International Draughts

网络国际跳棋对战游戏设计文档

郑少锟

计65 2016011381

目录

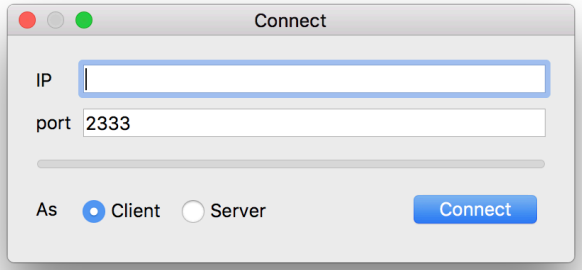
设计

Design

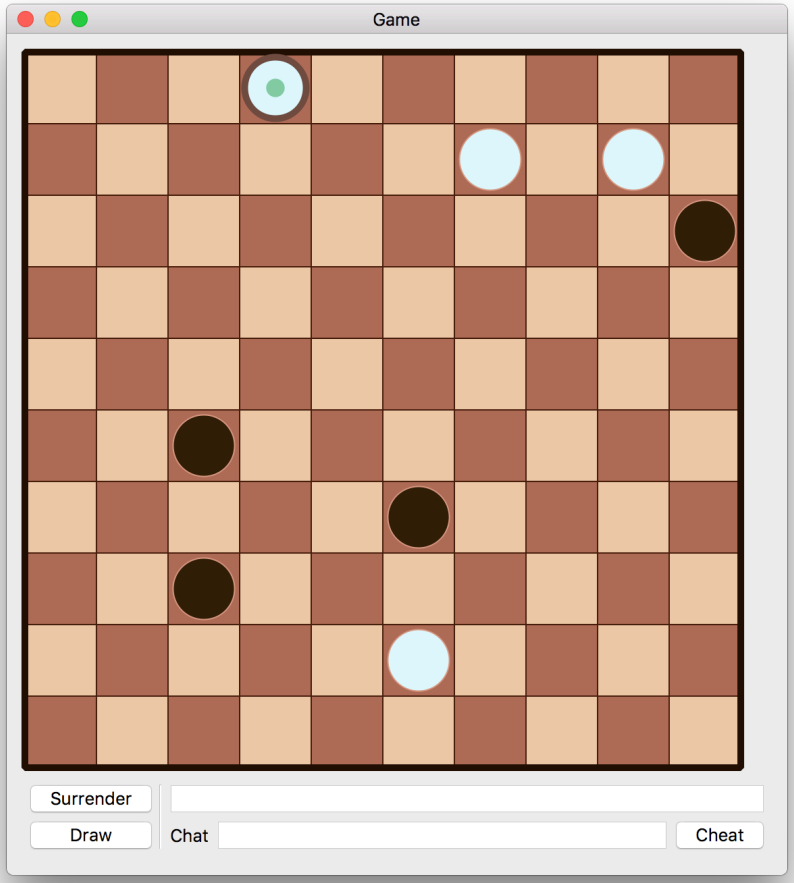
界面

主要界面分为登录与游戏。

其中，登陆界面使用一个自定义的Dialog实现。用户可以通过单选按钮方便地选择将本机作为服务器端或客户端。当选择做为服务器端时，IP一栏将自动提示本机IP方便客户端的连接。



按下connect按钮即可尝试建立连接，并通过进度条提示等待进度减轻用户等待时的烦躁。连接成功后将自动进入游戏界面；失败时将以消息框的形式提示用户。



游戏界面包括上方的棋盘和下方的投降、求和按钮及聊天区域（聊天按钮在作弊模式中用于开启或关闭棋盘的编辑功能）。棋盘中通过浅蓝色边缘标示可用棋子，灰色边缘标示选中棋子，红色细点标示可行着法，浅绿圆点标示加冕棋子。

游戏双方的棋子的选取与移动通过网络连接实时同步。

音效

游戏为用户的不同的操作提供了各种音效。例如选取棋子、移动棋子、投降、胜利、和局等等。这些音效增强了游戏的趣味性。

功能

Functionality

基本规则

本项目实现了国际跳棋的所有基本规则，能够正确处理棋子的移动、吃子、加冕等情况，并能过滤不正确的操作。

联机对战

游戏支持网络双人对战功能。用户可以在登陆界面选择创建或加入游戏，通过选取IP地址与端口号即可实现局域网内的在线对战。对战过程中支持投降及求和操作。

同时，为方便对战双方的交流，游戏时还提供了简单的聊天功能。

交互

用户在游戏进程中通过键鼠进行交互。而在诸如求和、投降等操作时，游戏也会通过对话框友好地提示用户进行确认，保证了操作的安全性。

作弊

为方便测试，游戏提供了作弊功能和完整的作弊控制功能。作弊功能有源码级和运行时两种层次的控制选项。

利用条件编译，只需在源码中更改宏ALLOW_CHEATING_MODE的值即可做到完全关闭作弊功能且不影响其他功能的正常使用；而当宏指令开启时，用户可在登陆时输入特定的端口号（65535）开启作弊模式。

作弊模式提供了友好的棋盘即时编辑功能。启用后只需按下界面上的Cheat按钮即可通过键盘和鼠标方便快捷地进行盘面编辑，并可与对手机实时同步。具体操作如下表所示。

操作	效果
选中Cheat按钮	进入作弊编辑模式并暂停游戏。
取消选中Cheat按钮	退出作弊编辑模式并继续游戏。
单击鼠标左键	选取当前需要编辑的方格。
按下键盘上的W	将当前格中的棋子颜色置为白色（**W**hite）。
按下键盘上的B	将当前格中的棋子颜色置为黑色（**B**lack）。
按下键盘上的E	清除当前格中的棋子（**E**mpty）。
按下键盘上的N	将当前格中的棋子类型设置为普通（**N**ormal）。
按下键盘上的C	将当前格（非空时）中的棋子类型设置为加冕（**C**rowned）。

架构

Architecture

MVC架构

游戏采用MVC架构。其中，模型部分位于/Model文件夹下，实现了国际跳棋的数据表示和路径搜索算法；视图和视图控制器位于/UI文件夹下，实现了国际跳棋视图和游戏逻辑控制器。良好的架构使得程序简单易读且便于修改。

数据结构与算法

Data Structures & Algorithms

玩家表示

玩家通过玩家（Player）和玩家帮助器（PlayerHelper）类型来表示。

其中，玩家是一个枚举类型，用于表示当前玩家、本机玩家等。玩家帮助器是一个结构体，提供了一系列用于生成或转换玩家类型的静态函数，如玩家类型与字符串的相互转换、随机生成玩家、交换玩家等等。通过显式地将构造函数标记为删除来防止该类的实例化。

棋子表示

棋子使用棋子类型（PieceType）、棋子颜色（PieceColor）和一个棋子帮助器（PieceHelper）来表示。

其中，棋子类型为一个枚举类型，用于表示当前棋子的加冕情况。棋子颜色和棋子帮助器实际上是玩家和玩家帮助器的别名。

棋盘表示

棋盘通过一个棋盘类（Board）、方格结构体，以及其内嵌套的一个位置结构体（Position）表示。

方格结构体是棋子结构体的别名。棋盘类内部通过一个二维的方格数组来表示棋盘，并利用位置结构体作为索引。

棋盘类实现了对于棋盘的增减和获取棋子、加冕棋子等基本操作。

路径表示

为了应对国际跳棋相对复杂的游戏规则，利用一个独立的路径类来实现棋子可行路径的搜索功能。

路径（Path）通过路径节点（PathNode）相互连接形成的树状结构来表示。使用路径搜索器（PathFinder）提供的一系列静态成员函数来构建此路径。路径搜索器内部使用了深度优先搜索算法来探索单个棋子可行的最长路径，并适当地使用了剪枝算法来缩小所需的内存空间。实现上，利用类成员函数指针的列表实现了针对不同类型棋子的统一而简洁的路径搜索算法。

游戏表示

跳棋游戏类（Draughts）是上述类型的聚合，实现了对跳棋游戏的抽象表示。利用上述游戏组件，游戏类提供了对于游戏的初始与结束、棋子选择、着法选择、回合轮转、游戏进程判断等等相对高阶的操作。

体会

Feelings

感受与收获

这次大作业带给我的最直接的收获就是初步的网络编程知识。相比于原先的单机编程，网络编程需要考虑的因素更加纷繁复杂。比如各种数据类型的编码、传输与解码，消息的延迟处理与时序控制等等，很多问题都是我之前在单机编程时从未注意到过的。而这次大作业无疑给了我这么一个机会去了解、思考与真正实际解决这些新问题、新挑战。

其次是软件工程方面的知识。其实最初几个版本时，我的程序架构并不是非常清晰，游戏控制器和跳棋视图没有正确分层，导致逻辑与视图混杂在一起，每次修改几乎都像是排雷，各种Bug是按下葫芦又浮起瓢。望着IDE里边的一坨坨积重难返的代码，再看看更新之后的大作业要求，我深感人生无望。于是，在一个阳光明媚的午后，不堪忍受的我终于做出重构代码的重要决定。所说当我再一次按下编译并通过时，时间已是深夜，但看着层次清晰、逻辑缜密的新代码时的喜悦心情还是难于言表；而后应对那些奇奇怪怪的大作业要求也已是游刃有余。

嗯，说了这么多，总之，良好的代码结构很重要，而有些设计之初就可以避免的问题也应尽早处理，免得到时仍需面对麻烦的代码重构。当然，有时代码重构带来的收益是远大于它的复杂性的，何况重构后的优美代码还可以带来巨大的心理愉悦感呢。