网络国际象棋对战软件

计 84　2018011446　刘泓尊

2019.8

1 软件用途

国际象棋是一种二人对弈的棋类游戏。本软件基于 Qt 和 Socket，实现了双人网络国际象棋对战功能，并且具有残局保存、读入，智能化路径提示，王车易位，自动判断“逼和”等功能。

2 运行方式

本压缩包目录\app 中放有 Chess.exe 可执行程序，如果您的电脑上安装有 Q 相关动态链接库，程序可正常打开。

以上方法若不可行，请您安装 QtCreator5.13，将\SourceCode 中的 Chess.pro 文件导入 QtCreator，重新构建、运行。

3 功能介绍

3.1 图形界面

3.1.1 界面结构

如图 1 所示，本程序界面由三部分构成，上侧为**菜单栏**，用户可以点击进行相关操作。左侧为**功能栏**，用户可以在游戏开始前进行网络的连接、游戏的开始，在游戏过程中读档、存档、发送认输请求等。功能栏下方提供了当前走棋方，对用户简洁直观，还提供了**计时器**显示当前走棋的步时限制，下面的提示信息可以在游戏过程中实时提醒用户状态。界面右侧主体为 8x8 国际象棋棋盘。

图 1 网络国际象对战软件主界面

3.1.2 用户友好

界面设计直观简洁，用户可以在两步之内实现他们下棋所需要的任何功能。界面不同操作区之间设计了分栏，使得界面层次清晰，界面风格采用扁平化。同时，由于双人对弈需要进行连接，在连接成功之前，我将其他游戏相关按钮置为 unable，使得软件更加贴近用户的体验，引导用户尽快熟悉此软件。

3.2 双人网络对战（容错机制）

3.2.1 创建主机

本程序采用 TCPSocket 创建服务端与客户端的连接，同一个程序，既可以充当客户端，也可以充当服务端。充当服务端的一方可以选择“创建主机”，默认以 IPV4 地址显示，如果电脑未连接 WiFi，则以本机 IP 地址显示。图 2 为点击“创建主机”时弹出的窗口，选择“OK”可以打开主机的监听端口；在连接成功之前，点击 close 可以关闭监听端口。

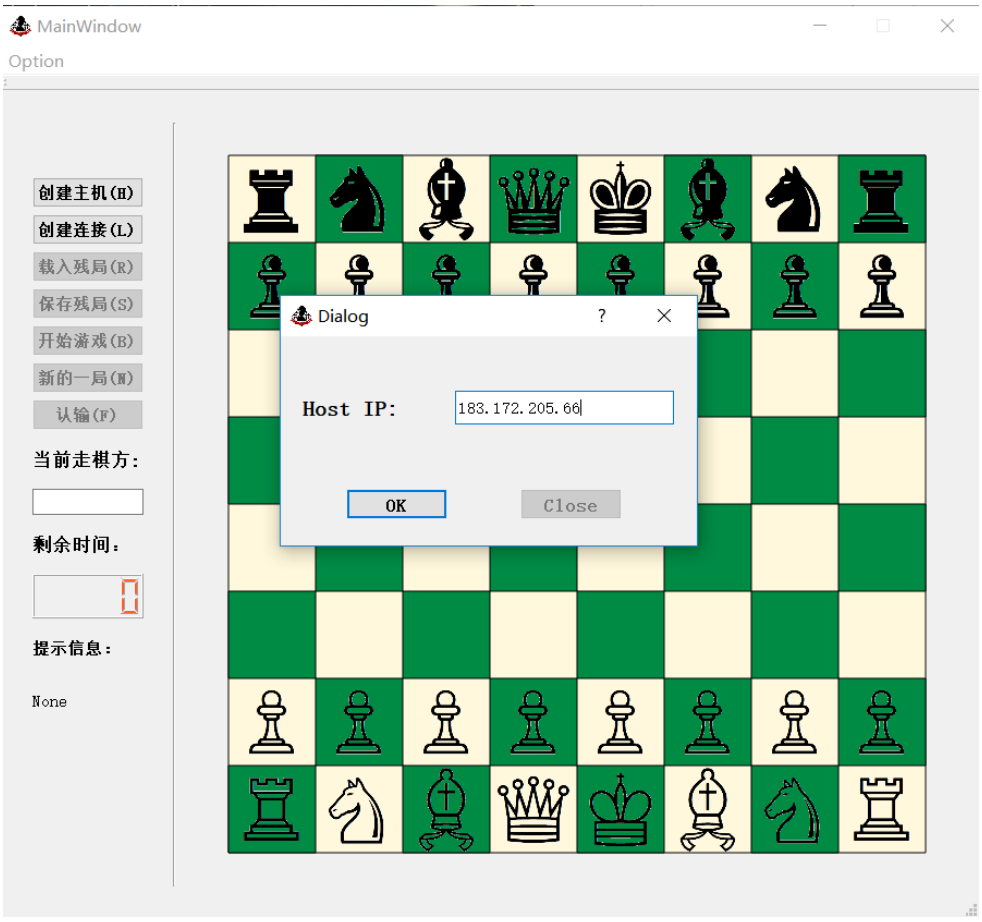


图 2 点击“创建主机”可以显示当前 IP，点击 OK 可开始监听

3.2.2 创建连接

充当客户端的一方可以选择“创建连接”，弹出窗口中同时支持软键盘输入和键盘输入（如图 3 所示）。点击“OK”可以尝试与主机进行连接。若连接等待时长超过 2.5s,将提示连接失败（如图 4 所示）。允许用户重新连接。连接成功后，将显示“连接成功”信息，默认主机端执白，客户端执黑。如果 IP 地址输入格式不正确，将提示错误，用户可重新输入（如图 5 所示）。

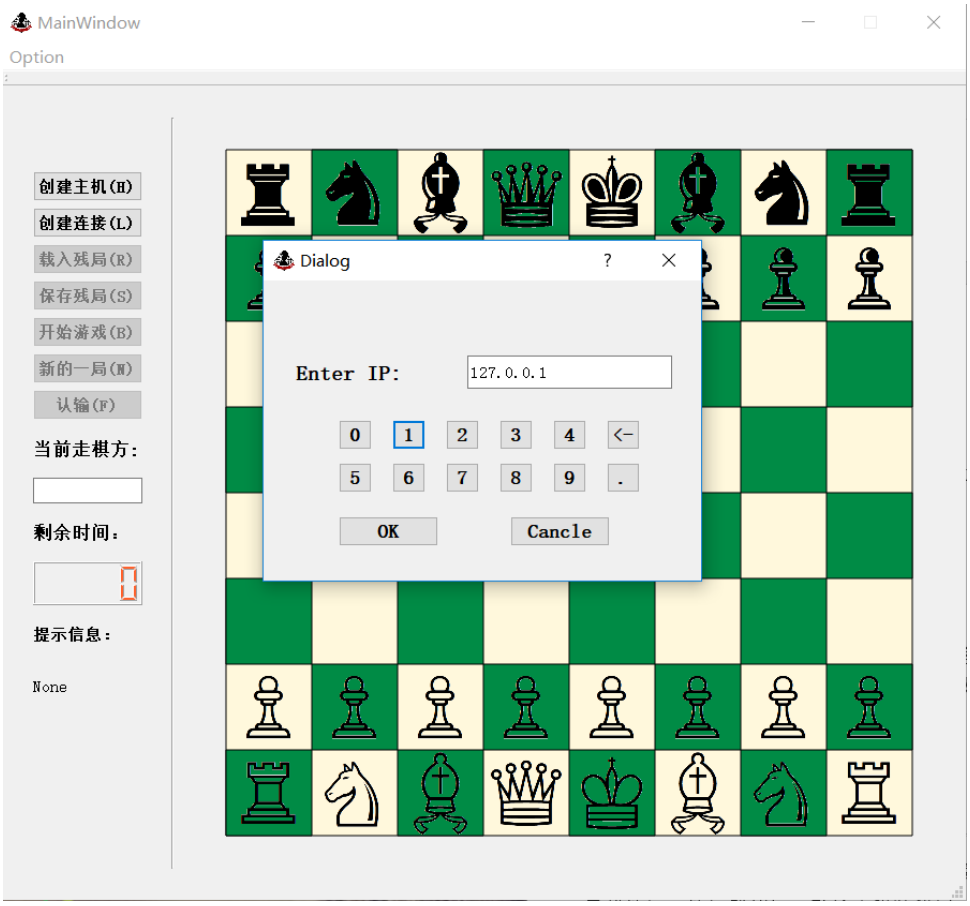


图 3 点击“创建连接”后弹出 IP 输入窗口

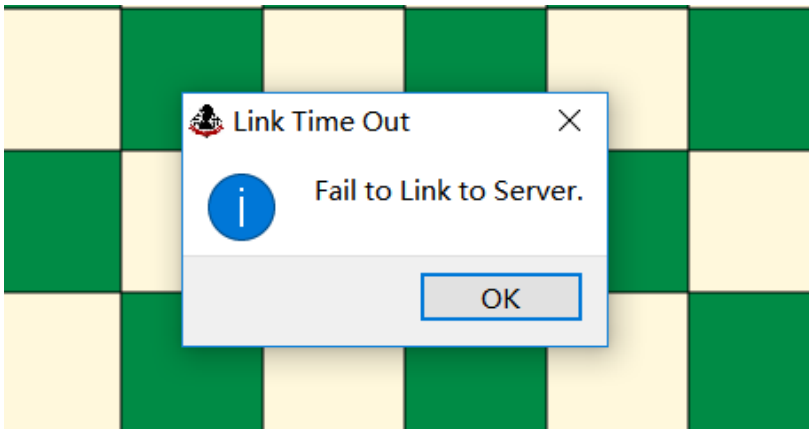


图 4 超时连接之后提示连接失败

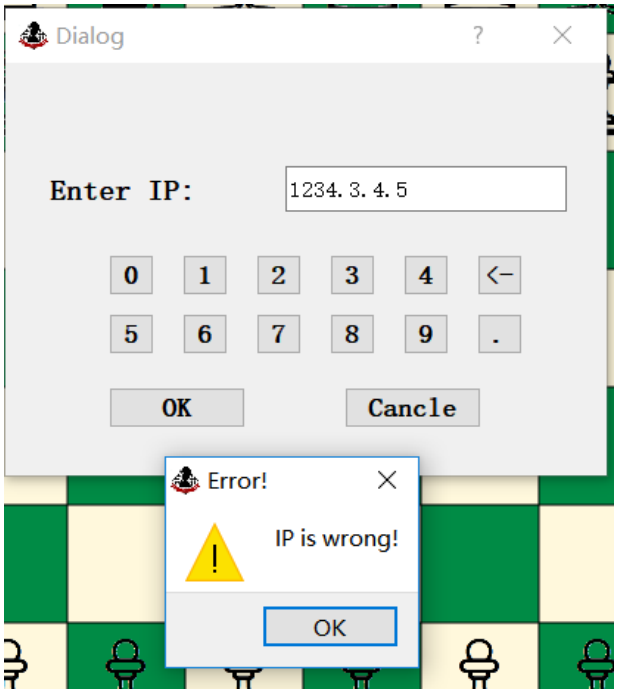


图 5 IP 格式不正确时，提示错误信息

3.3 规则判断

3.3.1 基本规则

本程序本着“用户友好”的原则，在用户走每一步棋时，在棋盘上标注它可走的路径。（如图 6 所示）当用户走到违反规则的位置上时，程序会将棋子退回原来位置，提示用户重新下子。同时，允许用户使用鼠标拖拽棋子，而不是使用点击的方式，使得程序的交互性更强，更加符合游戏开发的要求。

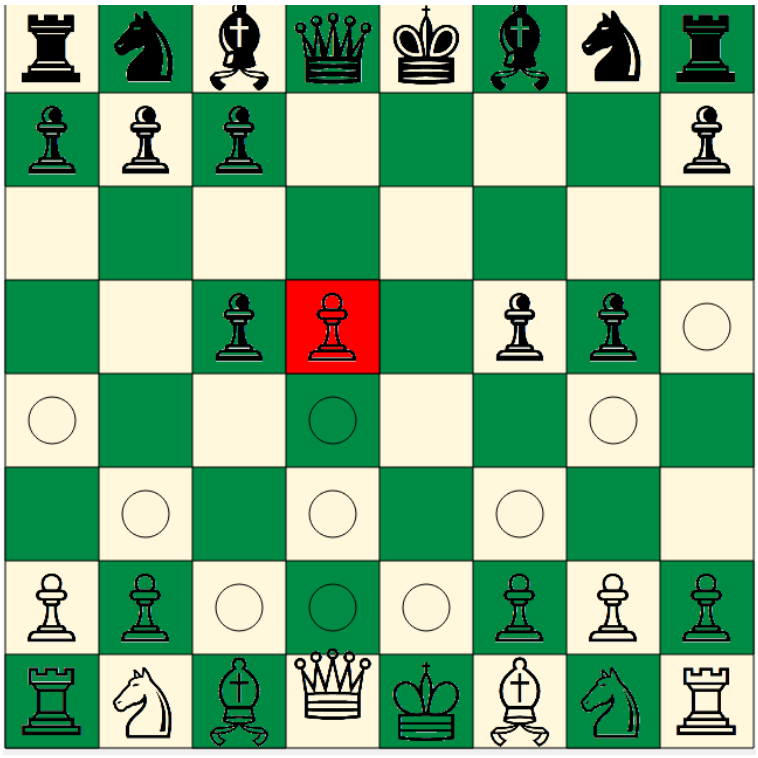


图 6 用户每走一步棋时，在棋盘上显示可走的路径

3.3.2 兵升变

根据国际象棋规则，当本方兵走到对方底线时，必须选择升变为除王之外的棋子，本程序自动检测兵的位置，当兵走到底线时，弹出选择窗口（如图 7 所示），用户可根据需要进行选择。

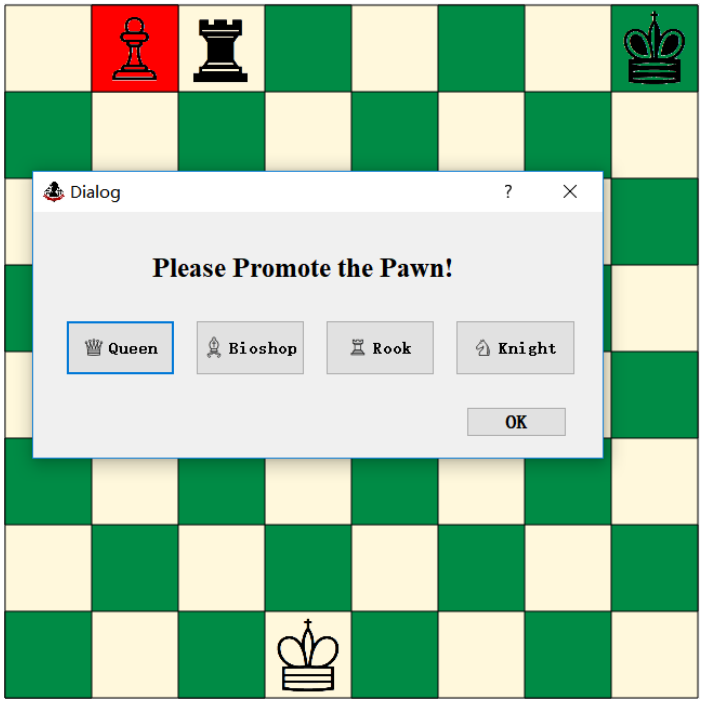


图 7 “兵升变”选择框

3.3.3 王车易位

在王、车都未移动过，王、车之间无间隔，且王所走路径上无攻击威胁时，可以进行“王车易位”。本程序在可以实现“王车易位”时，棋盘上将提示王可走两格（如图 8 所示），当检测到王自动向左或向右移动两格时，车将自动移至王移动的另一侧（如图 9 所示）。简化了用户操作流程。

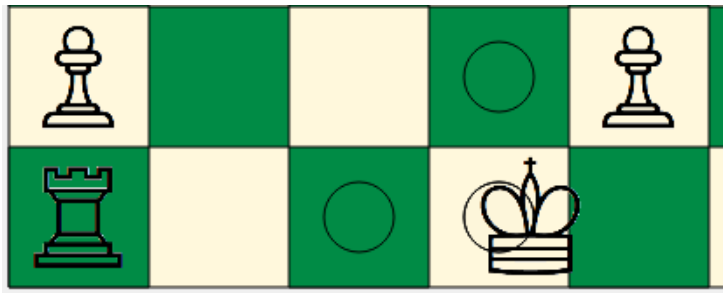


图 8 ‘’王车易位”前显示可走位置

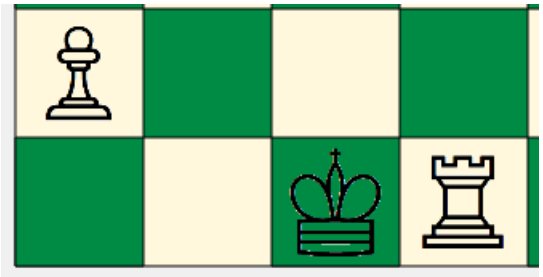


图 9“王车易位”后的棋局显示

3.3.4 逼和

在王未被将军但本方无子可走时（本软件不允许送吃），将形成“逼和”。软件将自动检测当前局面，“逼和”时输出相关信息（如图 10 所示），游戏结束。

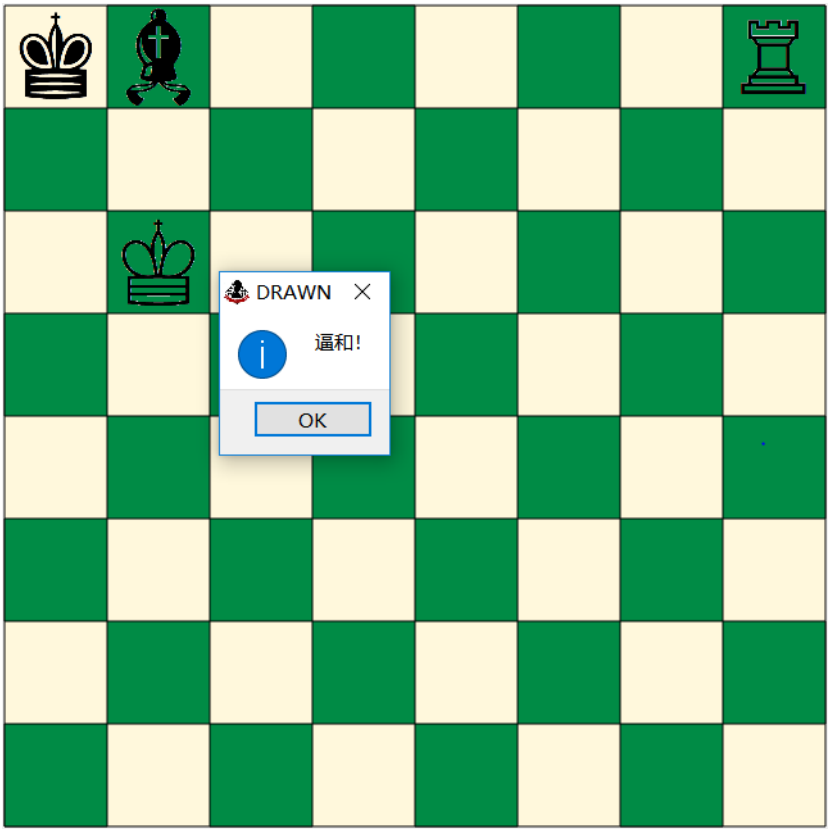


图 10“逼和”时输出相关信息

3.3.5 认输

在左侧功能栏有“认输”按钮，用户可根据需要点击“认输”。本着用户体验良好的原则，点击“认输”之后将提示用户进一步确认（如图 11 所示），之后将认输信息发送给另一玩家（如图 12 所示），游戏结束。

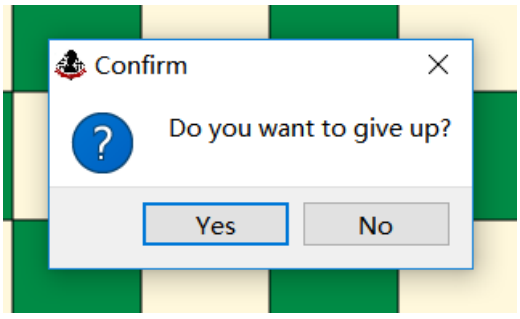


图 11 点击“认输”之后输出提示框

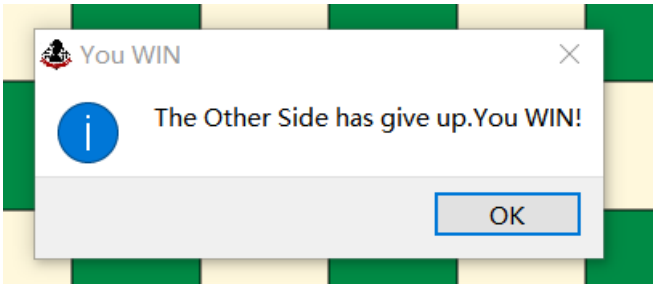


图 12 认输之后，对方收到提示信息

3.3.6 超时判负

本游戏设置步时限制为 20s，当用户走棋时间超过 20s 时，将输出提示信息，超时一方失败，并发送信息至另一方，显示另一方成功。（如图 13 所示）

图 13 “超时判负”功能提示框

3.4 残局保存与读入

左侧功能栏设有“载入残局”和“保存残局”功能，在游戏开始之后，用户可以点击“保存残局”，将当前文件存入硬盘。在游戏开始前可以“载入残局”，进行上次存档的游戏（如图 14 所示）。

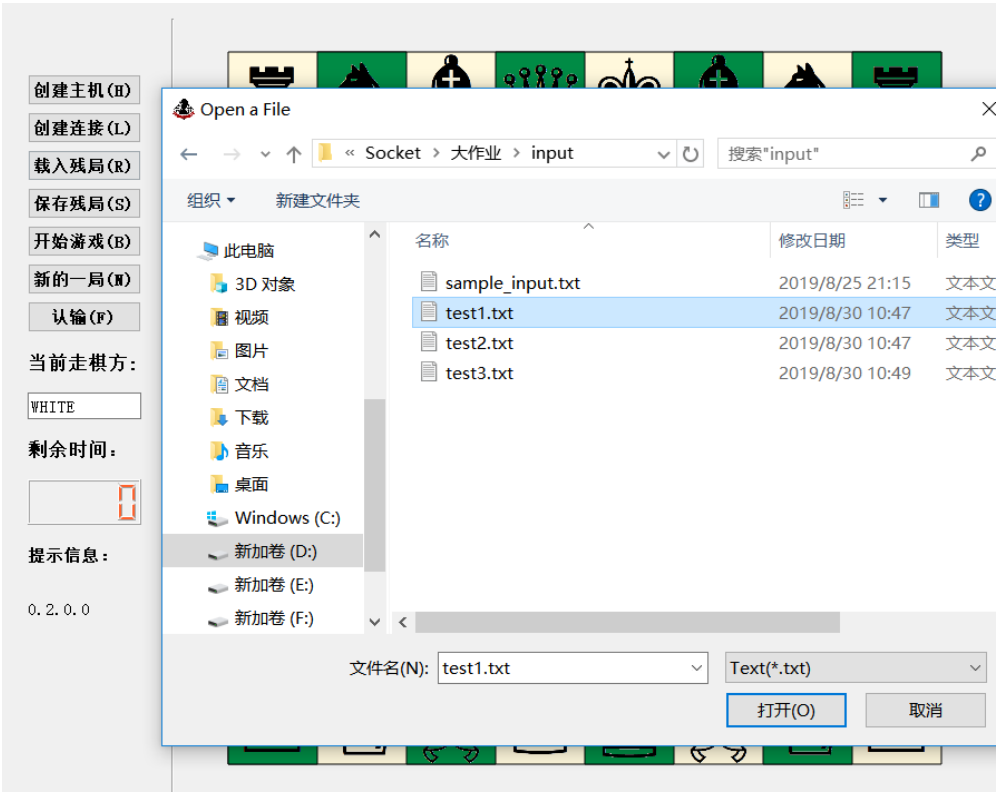


图 14 残局的读入与存储

3.5 扩展功能 1 – 起子可走路径提示

为了良好的用户体验，当用户用鼠标拖拽某一棋子时，棋盘上会显示相应的提示信息，提示用户当前棋子可走的位置，如果用户下子位置违反规则，则将该棋子自动放回原来位置。（如图 15 所示）

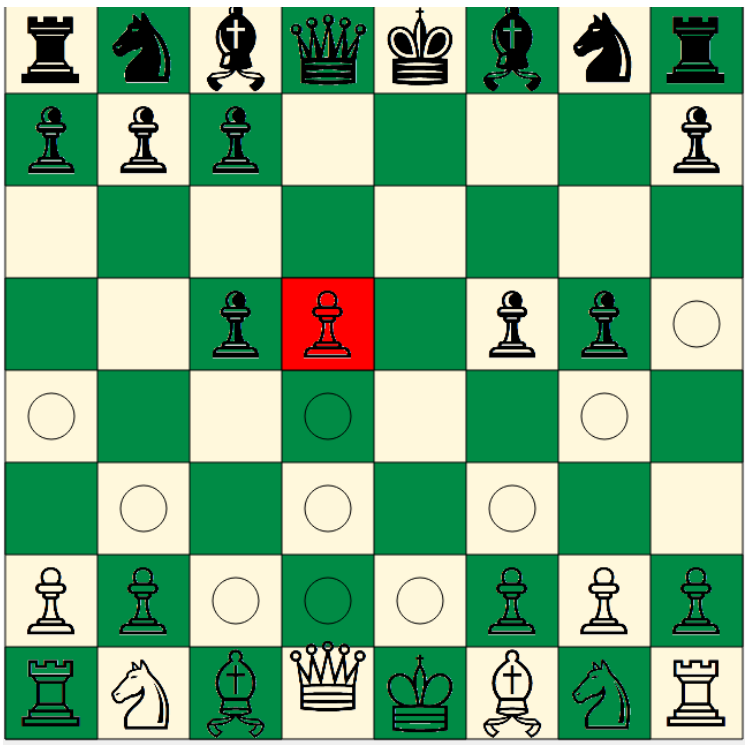


图 15 显示当前棋子可走路径

3.6 扩展功能 2 – 危险棋子高亮提示

在游戏过程中，当某一棋子处于地方棋子的攻击范围内时，该棋子所在格子将变为红色，以示该棋子处于危险之中（如图 16 所示），便于玩家采取措施。

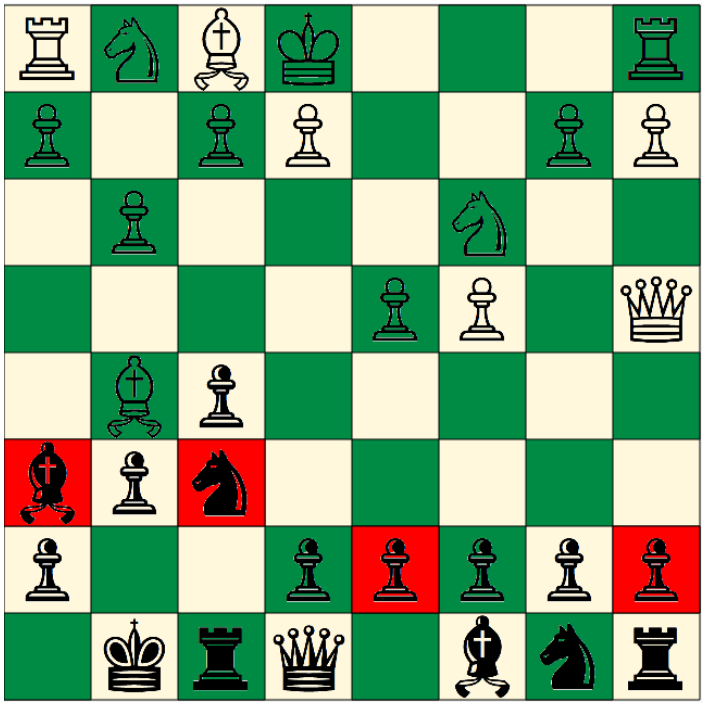


图 16 高亮显示本方处于危险状态的棋子

3.7 扩展功能 3 – “将军提示”

在游戏进行过程中，当本方被将军时，软件将弹出窗口提示用户被“将军”，以便用户“应将”(如图 17 所示)。

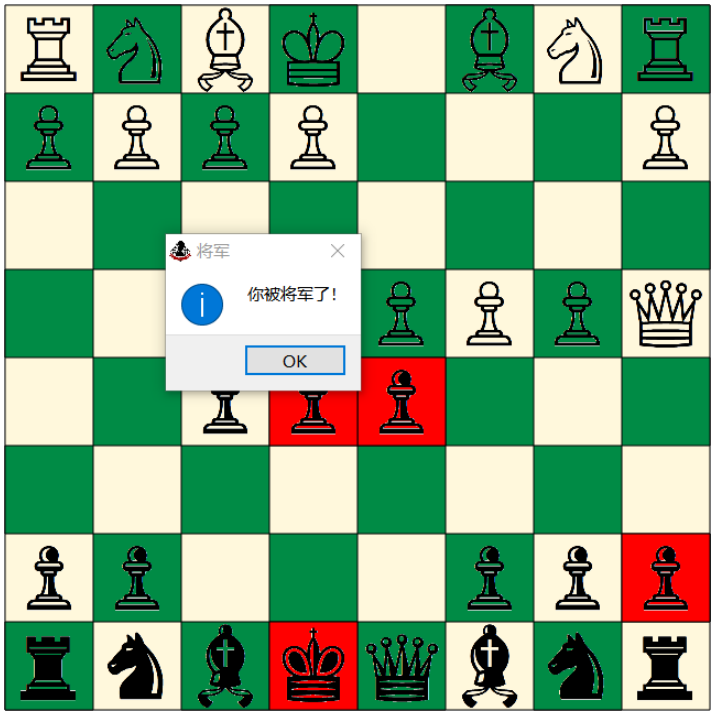


图 17 本方被“将军”时，输出提示信息

4 网络通信

4.1 客户端/服务端工作流程

客户端与服务端之间用 TCP 连接，在每次一方下完棋子，或者软件检测有某事件发生（如“逼和”）时，会自动将该信息以**“操作信息”**或**“棋子变动信息”**的形式传递给另一方。当一方用户按下某按钮触发某一事件时，该操作也会以“操作信息”的形式传递给另一方，使得两个玩家的信息可以在很短时间内就进行一次同步。

4.2 客户端/服务端通信协议

客户端和服务端通信采用了 TCP 协议，传输数据形式包括两种：**操作信息**（用英文单词表示，如“start”，“restart”等）；**棋子变动**（我为每个棋子属性进行了编号，通信时采用**“%d.%d.%d.%d.%d”**格式（如 1.1.1.5.5 表示编号为 1 的棋子（如白后）从(1,1)移到了(5,5)），表示某一编号的棋子从某位置移动到某位置）。

4.3 处理“粘包”问题

由于传输的数据都具有定长或者固定的英语单词，因此本程序处理“粘包”问题十分简便，只需要先确定是“操作信息”还是“棋子变动信息”，然后根据信息的长度对信息进行截取，便可以获得正确的信息。

4.4 网络通信编程框架

本程序使用 TCP/IP 协议,传输数据由“操作信息”和“棋子变动”两者构成，前者传递英文单词字符串，后者传递特定格式的代表整数的字符串，表示棋子变动信息。

5 代码简述

5.1 数据存储

主要数据有：

Matrix[10][10]:用于存储棋盘上的棋子，不同的棋子用不同的编号表示（如图 18 所示）。

Walkable[10][10]:用于存储当前正在移动的棋子可以到达的点，便于“可行路径提示”和“规则判断”。

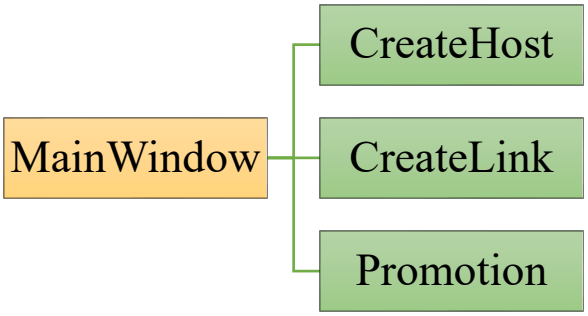
tempMatrix[10][10]:“逼和”时，对本方每一个棋子可能到达的地方进行一遍预演，再判断是否被将军，此数组用于存储预演之后的棋局情况。

attackMatrix[10][10]:存储对方的攻击范围，使得“将军”的判断易于进行。

```
//棋子身份
namespace Pieces {
    const int black_bishop = -6;
    const int black_king = -5;
    const int black_knight = -4;
    const int black_pawn = -3;
    const int black_queen = -2;
    const int black_rook = -1;
    const int none = 0;
    const int white_bishop = 1;
    const int white_king = 2;
    const int white_knight = 3;
    const int white_pawn = 4;
    const int white_queen = 5;
    const int white_rook = 6;
}
```

图 18 不同棋子的编号

5.2 架构



6 感想

相比于上周刚刚熟悉 Qt,这周我对于 Qt 的熟练度明显上升了，一方面也是由于这周的任务更加贴近生活，写起来也十分有熟悉感。同时，在这周写大作业的过程中，我学到了很多第一周没有学到的 Qt 相关知识。通信的学习内容较少，但是应用起来依然需要熟练，大作业给了我很好的锻炼机会，我也感到了我的编程水平在大作业的历练下逐渐提升。

这次大作业遇到的困难主要有“逼和”的逻辑判断以及 TCP 连接数据“粘包”的问题，后者我通过规范信息格式获得了解决，而前者的实现却不是特别优雅：我采用模拟的方式，对本方所走棋子可到达的点进行一遍预演，之后再判断有无“将军”出现，代码逻辑是十分复杂的，同时也带来了计算的开销。我希望将来有时间解决这个问题。

由于时间紧迫，我没有将“多线程”技术应用到局面的判断上，也算是本周大作业的一个遗憾，希望在考试结束之后，我可以改善“逼和”判断的逻辑，同时将计算任务分配到子线程中。

第二周即将结束，小学期带给我的意义也不断增加。聚沙成塔，正是这样一个个点滴般的知识点，铸就了我进步的道路。