

数独游戏设计文档

郑少锴

计65

2016011381

CONTENTS

数独游戏设计文档	1
功能	3
架构	4
数据结构与算法	5
资源	7
感受	8

功能

概述

本项目实现了一个带有基于Qt GUI界面的数独游戏。

基本功能

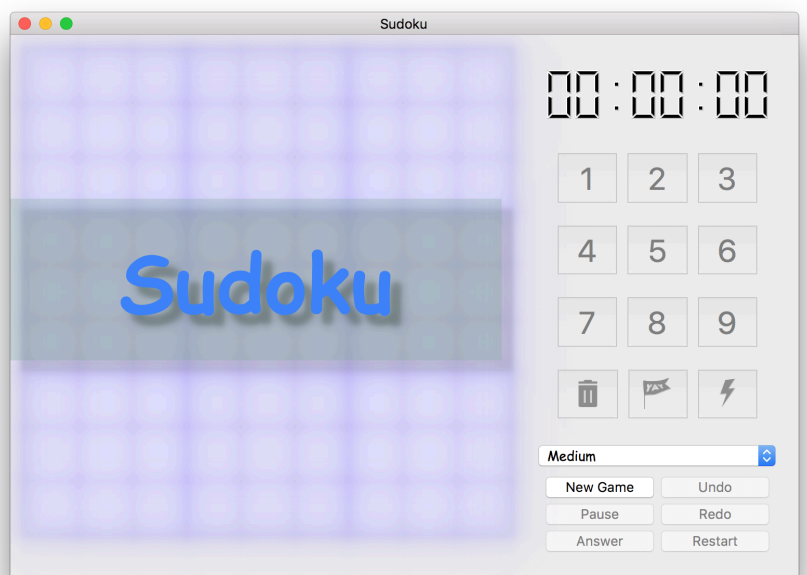
1. 9×9 标准数独的显示与操作；
2. 游戏计时；
3. 关卡难度选择；
4. 暂停与继续；
5. 撤销与重做；
6. 游戏进程控制等。

辅助功能

1. 方格内可填数字的提示；
2. 方格标记；
3. 相同数字方格的高亮显示。

附加项

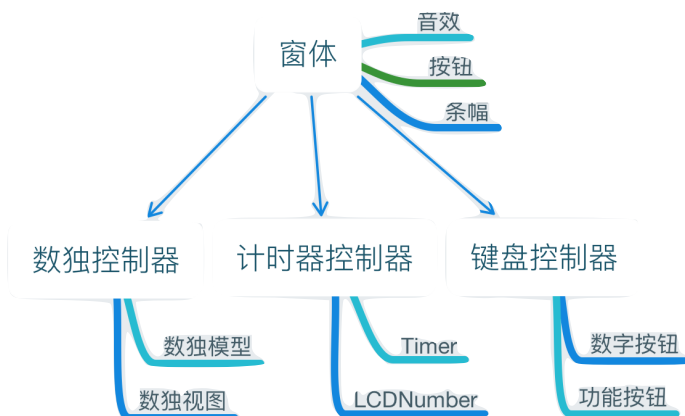
1. 用户友好的提示信息；
2. 游戏进程的条幅展示；
3. 游戏初始、暂停、胜利时的模糊特效；
4. 各难度数独的生成与求解；
5. 运算线程分离保证界面响应性；
6. 游戏音效。



架构

概述

游戏设计时采用了MVC模式，实现了界面和数据的分离。窗体MVC由一个Window类作为控制器，以下设有条幅视图、计时器控制器、键盘控制器，数独MVC等。



条幅视图

BannerView类，继承自QWidget，用于显示游戏初始、暂停和胜利画面。使用QGraphicsDropShadowEffect产生文字和边框的阴影效果。

计时器控制器

StopwatchController类。使用QTime作为模型，QLCDNumber作为显示。包含开始计时、暂停计时、重置计时等功能。用于在游戏中实时显示时间。

键盘控制器

KeyboardController类，继承自QWidget。用于控制数字按钮和清除、标记、提示等功能。可根据游戏状态和所选方格自动控制按钮的可用性和是否选中。

数独控制器

SudokuController类，继承自QWidget。使用SudokuView类对象作为视图，Sudoku类对象作为模型。

其中，SudokuView类继承自QWidget，实现了数独界面的绘制和鼠标和键盘事件的监视；Sudoku类定义了数独的数据结构和基本操作。

其他

使用一组QMediaPlayer对象实现游戏音效的控制，QComboBox实现难度的选取，以及一组功能按钮实现对游戏进程的控制。

数据结构与算法

数独模型

实现为Sudoku类。用一个记录了数字等内容的结构体存储每个方格，用一个 9×9 的二维数组记录整个盘面。为了提高运行效率，数字记录利用了压位的方法。

Sudoku类对外提供一系列方便的操作接口，其中包括：

1. 将一个或多个数字加入某个格子；
2. 从某个格子中移除一个或多个数字；
3. 翻转某个各种中的一个或多个数字；
4. 获取某个格子中所有可填入的数字；
5. 清除某个格之中的所有数字；
6. 设置某个格子中的所有数字；
7. 获取某个格子中所有冲突数字；
8. 判断某个格子中是否包含某一数字；
9. 获取某个格子中已经填入的所有数字；
10. 获取某个格中已填入的数字个数；
11. 读取或翻转某个格子的标记；
12. 读取某个格子的可写入性；
13. 判断游戏结束；
14. 重置游戏；
15. 一个基于函数式编程思想的forEachCell()接口，基于函数对象提供对各个格子的快速遍历操作。

数独生成器

实现为SudokuSolver类。该类为一个静态类，包含载入、随机生成和空白生成三种数独生成操作。其中随机生成算法可通过控制所生成数独的线索数在一定程度上控制难度。

其中随机生成算法流程如下：

1. 遍历81个格子：
 - a. 从当前不冲突的数字中任选一个填入当前格子；
 - b. 如果没有数字可选则清除盘面并回到1；
2. 获得数字序列[0, 81)的一个随机排列；
3. 根据所需难度，保留盘面上与序列最前部分相对应格中数字；
4. 清除其余数字。

在开发所用的计算机上，生成单个17线索数独的平均时间在10毫秒左右。

数独解算器

实现为SudokuSolver类。该类为一个静态类，包含一个使用经过优化的DFS算法求解数独的函数。

算法流程如下：

1. 如果盘面上存在一个可以确定需填数字的格子：
 - a. 在该格中填入此数字；
 - b. 如果游戏结束则返回解答成功，否则回到1。
2. 找到第一个尚未填数字的格子：
 - a. 如果该格没有可选数字，返回求解失败；
 - b. 在该格中依次填入一个可选数字，递归执行1：
 - i. 如果求解成功，则返回求解成功；
 - ii. 如果已经尝试所有可选数字，则返回求解失败。

在开发所用的计算机上，求解单个17线索数独的平均时间在5毫秒左右。

资源

图片

应用图标使用LogoScopic应用创建，键盘视图中的清除、标记与提示按钮的图标来自Create应用。

音效

游戏中使用的开场、胜利与失败音效均来自欢乐斗地主游戏。

感受

虽然在这么短的时间内要独立完成这么一个项目还是有点工作量的，但是是感触和收获也是蛮多的。

首先是关于Qt这个框架。说实话，原本我对Qt的“哲学”是略有抵触的；相较于Qt的略显臃肿的大而全，我甚至更加喜欢自己“造轮子”。所以说，其实第一次听说要学Qt的时候，我是拒绝的，因为我不愿意写完了之后再带一堆dll之类的上去，这样用户出来一定会骂我，根本没有这样写程序的，就证明上面那个是不好的。后来我也经过证实Qt确实还是蛮好用的，我大概写了三天左右，感觉还不错，后来我在写的时候发现我造过的一堆轮子在Qt里面都有更好的。我想我用完之后是这个样子，大家用完之后也会是这个样子！

其次是关于程序的更多思考。除了Qt这个框架里面connect之类的函数的神奇的实现以外，我还有了机会尝试用MVC架构写一个完整的游戏，也在实现过程中尝试了函数式编程的一些思想。能把理论付诸实践真好。

最后是能有机会听到各位同学做的课堂报告。不同的同学带来的不同的思维方式，不同的工具、工作流都能带来前所未有的思维碰撞。有时候我觉得已是必然的实现，同学们常常能讲解出更多更加优雅、高效的做法。也感谢老师和助教们的付出，讲解的知识让我受益匪浅。

总之，Qt这套理论远比我想象的要优秀；程序设计训练这门课程也远比我预计的要有趣而深刻；而我也将更期待在后面两周中所要迎接的更加精彩纷呈的程序设计之旅。
