国际象棋游戏程序设计文档

翁正洋　2021012588　未央水木-12
wengzy21@mails.tsinghua.edu.cn

2022 年 5 月 20 日

摘要

本程序为清华大学 2021-2022 春季学期计算机程序设计（20740073）大作业，指导老师为黄维通老师。本程序是一基于 C 语言、使用 Qt5.9.1 的编程环境与图形库的国际象棋游戏。本程序拥有完整的图形界面，支持本地双人对战，支持国际象棋中兵升变、吃过路兵、王车易位等特殊走法，支持自定义局时、步时等参数。

1 项目结构

本程序基于 Qt5.9.1 开发，以属于 QWidget 类的 mainscene 对象作为程序主界面，如图 1 所示。

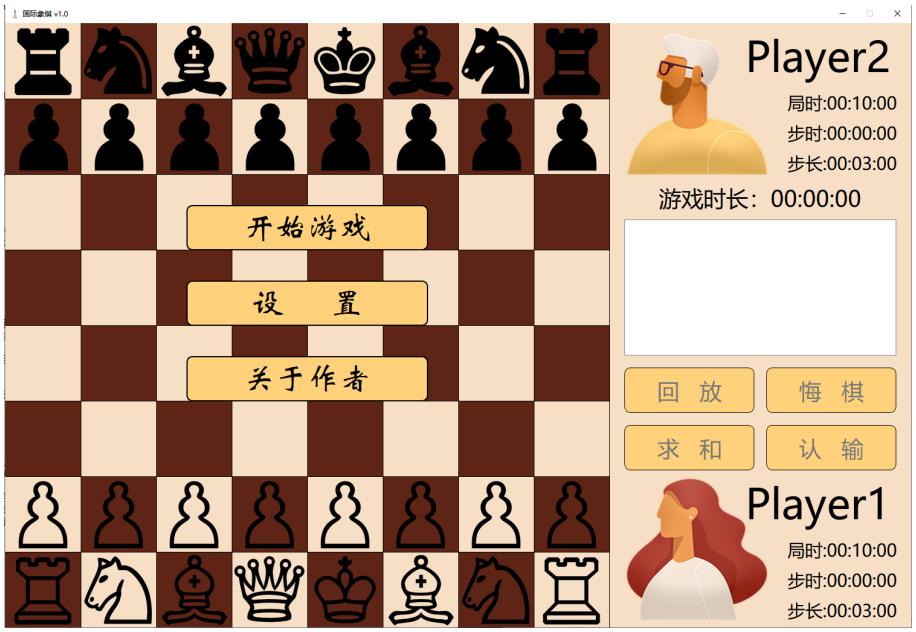

图 1: 主界面

具体的程序架构由图 2 所示。

1.1 mainscene.h,mainscene.cpp

本部分为程序主体，包含大多核心功能，包括 ui 的设计、人机交互的逻辑、棋子移动的逻辑等。

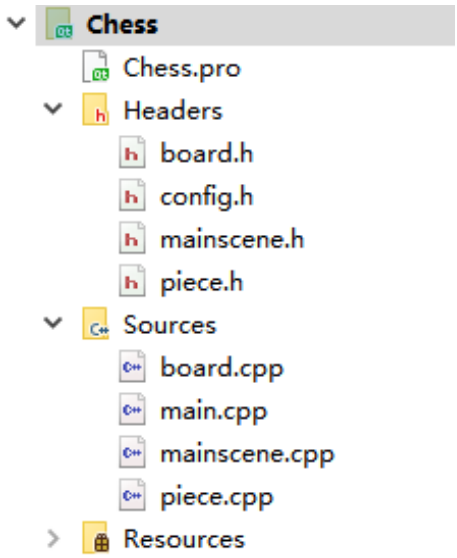

图 2: 程序架构

1.2 board.h,board.cpp

本部分定义了棋盘对象,主要负责加载棋盘资源,并提供至 mainscene.cpp 进行绘制。

1.3 piece.h,piece.cpp

本部分定义了棋子对象，定义了棋子的各个参数与各个接口，还负责棋子部分功能的的初始化。

1.4 config.h

本部分负责定义游戏运行的各参数，包括帧率、棋盘与 ui 几何尺寸、默认时长等。

1.5 Resources

本部分负责管理程序所需的素材，由 Qt 原生的.qrc 文件实现。

2 程序功能

本部分将简单介绍本程序实现的功能，以及具体实现的方式。

2.1 图形界面

本程序的图形界面基于 Qt5.9.1 的图形库搭建。具体而言，图形界面分为两部分。

2.1.1 ui

该部分包括程序运行过程中不实时刷新的部分，包括棋盘与棋盘右侧功能区中的全部内容，如图 3 所示。

图 3: 功能区 (红框中)

该部分的特点是不需要实时刷新，因此由 Qt 提供的各类图形控件实现，包括 $QLabel$、$QPushbutton$、$QPixmap$ 与 $QScrollArea$。

2.1.2 棋子

该部分包含所有的棋子。由于棋子移动的需要 (详见鼠标操作部分)，该部分需要实时刷新。

由于需要实时刷新，该部分由 Qt 自带的 *paintEvent* 函数与 *update* 函数实现。*paintEvent* 函数处理各棋子的坐标并进行绘制，由 *update* 函数更新到屏幕上。update 函数置于频率对应游戏帧率的计时器的槽中，从而以设定好的帧率刷新屏幕内容。

2.2 本地双人对战

本程序实现了本地双人对战功能。玩家轮流控制鼠标，操控己方棋子。

本程序确保了每回合仅有一方棋子可被操控。

2.3 鼠标操作

本程序支持使用鼠标拖拽、点击的方式进行操作，移动棋子。这部分的功能由 Qt 自带的 *mousePressEvent*、*mouseReleaseEvent* 与 *mouseMoveEvent* 函数实现，可以兼顾拖拽与点击两种操作方式。

2.3.1 选中棋子

本程序用 int 型变量 *selectPiece* 表示当前棋子的选中状态，并用 *getRowCol* 函数计算鼠标当前坐标对应的棋盘位置。

当前没有棋子被选中时，按压鼠标左键与单击鼠标左键都可以选中鼠标所在方格中的棋子。棋子一旦被选中，程序就会自动显示其可行区域，如图 4 所示。

2.3.2 拖拽移动

若棋子以按压鼠标左键的方式被选中，即选中结束后鼠标左键仍处于按压状态，玩家可以通过拖拽棋子并将其置于可行区域后释放鼠标左键的方式移动它。该过程中，*mouseMoveEvent* 函数会追踪鼠标位置，并使得棋子图案随着鼠标的移动而移动。

int 型变量 *dragChess* 表示了当前是否有棋子正在被拖拽。

图 4: 可行区域

2.3.3 点击移动

若棋子以单击鼠标左键的方式被选中，即选中结束后鼠标左键处于释放状态，玩家可以通过点击可行区域的方式移动它。

若使用点击的移动方式，棋子会以平滑的动画从起始点移向目标点。*Slide* 函数实现了这一动画。

2.3.4 取消选中

若玩家想要取消选中或更改选中的棋子，可以通过单击可行区域外的任意一点实现。

若玩家想要更换选中的棋子，可以直接单击想选中的棋子，便可更改选中对象。

2.4 行棋逻辑

本程序实现了国际象棋中所有的行棋逻辑，包括各棋子的基础走法，与兵升变、吃过路兵、王车易位等特殊走法。

本程序定义了结构体 *Route* 来存储棋子可行区域。*Route* 中包含了对应棋子可行区域的数量、位置与特殊走法。在此基础上，本程序定义了返回 *Route* 值的函数 *findRoute* 来为每一个棋子寻找可行区域，即确定走法。

2.4.1 *findRoute* 函数

该函数分为六个子函数 (*findRouteKing*,*findRouteQueen*,etc.)，分别对应国际象棋中的六种棋子。每个子函数都会根据输入的棋子做出判断，根据棋子类型对应的走法返回可行区域。

findRoute 函数的各子函数不仅会检索各棋子的基础走法，也会检索特殊走法。

值得注意的是，可行区域可以包括敌方棋子所在的区域。当棋子走到这些区域上时会消灭敌方棋子，即“吃子”。

2.4.2 棋子移动

当确定了棋子的可行区域后，程序会将其与从鼠标事件中得到的选中棋子的目标区域相比较，该过程由 bool 型函数 *checkRoute* 实现。如果目标区域属于选中棋子的可行区域，选中棋子就会通过 *moveChess* 函数移动到目标区域。

moveChess 函数确保了如果目标区域有敌方棋子，则敌方棋子会被消灭。

如果目标区域属于选中棋子的特殊走法，则 *checkRoute* 函数会在返回真值前，完成走法的特殊部分，如王车易位中车的移动，王的移动则将由 *moveChess* 函数执行。

2.4.3 兵升变

国际象棋中，本方任何一个兵直进达到对方底线时，即可升变为本方后、车、马、象中的一种棋子，不能不变。在本程序中，任何一个兵到达对方底线时，程序会提供弹窗，以供玩家选择兵升变棋子的种类。

该部分由函数 *Promotion* 实现。

兵升变过程中提供的弹窗不可关闭且置于模态，意味着玩家必须优先处理弹窗事件，弹窗存在时主界面被禁用。玩家必须选择一种兵升变的棋子，不能不变。

2.4.4 吃过路兵

国际象棋中，若对方的兵第一次行棋且直进两格，刚好形成本方有兵与其横向紧贴并列，则本方的兵可以立即斜进，把对方的兵吃掉，并视为一步棋。这个动作必须立刻进行，缓着后无效。

本程序中，若存在吃过路兵的合法走法，会以可行区域的形式显示。

2.4.5 王车易位

国际象棋中，每局双方各有一次机会，让王向车的方向移动两格，然后车越过王，放在与王相邻的一格上，作为王执行的一步棋。

王车易位还要求王与车均未移动过，王与车之间不存在别的任何棋子，王不能处于被将军的情况下，王不可穿越被敌方攻击的格。

本程序中，若存在王车易位的合法走法，会以王的可行区域的形式显示。

由于王的常规走法只能走与其相邻的格。因此选中王时，若存在距王两格的可行区域，则该可行区域代表王车易位的走法。

王车易位属于王的一种走法，因此选中车时，不会显示代表王车易位的可行区域。玩家仅可以通过选中王执行王车易位。

2.4.6 将军

下一部分中，“控制范围”一词代表可行区域。某子处于对方的控制范围内意味着，若该子不移动，下一步对方可以吃掉该子。

若一方行棋后，对方王处于该方的控制范围内，则对方被将军。此时游戏界面会显示“将军”，以提示对方应将。

该部分由函数 $Check$ 实现。

2.4.7 送将

若一方进行行棋后，本方王处于对方的控制范围内，下一步即可被吃，则称这种行为为送将。

本程序不允许送将，若一方有意或无意送将，游戏界面会显示“不能送将”，并将当前步无效化，继续由送将方行棋。

该部分由函数 $giceCheckMate$ 实现。

2.4.8 将杀

若一方行棋后，对方王处于该方的控制范围内，且对方没有方法使王脱困，则称该方将杀对方，该方获胜。

本程序通过遍历一方将军后对方的所有走法来判断是否将杀。若将杀，则本程序会在该方完成将军后直接宣布该方胜利，游戏结束。

该部分由函数 $CheckMate$ 实现，其中由 $ifCheckMate$ 函数来对应将方所有可能的走法进行遍历，判断是否将杀。

2.4.9 困毙

若一方行棋后，对方王未处于该方的控制范围内，但是对方下一回合无论如何行棋，都会使王暴露在该方的控制范围内，则称对方被困毙。

与中国象棋不同，国际象棋中，若一方被困毙，则双方和棋。

本程序通过遍历一方行棋后对方的所有走法来判断是否困毙。若困毙，则本程序会直接宣布和棋，游戏结束。

注意到检查是否困毙的逻辑与检查是否将杀的逻辑基本相同，因此该部分同样由函数 $ifCheckMate$ 进行遍历。

2.4.10 $newTurn$ 函数

以上所述的兵升变、将军、送将、将杀、困毙等特殊事件均在每回合结束后执行的 $newTurn$ 函数中进行判断，并进行各分支的处理。若新一回合能正常进行，$newTurn$ 函数会初始化新回合。

在玩家选中棋子并指定可行区域后，$newTurn$ 函数就会被调用。它假设玩家的走法合法，并在此基础上计算双方的控制范围。这部分由 $calControl$ 函数实现。

完成对控制范围的计算后，$newTurn$ 函数会逐一判断是否出现特殊事件。若送将、将杀、困毙这三种不合法或会直接结束游戏的分支不成立，$newTurn$ 函数会初始化新回合，完成计时、记谱等工作，并移交控制权。

2.5 回放

本程序支持通过功能区的回放按钮，回放上一步操作。

单击回放按钮后，上一步移动或被消灭的棋子会回到上一步的状态，并以平滑的动画重现上一步的操作。该动画速度相对较慢，以方便玩家观察。

该部分由按钮 *BtlReplay* 与对应的槽 *replay* 实现。

2.6 悔棋

本程序支持通过功能区的悔棋按钮，使执行上一回合的玩家悔棋。

单击悔棋按钮后，程序会提供弹窗，询问执行当前回合的玩家是否允许执行上一回合的玩家悔棋。若是，程序将自动回到上一回合状态，由上一回合玩家进行操作；若否，则游戏继续，当前回合玩家继续操作。

本程序不支持连续悔棋。即上一回合玩家请求悔棋并同意后，当前回合玩家不得再次请求悔棋以回到两回合之前。

该部分由按钮 *BtlRetract* 与对应的槽 *retract* 实现。

2.7 求和

本程序支持通过功能区的求和按钮，使当前回合的玩家请求和棋。

单击求和按钮后，程序会提供弹窗，询问非当前回合玩家是否同意和棋。若是，则游戏结束，双方平局；若否，则游戏继续，当前回合玩家继续操作。

该部分由按钮 *BtltoDraw* 与对应的槽 *toDraw* 实现。

2.8 认输

本程序支持通过功能区的认输按钮，使当前回合的玩家投子认负。

单击认输按钮后，程序会提供弹窗，询问当前回合玩家是否确认投降。若是，则游戏结束，非当前回合玩家胜利；若否，则游戏继续，当前回合玩家继续操作。

该部分由按钮 *BtlConcede* 与对应的槽 *concede* 实现。

2.9 记谱

本程序使用标准的代数记法进行记谱，在每一步完成之后实时记录于功能区中的记谱区中，如图 5 所示。

本程序通过 QSring 变量 *MoveText* 记录本回合中各操作的记谱记号，将其传递到 *freshHistory* 函数中更新记谱区显示的内容，从而实现每回合更新记谱。

图 5: 记谱区 (红框中)

2.10 计时

本程序参考各类棋类比赛，提供了丰富可调的计时系统，包括游戏时间、局时与步时。

本程序定义了专用的 QTimer 型计时器 *TimeTimer* 以实现计时功能。

2.10.1 游戏时长

游戏时长记录了自点击开始游戏按钮以来的时间，即棋局的总时长。

2.10.2 局时

局时指的是对战双方各自拥有的总行棋时长，最高可设置为 8 小时。在某方回合，玩家进行思考与行棋时，其对应的局时会减少。一方行棋完成后，回合会进行转换，由对方进行思考与行棋。

若一方局时减少至 0，则对方直接获胜。

每次轮到双方行棋，局时都会自动增加一定的时间。也可设置成局时不增加。

默认局时为 10 分钟，每步增加局时为 10 秒。

2.10.3 步时

步时指的是对战双方每步拥有的行棋时长。步时不可以超过规定的步长，否则对方直接胜利。

步时也可设置，默认步时为 3 分钟。

2.11 初始界面

打开程序后，玩家首先面对的是初始界面，如图 6 所示。

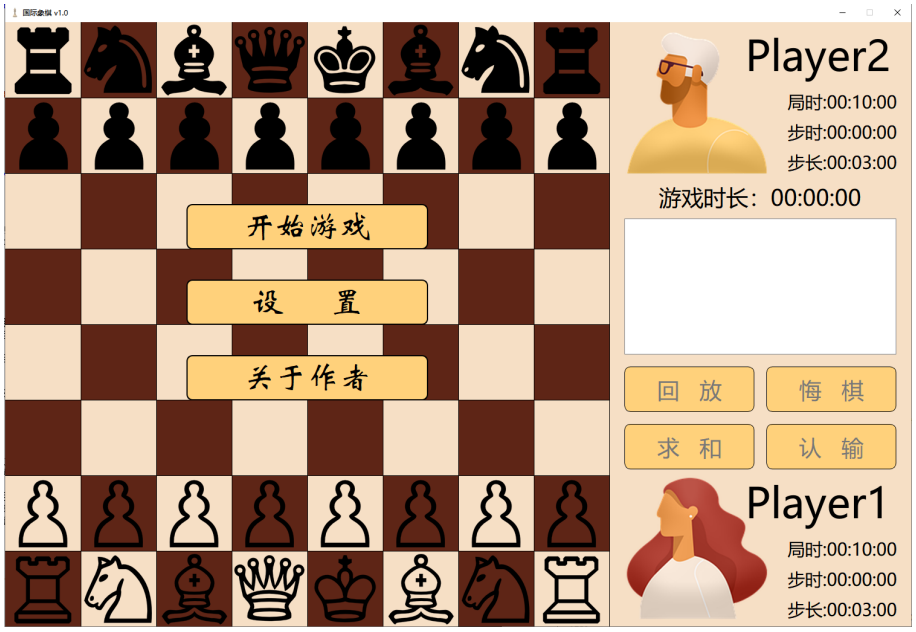

图 6: 初始界面

该界面中，只有棋盘中央的三个主菜单按钮可用，其余功能均被禁用。当玩家单机开始游戏按钮后，才会进入游戏界面，如图 7 所示。

2.12 设置

在棋局开始前，玩家可以通过主菜单中的设置按钮访问设置界面。在设置界面中，玩家可以设置本局游戏双方的局时、步长与每步增加局时，如图

图 7: 游戏界面

8 所示。

设置完成后，玩家需点击保存按钮方可退出。

2.13 玩家信息

本程序在功能区中显示玩家头像与名称。

2.14 结束界面

当有一方获得胜利或和棋后，游戏结束。

此时棋盘中央显示游戏结果，棋盘、棋子、功能区各功能均被禁用，如图 9 所示。

3 non-goals

本部分简要介绍本程序由于各种原因未实现的一些功能。


Player2
局时:00:10:00
步时:00:00:00
步长:00:03:00
游戏时长: 00:00:00
回 放
悔 棋
求 和
认 输
Player1
局时:00:10:00
步时:00:00:00
步长:00:03:00
Chess
局时:
600
秒 (1~28800秒)
步长:
180
秒 (1~28800秒)
每步增加局时:
10
秒 (0~600秒)
保 存

图 8: 设置


黑 棋 胜 利 !
Player2
局时:00:10:18
步时:00:00:01
步长:00:03:00
游戏时长: 00:00:05
1. f4 e5
2. g4 Qh4#
回 放
悔 棋
求 和
认 输
Player1
局时:00:10:17
步时:00:00:00
步长:00:03:00

图 9: 结束界面

3.1 帧率可调

考虑到本程序对硬件要求较小，绝大部分电脑可以流畅运行，以及国际象棋游戏本身对高帧率并无要求，本程序锁定 100 帧运行，不自动调整帧率也不提供调整帧率的接口。

3.2 分辨率可调

调整分辨率的功能会对 ui 设计造成较大影响，带来大量工作量。考虑到 ui 设计并非本课程的教学重点，本程序不提供分辨率可调的功能，锁定分辨率为 1500 × 1000。

3.3 玩家信息

由于本程序只支持线下双人对战，因此未实现账户与登录系统。相应地，玩家信息只显示默认信息，不能更改。

3.4 死局判断、三次重复

出于复杂度以及国际象棋知识的不足，本程序不支持死局判断与三次重复，需要玩家对战时由线下沟通的方式自行实施这些规则。

4 项目总结

本项目是作者入学以来第一次完成的从创建到封装的完整项目，代码量 3000 余行，用时三周 +。在整个过程中，相比最后的作品本身，学习的经验对我而言更加珍贵。

作为一名没有信息竞赛基础的高考生，在大学我经历的编程经验仅有浙江技术高考的学习。本门课程是我在大学的第一门计算机相关课程，在课堂上学到的许多知识令我受益匪浅。

在完成大作业的过程中，我则深刻认识到了自学的重要性。在大作业布置后，我自学了《C++ Primer Plus》第六版的全部内容，并选定了 Qt5 作为本次大作业的图形库，开始完成大作业。

在学习 Qt 的过程中，我阅读了《C++ GUI Qt4》编程第二版的部分章节，也学习了 Github、CSDN、B 站上的一些教程。不过，本项目的所有代码均由作者本人完成。

作为一个项目编写的新手，我在本项目中遇到了一些项目组织与代码规范方面的问题。我尽量按照多人开发的规范与标准组织项目、命名变量，也尽可能注意了内存占用等问题。不过由于经验及知识的匮乏，不免有缺漏之处。

在开发的过程中，我遇到过不少恶性 bug，并逐一完成了修复。最终的发布版本中无已知 bug，但是由于测试条件的匮乏与测试时间的不足，仍可能有 bug 存在，如有发现，望及时与作者联系。

祝游玩愉快!