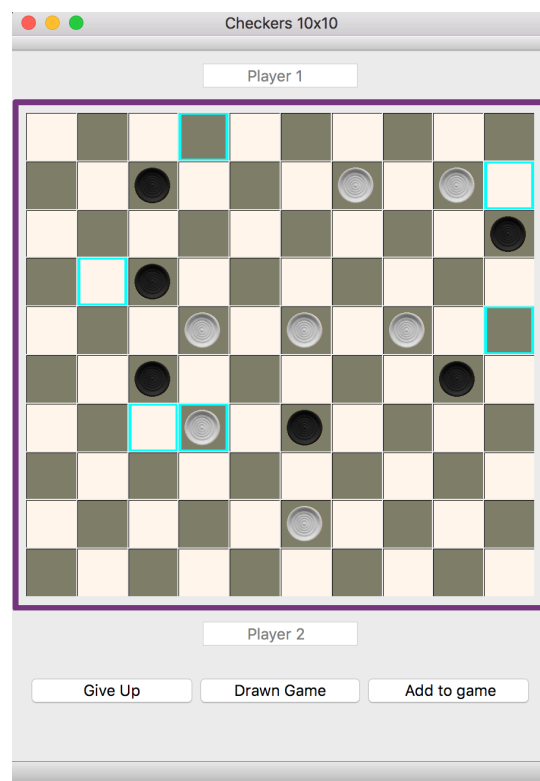
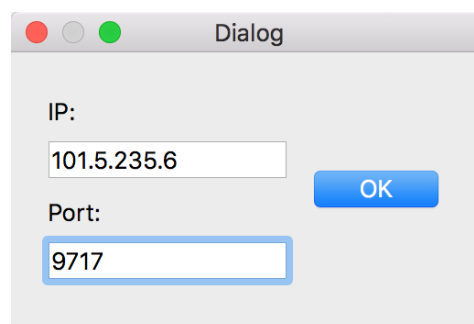


创建连接界面

## 2. client 端

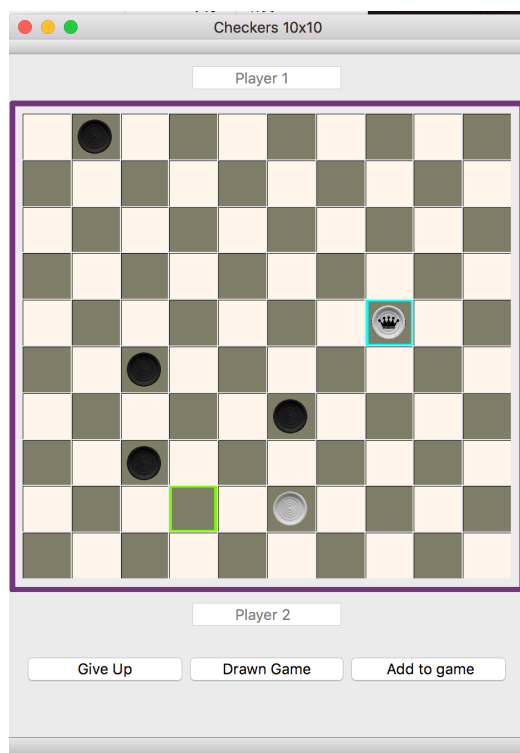


主界面

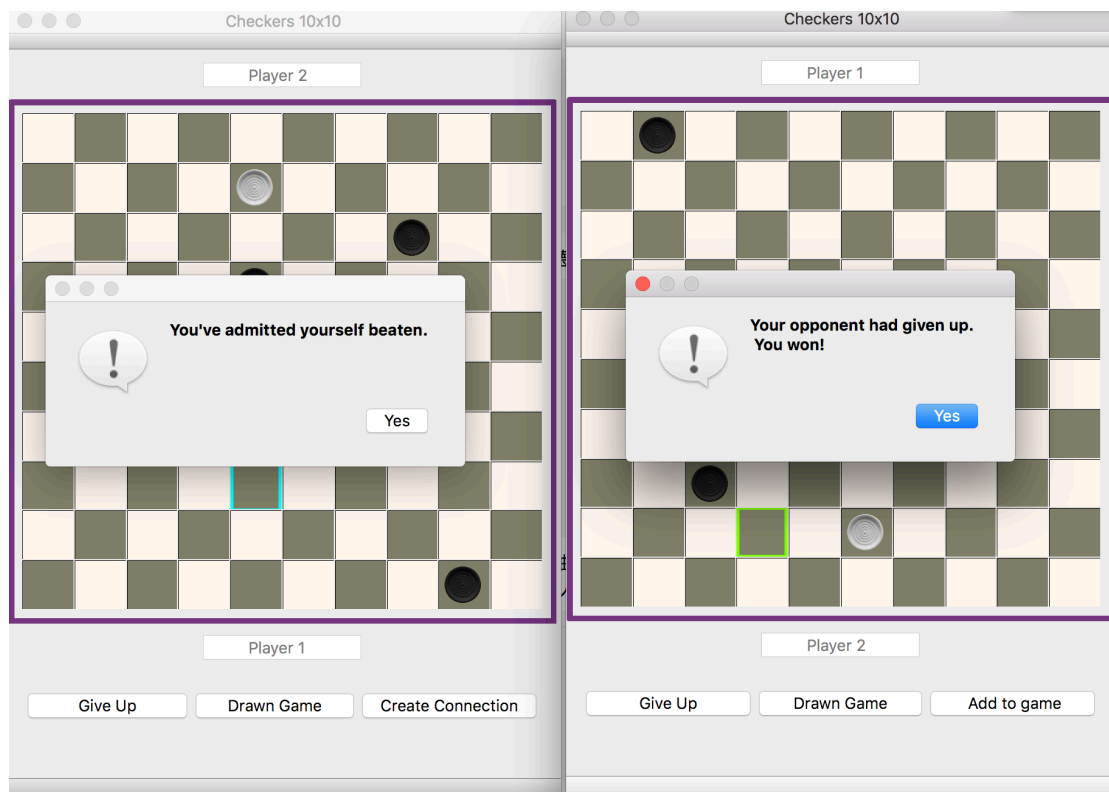


连接到服务器界面

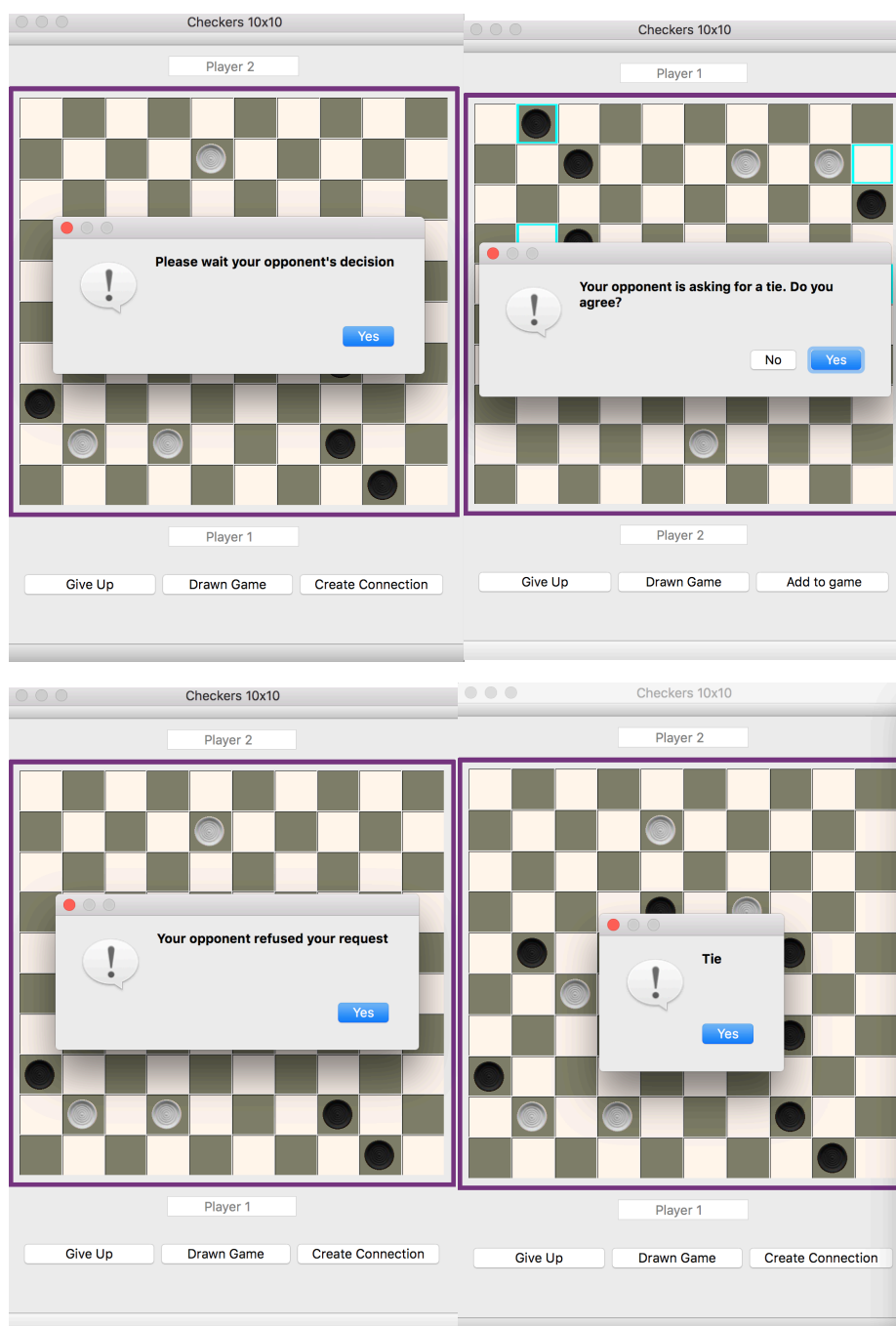
### 3. 游戏界面



#### 4. 认输功能



## 5. 求和功能



## B. 程序实现

整个程序由内核和界面两部分构成。算法由 `Draughts` 类实现，在 `Chess` 类当中，根据算法类的数据，重写 `paintEvent` 函数实时更新界面显示情况。而 `Checkerboard` 类的功能则是对 100 个 `chess` 类对象进行管理、使界面与算法相联系起来。

## 1) MainWindow 类

MainWindow 类继承自 QMainWindow，主要负责主界面的绘制以及网络连接。

✧ void connectHost(QString str), void sendMessage(QString), void recvMessage() 三个函数在用户端通过使用 QTcpSocket 实现了连接服务器、数据的传送和接收。

✧ QString initServer(buildConnectionDialog\*), void acceptConnection(), void sendMessage(QString), void recvMessage(), void closeServer() 五个函数在服务器端使用 QTcpServer 和 QTcpSocket 实现了服务器初始化、连接，数据传送和接收，关闭服务器的功能。

## 2) Checkerboard 类

✧ recv 函数实现了客户端以及服务器端的字符串的解析。

✧ Upd 函数则是将内核当中数字的改变实时让 chess 类的 paintEvent 函数画出来。其余是信号槽的连接，实现了算法类和界面类信号的互通。

## 3) Chess 类

✧ paintEvent 函数负责 100 个格子的绘制以及棋子的绘制，根据内核当中棋盘数组数字，来实现移动棋子的动画效果。0 为没有棋子，1 为黑旗，2 为白棋，-1 为黑王，-2 为白王，

✧ setHighLight 函数则是根据算法当中下一步能走的位置进行高亮，以及对当前选中的棋子的高亮。

#### 4) Draughts 类

✧ 本类是算法的负责类。运用到了DFS的算法，searchForPossible函数负责在每一跳的位置每一个棋子下一步可以走的位置进行深搜，运用数据结构当中的树，来探索每一个棋子能走的最大步数为多少，从而符合了题目要求：必须走吃子最多的路线。其中具体对每一个棋子实现DFS算法的是goStore函数。

✧ initStore函数则负责在每一步起跳的时候更新各变量的值，并且检查当前所有子是否有可走的路线，如果没有，则判定棋手出局。

✧ endStep函数则是在每一次一个棋子被跳过之后将其值设为3，保证被吃的棋子不能被反复跳过。

✧ checkQueen函数负责对到达底线的棋子进行检查，如果符合条件则将其变为皇后。

#### 5) inputIPDialog 类

inputIPDialog继承自QDialog，实现了Client端的IP输入。

#### 6) buildConnectionDialog 类

buildConnectionDialog类继承自QDialog，用以显示Server的IP以便Client端进行连接。

### C. 网络通信实现

- 实现基础

本程序才用Tcp协议，以QT提供的QTcpServer和QTcpSocket作为通信基础。

1. 具体流程

- Server端

当用户点击Create Game，程序调用QString  
initServer(buildConnectionDialog\* w)，通过将QTcpServer listenSocket新建出来并  
设置listen(QHostAddress::Any, 8888)，设定监听IP为任意，端口8888。然后将  
listenSocket的newConnection()与acceptConnection()连接。若是用户点击cancel，  
程序将执行listenSocket->close()关闭服务器。

在void acceptConnection()中，QTcpSocket readWriteSocket赋值为  
listenSocket->nextPendingConnection()，并将其readyRead()与recvMessage()相  
连。

在void recvMessage()中，readWriteSocket通过readAll()操作读取字符串，其  
中字符串为Client端发来的下棋信息，其中包括两个整数x，y表示对手下棋的坐  
标。

在void sendMessage(QString info)中，readWriteSocket通过write向Client发  
送同样的两个整数x，y表示自己下棋的坐标。

- Client端

用户在连接界面输入IP后通过点击确认，将IP地址存在str中传给void  
connectHost(QSring str)函数。其中，QTcpSocket readWrite被初始化并连接str的  
IP，端口8888，并将其readyRead()与recvMessage()相连。

在void recvMessage()中，readWriteSocket通过readAll()操作读取字符串，其  
中字符串为Server端发来的下棋信息，其中包括两个整数x，y表示对手下棋的坐  
标。

在void sendMessage(QString info)中，readWriteSocket通过write向Server发送同样的两个整数x, y表示自己下棋的坐标。

#### D. 总结

本周的大作业相对第一周来说难度较大，但我也在规定时间内做完了题目中的所有要求。最大的感触是，比起第一周来，我的看文档能力以及代码能力感觉有了较大的提升。之前有自学过IOS开发，但是由于缺少练习，学到的一点东西也很快忘记了。但是这次对于QT，网络的学习，几乎无时无刻不在高强度的练习给了我较为扎实的基础。由此可见唯有脚踏实地地实践才能真正掌握知识。