网络军棋游戏

计算机系 2020010938 熊志成

1. 客户端与服务器端工作流程

服务器与客户端使用同一套代码，通过一个bool变量确定己方为客户端或是服务器端。

首先发起建立连接的一段为服务器端，客户端通过选项栏中的“连接”与服务器建立连接。接下来，在双方都点击“play”之后，开始游戏。首先是翻棋，本地点击点击一个未翻开的棋子，程序会进行随机分配棋子，并将这颗棋子的全部信息传给另一方，另一方将在棋盘上相同的位置放置一颗同类型棋子。走棋与翻棋类似，在本地点击走棋之后，将相关坐标发送给对方，以同步棋盘。当超时/军旗被拔/无子可走时游戏会弹窗告知双方胜负结果。此外，在走超过20步之后，可以点击admit defeat结束游戏。

以下为相关代码的详细信息。

Junqi.pro

Piece——棋子类，有多种类型，并储存棋子的位置、阵营等相关信息

Square——方格类，对应于棋盘上的每个方格

board——棋盘类

ConnectDialog&CreateConnectDialog——连接&建立连接对话框

MainWindow——主界面

设计时的思路如下。希望将ui与底层逻辑解耦来使代码架构变得清晰，也便于一步一步推进。故先写了piece，square，board这三个自建类，这些内容基本与qt的ui无关，基于此写完了游戏的底层逻辑，譬如piece类中的棋子比较函数、square类中的随机分配棋子函数、board类中的检测棋子移动类型函数，等等。此后添加了界面，譬如给每个square添加指针，指向界面上的qlabel，用这种方式将底层逻辑与ui联系在一起。在写完单机游戏之后，又添加了网络功能，使游戏可以联机对战。

以下为实现游戏相关功能的主要函数

void whoWin()——判断胜负，并弹窗提示

void action(int x, int y, bool random, int chosenSide, pieceType chosenType)——进行棋子的操作

void start()——开始游戏

void init()——初始化游戏

int assignPiece(bool random, int chosenSide, pieceType chosenType)——给方格上随机或指定分配棋子

int isMovable(square\* s1, square\* s2)——判断是否可以移动，并返回一个整数以示移动类型

1. 通信协议

使用TcpSocket进行传输，传输载体为QbyteArray，传输内容为四个整数，int x, int y, int side, int type：

当x，y为大于等于0 的整数时，传递内容为由鼠标点击坐标换算出的二维数组下标，可被用来在棋盘上执行操作。

当x == -1时，表示己方失败，对方胜利

当x == -2时，传递开始游戏的信号

当x == -3时，传递断开连接的信号

当x == -4时，表示己方胜利，对方失败

1. 网络通信编程框架

使用TcpSocket进行传输，利用readyRead、readAll、write、waitForBytesWritten函数进行传输，网络编程部分的代码见connectdialog.cpp（客户端建立连接）、create connectdialog.cpp（服务器端建立连接）、MainWindow::sendMessage(args)（发送数据）、MainWindow::recvMessage()（接受数据）

1. 信号与槽机制设计

游戏在设计时，大量用到了信号与槽机制。如计时器与显示时间、切换当前玩家事件相连接，鼠标点击则被mousepressevent捕捉等等。以下为相关代码的详细信息。

void mousePressEvent(QMouseEvent \*e)——捕捉鼠标在屏幕上的点击坐标

int getRowIndex(QMouseEvent \*e)

int getColumnIndex(QMouseEvent \*e)——与上一个函数将鼠标坐标转换成二维数组下标

void timerUpdate()——当计时器timeout时，触发该slot，更新计时表显显示栏的数据，当从20到0之后记录超时并更换玩家

void recvMessage()——当socket接受到信号后此函数负责处理

void sendMessage(int x, int y, int side, int type)——通过socket向对方发送数据

void on\_actioncreate\_connection\_triggered()——处理建立连接

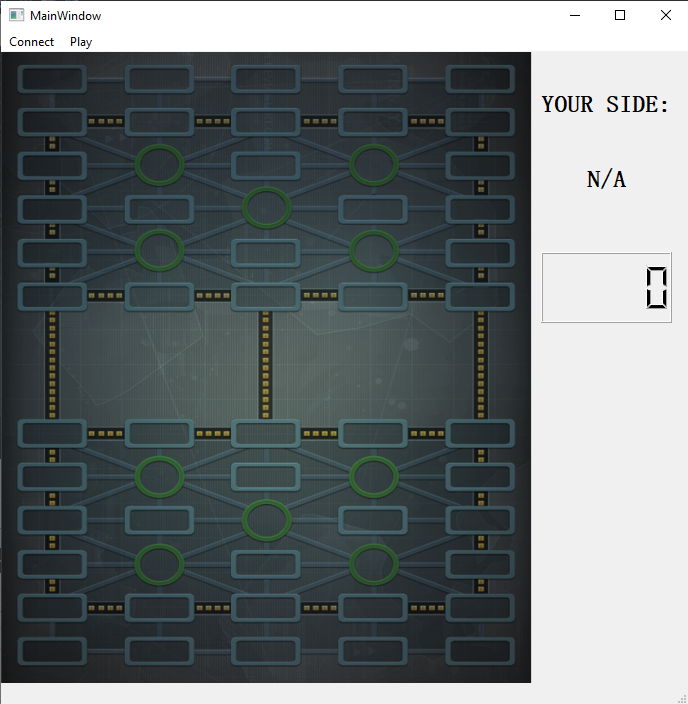
void on\_actionconnect\_triggered()——处理连接到服务端

void on\_actionStart\_triggered()——处理游戏开始

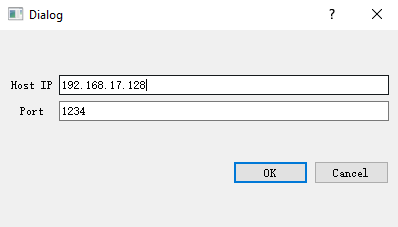
void on\_actionAdmit\_defeat\_triggered()——处理认输，超过20步才可认输

void on\_actionDisconnect\_triggered()——处理断开连接

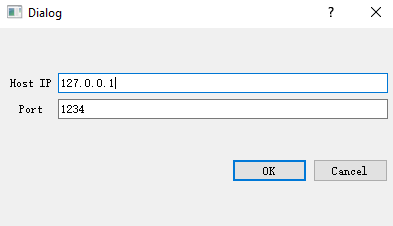
1. GUI界面设计



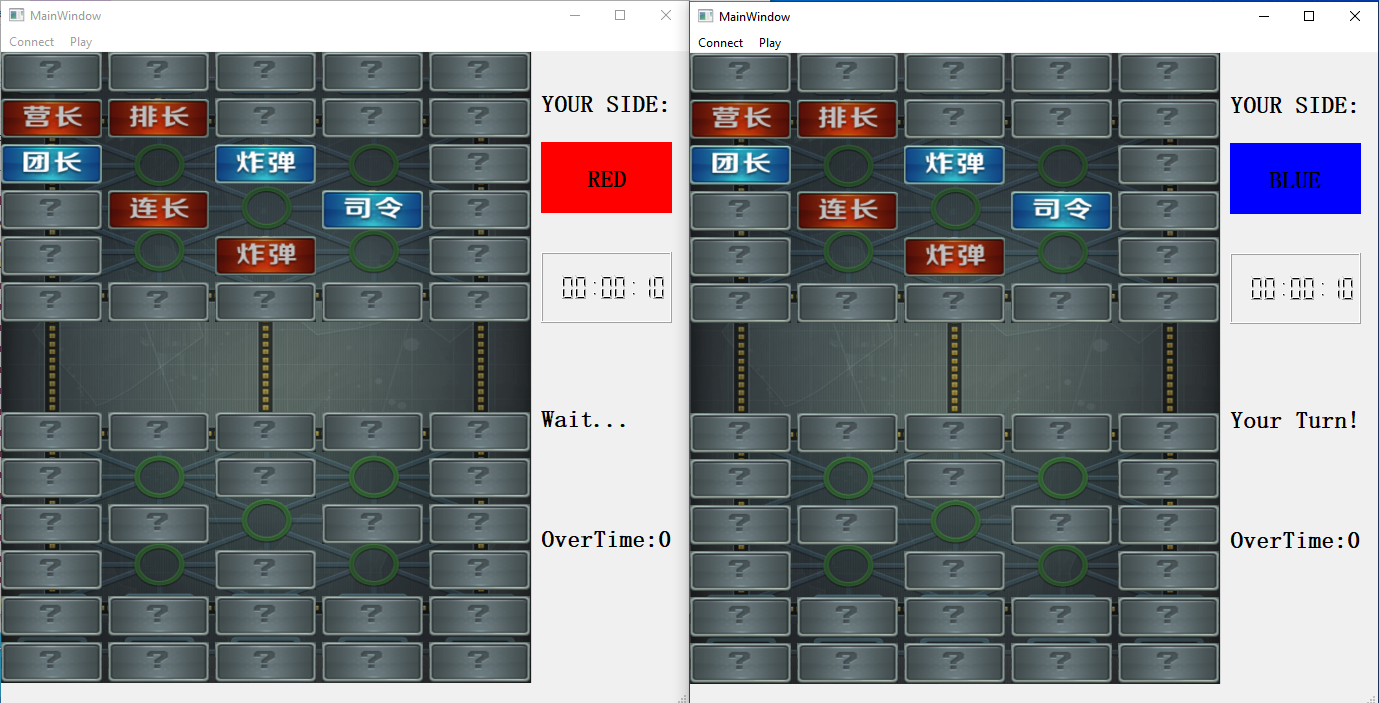
上面的connect与play会显示连接、开始游戏的相关按钮，中间为棋盘，右侧为游戏的相关信息。此时游戏尚未开始，故显示默认信息。



点击建立连接后弹出此窗口，在代码中获得了本机ip地址，并自动填写。



此为连接窗口，默认ip为localhost。



此为游戏时的窗口，可以看到两侧显示的游戏信息。游戏结束后则会以qmessagebox的形式弹窗提醒双方玩家胜负情况，由于内容简单，故此处不再展示。

以上便是网络军棋游戏的设计文档，这次大作业给我的收获很大，尤其是给了我一次工程设计的经历，比如在写代码之前一定要想好写完之后的效果，添加一个较大的功能应该拆分成小的功能逐步完成并测试、要做好版本保存等。可以说，这是一次难忘的经历，也让我对计算机这个专业有了更加深入的认识。