

Project Report - Week 1

Sudoku Lite

陈晟祺

2016010981

计算机科学与技术系 计 63 班

1 项目简介

本项目作为计算机系《程序设计基础》第一周的大作业编写，是一款简单的数独游戏。本项目的图形界面部分参照了 https://github.com/ShenHongFei/Sudoku-Qt5.7.0-by_shfGroup 的设计，并使用了该项目中的图片资源，在此表示感谢。逻辑部分则为本人重新编写。

- 作为作业基本要求，我实现的功能有：
 1. 计时、暂停、重玩
 2. 方格内多个数字的填写、清除
 3. 各种操作的撤销、恢复（限制在 10 步以内）
 4. 内置 4 个难度层级，共 40 个不同的高质量关卡
 5. 相同数字、相同行列、相同宫的高亮显示（可切换开关）
- 在作业的要求之外，我实现的额外功能有：
 1. 随机数独题生成器，可切换三种难度
 2. 使用 Dancing Links 算法实现的高效数独求解器
 3. 提示按钮，每次可提示一个填错的数字，或填充一个空格
 4. 多语言切换，目前制作了中文（简体）与英语（美国）两种语言的翻译
- 此外，我还开发了下列的增强特性：
 1. 适配高分屏，在高 DPI 的显示器上进行恰当的缩放，保证可用性
 2. 使用 SQLite 数据库保存内置数独，实现状态保存与恢复、存储最佳时间等记录
 3. 使用 Qt Style Sheet 进行布局的调整与美化，方便切换各种自定义主题

2 程序截图

下面是部分程序运行时的截图。

简要说明一下软件的界面。详见图1，此为随机生成模式。其中最上方为模式、难度切换区域，以及解题计时和剩余未填的方块计数。下面为工具栏区，从左到右依次为新建游

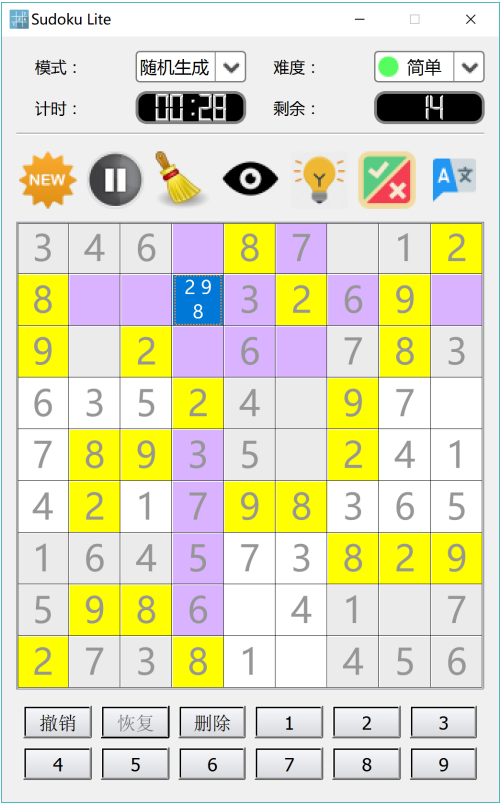


图 1: 随机生成模式

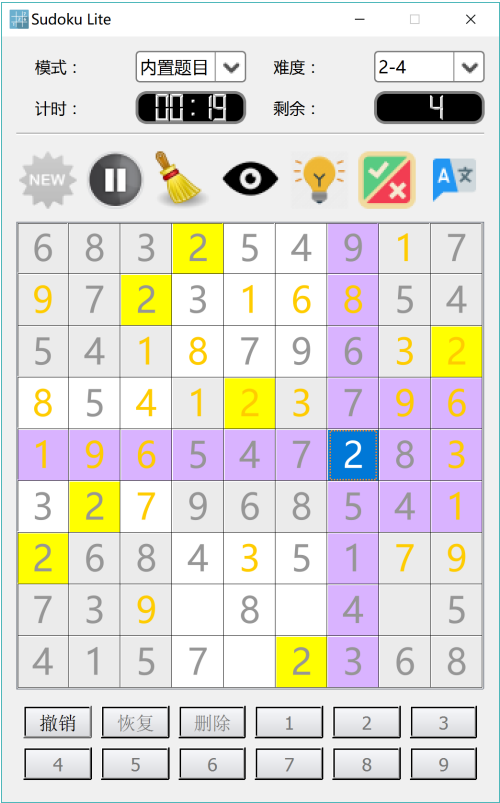


图 2: 内置游戏模式

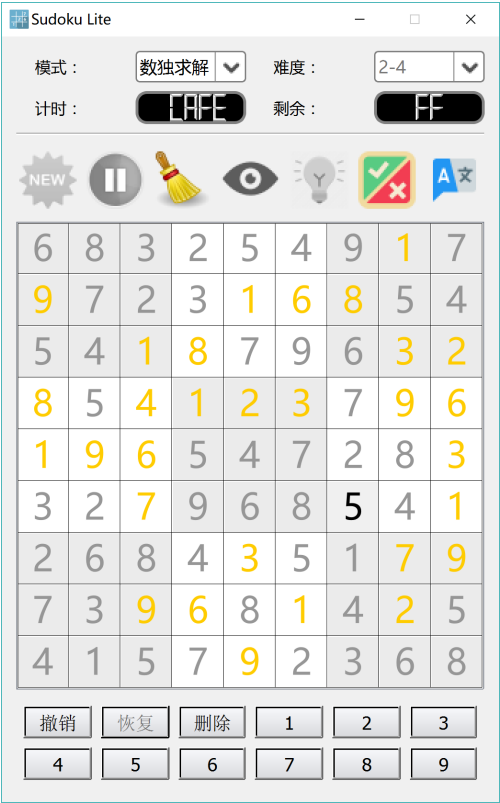


图 3: 解数独模式

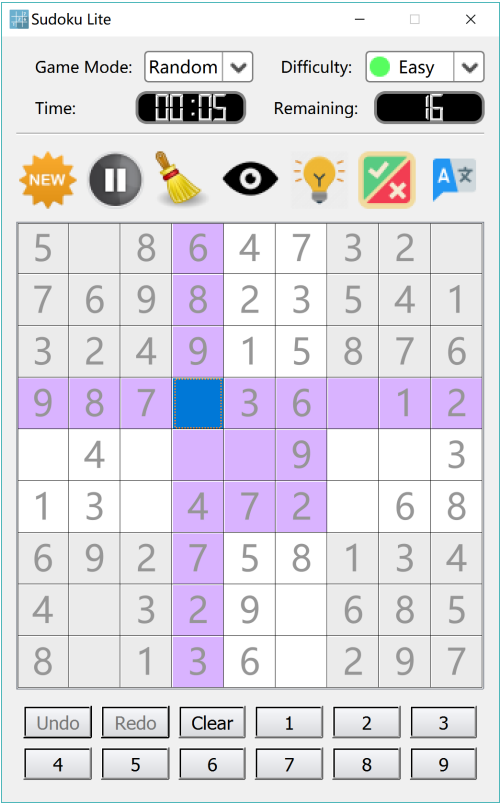


图 4: 英语界面

戏、暂停游戏、重新开始（清除所有已填数字）、开关高亮功能、显示提示、检查答案、切换语言。中间为数独区域，如果高亮功能打开，与当前数字处于同一行、列、九宫的数字背景将变为浅紫色，而所有相同数字的背景将变为黄色。

图2中是内置游戏，此时界面与图1大致相同，但新游戏按钮被禁用。难度切换变为内部的关卡选择，从 1-1 到 4-10，共有 40 关。此图中还演示了提示功能，每次使用提示，将把一个错误填写的数字变为红色，或者填写一个空格（图中的土黄色数字）。

图3为问题求解，此时仅有清空、解答功能可用，计数器与计时器均被禁用。如果有解，将会以提示的颜色填满数字；如果无解，将给出提示。

对于其他按钮，撤销、恢复将在有相应操作进行时动态地被启用或禁用，数字按键只有在空白中才有用，并最多可填入四个候选数字。使用暂停、检查答案功能都会触发弹出一个遮罩层盖住数独本身，避免对下层数字的操作，关闭后方可继续进行。如果答案正确，所有数字将变成绿色。此外，图4为程序的英语版本。

3 模块详述

下面简要介绍一下项目各个文件的功能或作用。

main.cpp 程序主入口。第一次启动时将内置的数据库资源文件释放到（平台无关的）应用程序数据目录下以便读写。初始化数据库连接，并初始化 **MainWindow** 类。

mainwindow.{cpp, h, ui} **MainWindow** 类，游戏主窗口。负责除数独内容外各部分控件的控制、事件处理。管理游戏模式切换、难度切换、语言切换。负责内置游戏的读取、记录的写入、程序关闭时的状态保存和启动时的上一次状态恢复等数据库 I/O 操作。

tableio.{cpp, h} **TableIO** 类，数独 UI 的管理。实际负责 **MainWindow** 中 **QTableWidget** 对象（即数独内容）在各种操作（如点击数字按钮、删除按钮、重新启动、开关高亮显示、请求提示）后的响应与重绘。同时内含 **Data** 类对象指针，作为当前的题目。

data.{cpp, h} **Data** 类，数独底层实现。可根据给定空格数量生成数独题（以及其解答），以及解答特定的数独（或给出无解提示）。采用的核心算法为 Dancing Links，即将数独转化为精准覆盖问题求解，速度较快，一般题目能在 20ms 内得到答案。数独的生成算法为：先随机填充一个九宫格，后用解题方法生成终盘，再根据难度进行挖空。

exactcovermatrix.{cpp, h} **ExactCoverMatrix** 类，用于精确覆盖问题的求解。来源于 <http://philoscience.iteye.com/blog/1537004>，并进行了 bug 修复与改动，在此表示感谢。

internalsudoku.{cpp, h} **InternalSudoku** 类，程序内部的数独表示格式，含题目、解、用户输入、最佳时间（如果有）等数据。用于 **MainWindow** 与 **TableIO** 间的通信，以及数据库和程序间的数据传递。

timer{cpp, h} **Timer** 类，对 **QTimer** 类的包装。提供程序的计时功能，以及对程序中 **QLCDNumber** 的控制。

changecellcommand{cpp, h} **ChangeCellCommand** 类，继承于 **QUndoCommand** 类，用于程序中撤销、恢复功能的实现。重写 **void undo()**，**void redo()** 并存放每个方格变化前后的状态（字体大小、文字颜色、文字内容、背景颜色），以供 **QUndoStack** 在进行撤销、恢复操作时进行调用。

common.h 附加于所有编译单元上的预编译头。主要用于给 MSVC 编译器提供指令，以改变可执行文件的运行编码为 UTF-8，避免乱码发生。

style.qss MainWindow 美化所用的 Qt Style Sheet 样式文件。

sudoku.db 程序内置的数独数据库。将释放到程序数据目录下以供读写。

res/*.png 程序中所用图片资源，用于工具栏按钮、程序图标等。

lang/*.qm, ts 程序 i18n 所用语言文件，含中文（简体）和英语（美国）。

4 设计理念

在本项目的开发中，我秉持以下的理念：

- 尽量遵循 Google C++ Code Style，命名规范，风格完善
- 多使用 Qt 提供的基础设施，实现完美的跨平台运行
- 拒绝硬编码，将常量、字符串、资源等集中管理
- 坚持 OOP 思想，明确区分各功能模块职责，降低各部分的耦合

5 项目感想

这是我第一个独立开发的项目。规模虽小，但对我利用信号实现事件驱动编程有了很大的启发，也给了我很好的练习 Qt 的机会。很遗憾的是，由于时间原因，我未能在项目中完整地使用经典的 MVC 架构。但总的来说，我在这一周时间中收获很大，希望接下来我能有更大的收获。最后，十分感谢老师以及助教的帮助。