**网络军棋对战小游戏设计报告**

**A screenshot of a computer

Description automatically generated with low confidenceGUI界面设计**

**左图**为程序的初始主页面设计。

初始主页面包含基本的操作指南，游戏开始后操作指南将被隐藏知道游戏结束或断开连接。

主页面的上方的菜单栏包含两个选项，“Connect”及“Play”。

“Connect ”栏中包含与网络相关的操作按钮：Graphical user interface, text, application, chat or text message

Description automatically generated

点击“Create Connection”将弹出对话框，对话框中包括Server端IP显示框、确认及取消按钮。点击“确认”，对话框消失，成功创建服务间端口，并等待连接请求。

点击“Connect to Server”菜单项将弹出对话框，对话框中包括需要连接的服务器端IP输入框、确认及取消按钮。点击“确认”，如IP格式正确及服务器存在，将完成连接；若出错，将要求用户重新输入。

点击“Disconnect”将弹出对话框，对话框中包括“是否断开链接？”的确认信息、确认及取消按钮。点击“确认”，连接将被断开。此选项在连接的情况下才能使用

“Play”栏中包含与游戏进行的相关的操作选项：

Graphical user interface, text, application, chat or text message

Description automatically generated

点击“Start”将绘制棋盘，开始游戏。

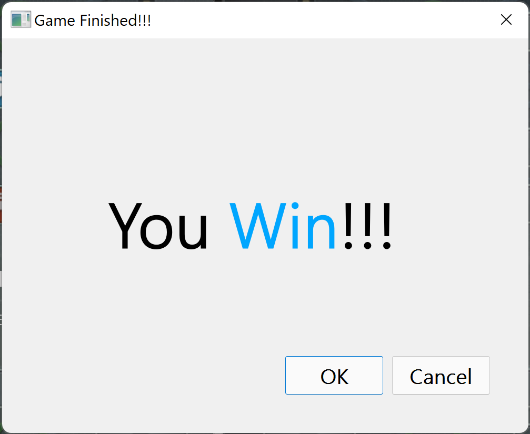
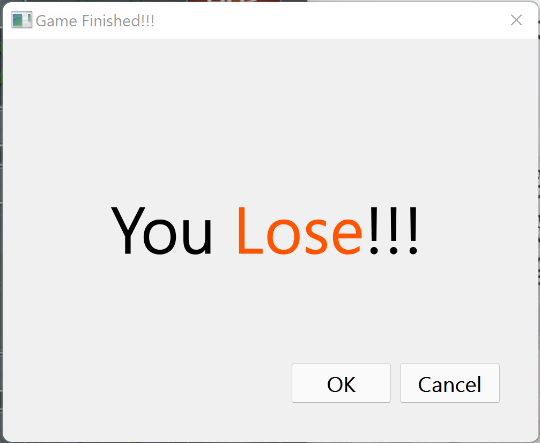
20步后，点击“Forfeit“，将弹出对话框，对话框包含“是否认输？”的确认信息、确认及取消按钮。点击“确认“将弹出胜负信息，确认后初始化棋盘。20步前点击此按钮将显示为违规操作。

点击“Refresh“将初始化棋盘。

**右图**为点击开始后的棋盘。

**A screenshot of a computer

Description automatically generated with low confidence**

****A screenshot of a computer screen

Description automatically generated with medium confidenceA screenshot of a computer screen

Description automatically generated with medium confidence

当分出胜负后，胜利信息的弹窗将弹出，点击确认后将初始化棋盘。

**左图**为游戏进行中的主页面设计。

左上角将显示该玩家的持方（颜色），及该轮所剩余的时间

右上角显示此颜色玩家所剩余的生命（剩余能超时的次数）

主页面中心的左侧显示当前被选中的棋子，且该棋子将被黄色的框包围；右侧显示当前是否轮到该玩家走棋，以及游戏当前的总步数。

**下图**主页面拥有两种警示信息：

-右上角为超时警示，超时将弹出。

-中心左侧为非法操作的警示信息，出现如选择非己方棋子、吃等级高于自己的棋子及20步前投降等非法操作时弹出

**信号与槽机制设计**

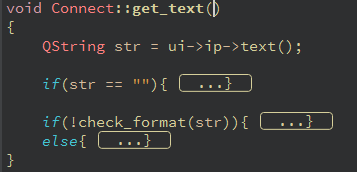
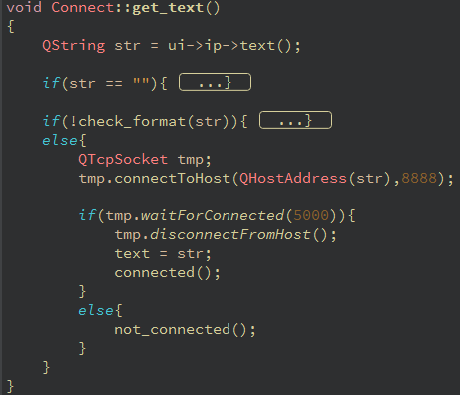
**Text

Description automatically generated**Text

Description automatically generated**Text

Description automatically generated**

Text

Description automatically generated******网络通信设计**

上图显示了其他弹窗类中的信号槽链接。

图一显示的是，创建服务器、认输及断开连接弹窗类的信号槽连接，只有在确认后才会告知MainWindow进行相关操作。

图二是结果的弹窗类，无论点击任何按钮都将关闭弹窗，告诉MainWindow初始化棋盘

上图显示了connect类中的信号槽链接。

每当点击“确认”都将进入IP地址的判断和可连接性的测试，若成功则将结果传回给MainWindow

上图显示了MainWindow的信号槽链接。

图一显示的是菜单栏中除了disconnect的连接。

图二、三显示只有在建立网络连接后才连接disconnect的信号槽，断开连接后则断开disconnect的信号槽连接。

图四则显示游戏开始后则断开start的信号槽链接。

右图为客户端的相关代码。

首先，将在connect类里判断所输入的IP地址是否符合格式，若符合则判断是否能够连接该服务器。

如能连接则将结果返回MainWindow，建立socket链接该服务器。

成功建立连接后将socket与传输信息的slot进行连接，同时也将disconnect的菜单栏选项进行信号槽链接。

两者的通信都是使用TCP协议进行信息传输。

左图为服务器端的相关代码。

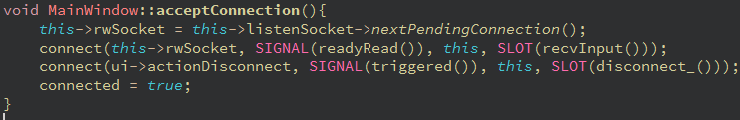
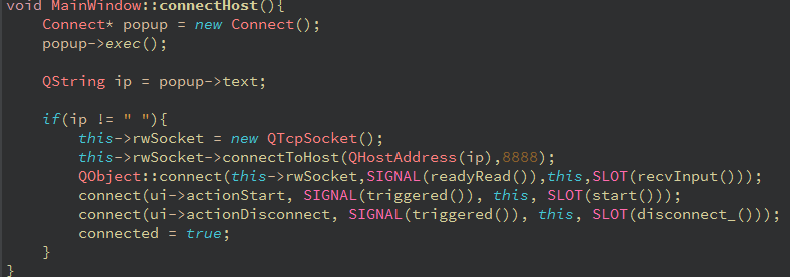
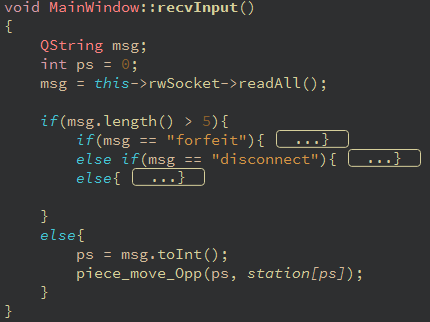
首先，获取本地主机信息，获取本地主机的所有IP地址，将显示第一个IPv4地址供客户端连接。

当用户确认建立服务器后，将开始监听所有进来的连接请求。

成功建立连接后将socket与传输信息的slot进行连接，同时也将disconnect的菜单栏选项进行信号槽链接。

**Text

Description automatically generated with medium confidenceText

Description automatically generatedText

Description automatically generated**

两者通信所传输的信息是所点击的按键或棋子，该信息将由sendInput(QString msg)函数写入socket中，另一方者通过recvInput()函数读取socket中的信息，根据所读的信息进行与另一端进行相同的操作或判断工作。

因此所设计的start函数分为点击“开始”那一方和非点击那方。一方点击即可开始游戏，其信息将发给另一方。另一方将从socket中读取点击开始放所生成的rand()的seed来绘制棋盘。

走棋、吃棋输入的操作函数也分为两个，即己方和对方两种。其信息都是由TCP协议进行传出和传入。