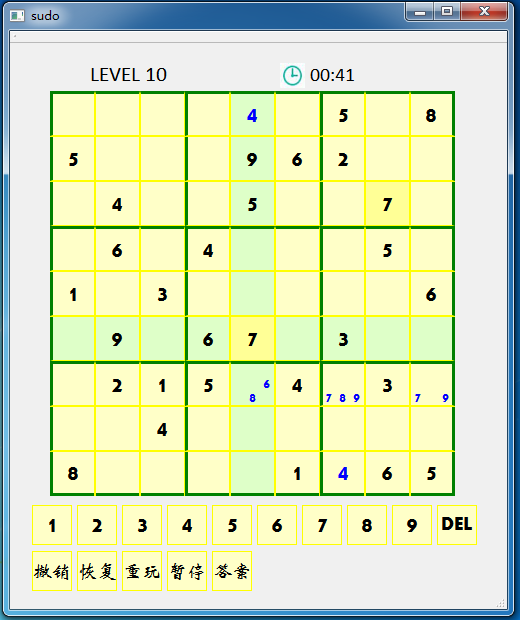
1. 用户界面以及操作

打开可执行性文件后，先有一个关卡选择的画面，如下



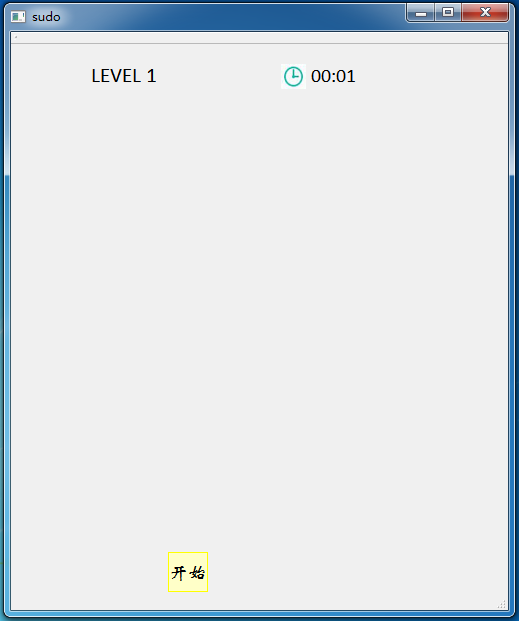
任选一关之后，进入游戏界面，如下

操作填数的方法为，先点击画面中的一格，再点击下面的1~9和Del进行填数操作，点击1~9的效果为：如原格中无该数则加上该数，有该数则消去，支持一格多数功能；点击del为删去格中所有数。黑色数为原有，蓝色数为自己填入。程序对于点击格所在行列以及相同数字的格（对格中有多数的格无效）有高亮效果，如下。



下方的几个功能按钮，撤销和恢复即如题意。如果撤销多步后有除了恢复外的其他操作，则再次撤销时以此操作后的界面为初始状态，即只可恢复致此处。

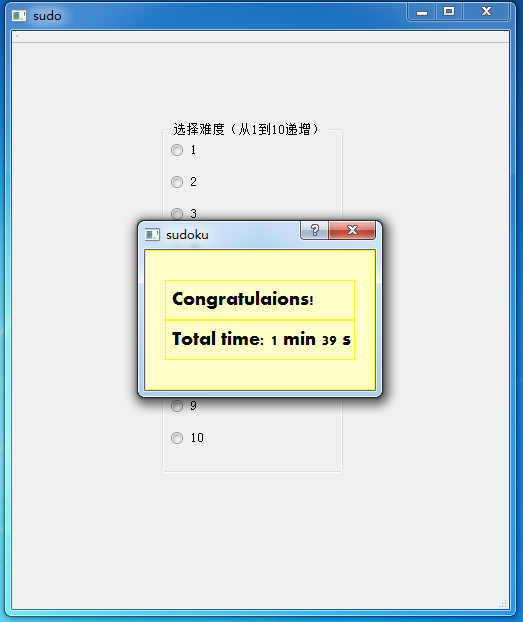
暂停时，所有按钮均被隐藏，时间停止，如下图



再按开始则恢复到暂停前的页面。

“答案”按钮在按下后，会显示出此则数独的答案，再次按下则恢复到玩家自己之前填的状态

当玩家手动填完所有空且为正确答案时，跳出Congratulations的窗口并恢复到选择关卡的页面，可重新开始，如下。



大致操作即如此。

二、数独生成器和求解器

求解数独采用了计64翁家翌同学分享的RDSS算法（搜索优化），对于给定数独，能够高效出解。

生成数独方法：随机生成一个1~9的全排列作为第一行，然后先用RDSS求出一个可行的前三行，然后再以前三行为基础求出一个可行的数独终盘，即作为此终盘。

然后依据所选择的难度，从终盘上随机删去25+3\*x个数，每删一个数都用RDSS检验解是否唯一，如唯一则继续删数。删够数为止。

三、心路历程

其实并没有什么心路历程，但有一点确实想说，在dubug的时候出现的一些问题。给程序加上一个功能，多了一个QPushButton之后，程序出现了各种玄学问题，诸如QSignalMapper的信号mapped(int)报错，明明没有触发SIGNAL，也没有触发SLOT函数，但SLOT函数里写的内容就是被执行了，该hide的窗口不hide，该show的窗口不show，简直像厄尔尼诺一样。Debug了接近2h+后，发现是一个QPushButton的数组开小了，导致空间溢出，所以才引起了各种无法用常理解释的现象。这提示我们，当程序出现玄学问题的时候，首先应检查一下是否有空间溢出。

四、关于

特别鸣谢计64翁家翌同学提供高效求解数独的RDSS算法。