

Draughts

Design Document

For Alpha Version

Contents

Chapter 1:页面及布局.....

3

Chapter 2:功能说明.....

7

Chapter 3:设计思路.....

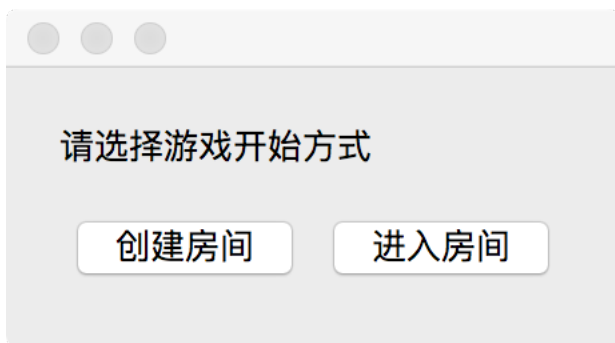
11

Chapter 1

页面及布局

*Draughts*是一款具有网络对战功能的PC平台国际跳棋游戏。

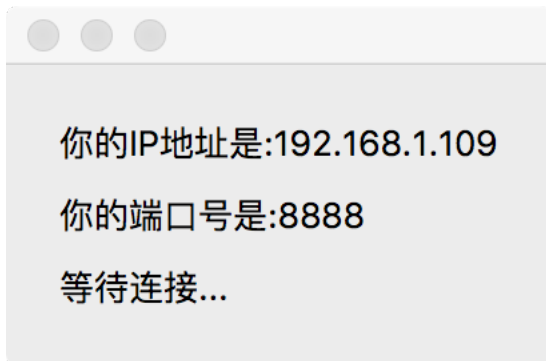
首先，打开软件之后，会出现选择创建房间或是进入房间的对话框。要想进行游戏，必须有两个游戏端，一端创建房间，另一端进入房间，二者才可以开始对战。



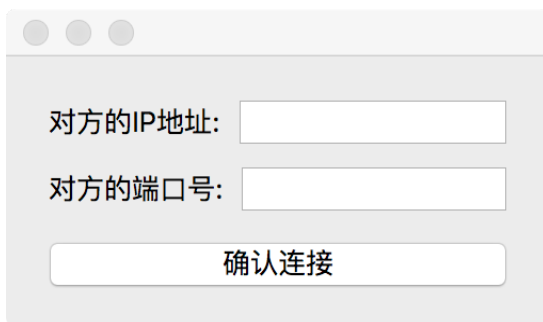
若选择“创建房间”，会进入创建房间的设置界面，需要输入自己设置的端口号。



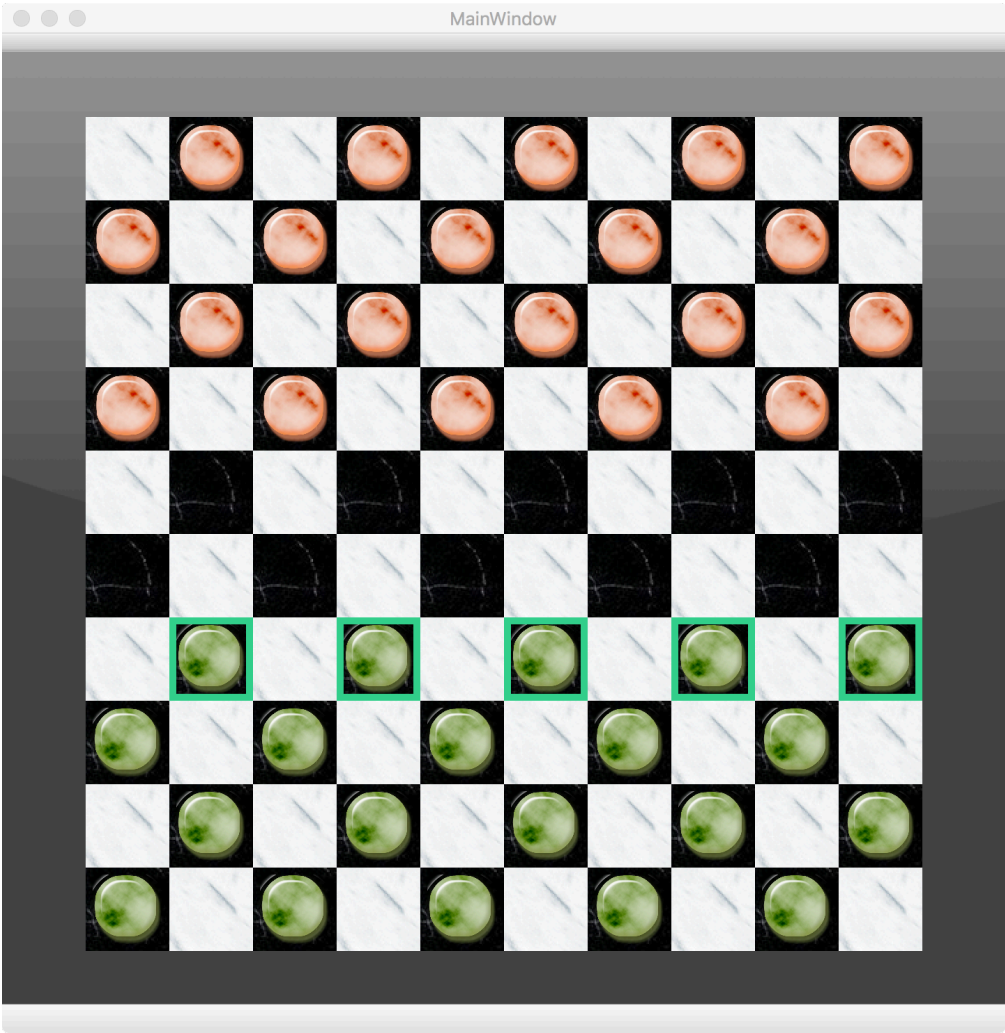
输入自己设置的端口号并按下“确定”按钮之后，会进入等待状态，等待另一个玩家进入房间。



若选择“进入房间”，则需要输入房主的IP地址和端口号进行连接。

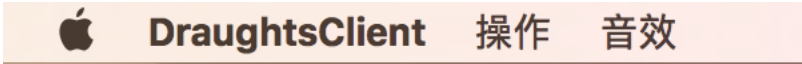


建立起连接之后，两位玩家会同时进入游戏界面。
房主持红色棋子，后行；另一个玩家持绿色棋子，先行。

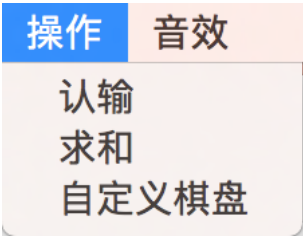


持绿色棋子玩家游戏开始之后的游戏界面

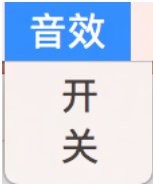
菜单栏中可以做出认输、求和、自定义棋盘、开关音效的操作。



菜单栏



“操作”菜单



“音效”菜单

在玩家做出“认输”、“求和”、“自定义棋盘”等操作后，会出现相应的对话框，在比赛结果已定时，会弹出对话框提示玩家游戏结果，同时使棋子不可移动。

游戏中出现的颜色：



对方上一次的棋子移动



本方可以移动的棋子



当前已选中的可以动的棋子



当前选中的棋子可以到达的地方



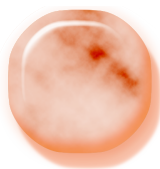
当前选中的棋子的路径的中间点

Chapter 2

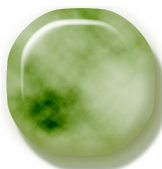
功能说明

1.网络对战功能。创建房间方可以显示IP，自定义端口号并进行监听；进入房间方可以填写对方IP及端口号并进行连接。对方的走棋可以显示在自己的棋盘上，本方的走棋也会实时地显示在对方的棋盘上。

2.国际跳棋的完整规则。可以检查用户操作的合法性，只对正确的操作作出响应，并且实现吃子、王棋等功能。



红方棋子



绿方棋子

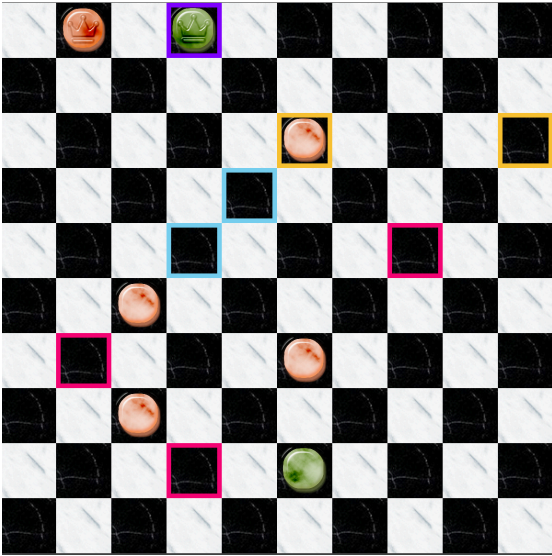
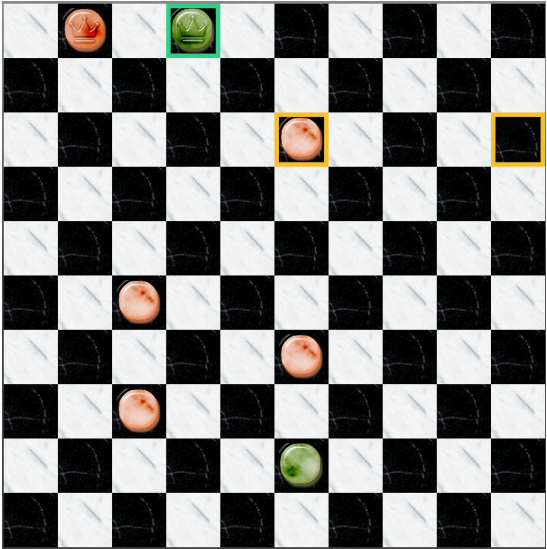


红方王棋

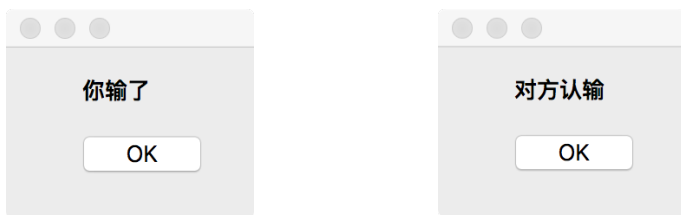


绿方王棋

3.提示功能。到本方回合之后，会用绿色边框显示当前可以移动的棋子。第一次点击可以选中棋子，棋盘上会用蓝色边框显示该棋子可以到达的地方，用红色边框显示路线。



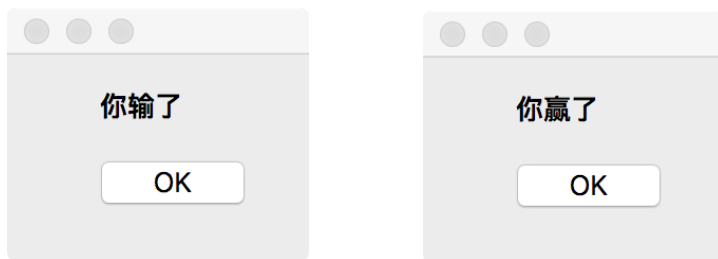
4.认输功能。认输功能在菜单栏中。一方选择认输后，会直接判定认输一方失败，另一方因对方认输而胜利。



5.求和功能。求和功能在菜单栏中。一方选择求和后，等待对方做出选择。若对方选择同意求和，则直接判定比赛结果为平局，否则开始计数，按照标准比赛规则，不同意求和的一方在40步之后仍无法结束比赛的话，判定比赛结果为平局。



6.判断输赢。当一方无棋子或者所有棋子无法移动时，显示对方获胜。



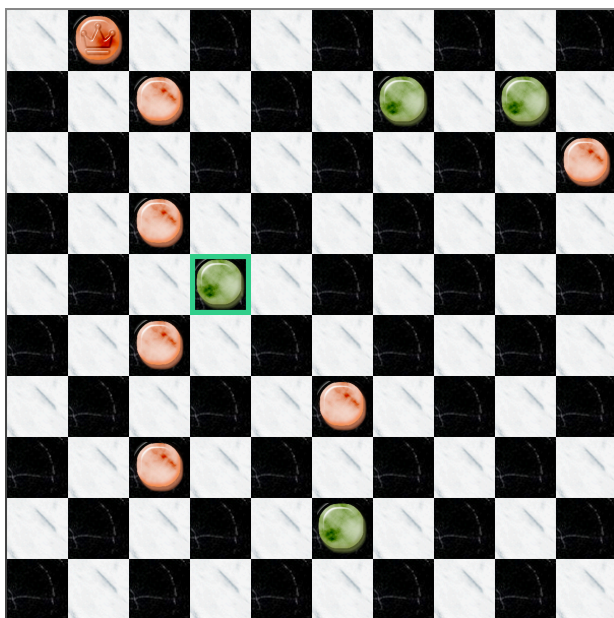
7.音效。在游戏开始、获得胜利、普通棋子变为王棋、棋子移动、棋子跳吃时，均会有对应的音效。

8.自定义棋局。可以以文本的方式输入棋局，按照从左到右、从上到下的顺序设置棋盘方格中的棋子类型，0代表无棋子，1和2分别代表红色和绿色普通棋子，3和4分别代表红色王棋和绿色王棋，其他数字也代表无棋子。

例如：

```
0300000000000100020200000000000100100000000002
000000001000000000000010000001000000000000200
00000000000000
```

表示



Chapter 3

设计思路

我觉得国际跳棋这个大作业的的工作量主要在三个方面：QT界面的设计与互动效果、符合国际跳棋规则的提示玩家的算法和socket通信。下面就从这三个方面谈一下我的设计思路吧。

1.QT界面的设计与互动效果

首先谈一下我对ui设计的一点想法。首先就是要根据自己的软件类型和受众人群来确定自己的ui设计风格，是卡通还是科幻，是小清新还是重金属。一定要首先想好风格，这样才能达到ui设计的一个重要目标：和谐统一。不同的风格在用色、元件形状和布局方面都有很大的区别。

关于色彩方面，ui设计中一般使用比较淡雅的低饱和度色彩，各种app一般是有一个主色调，如果某个界面颜色比较多的话，为了避免色彩的冲突，就会倾向于使用低饱和度的色彩。高饱和度的颜色一般比较鲜艳，可以用在海报或者需要吸引注意力的平面设计上，如果是设计app的界面的话，为了避免视觉疲劳，尽量使用低饱和度的颜色。

这里要推荐一个色系：莫兰迪色系。他的画作具有治愈的力量，因为他的画中大面积地用到了低饱和度和明度相近的颜色。



莫兰迪色系

扁平化设计也是一个重要的趋势。扁平化的重要意义就是去除冗余、厚重和繁杂的装饰效果，在设计元素上强调抽象、极简和符号化。也许还可以降低功耗、延长待机时间和提高运算速度。大家众所周知的ios系统就是使用了扁平化设计。

现在还有一个ui设计的大趋势，就是扁平化设计。扁平化的重要意义就是去除冗余、厚重和繁杂的装饰效果，在设计元素上强调抽象、极简和符号化。也许还可以降低功耗、延长待机时间和提高运算速度。大家众所周知的ios系统就是使用了扁平化设计。



扁平化设计例子

我写了一个新类(Cell类)来继承QPushButton类，这个类既可以做棋盘的方格，又可以做棋子。我在MainWindow中开了两个10*10的Cell*的二维数组，一个用来显示棋盘，初始化之后就不再改变，另一个用来显示棋子，棋子移动时就相当于设置相应位置Cell*的属性。设置棋子的颜色时我使用了stylesheet中的border-image属性，这样的话，图片会成为背景图片并且根据对象的实际大小来进行缩放。需要注意的是，对于已经设置了border属性的对象来说，就只能使用background-image了。

在摆放组件的位置的时候，我没有使用layout来设置，而是使用了setGeometry函数，这样对组件的位置和大小有更加精确的把握，而且对组组件的大小和位置进行批量化修改的时候也比较方便。

最后，当然要使用QSignalMapper啦。

以及，一些图片素材和声音素材是通过拆安卓国际跳棋游戏和Mac上的国际象棋游戏的包获得的。

2.符合国际象棋规则的提示玩家的算法

这里说的提示玩家包括两部分，一部分是提示玩家当前可以移动的棋子有哪些，另一部分是提示玩家当前选中的的棋子可以移动到哪些地方。这两部分都涉及到一个核心问题：找到棋盘上所有棋子在只能移动一个棋子的条件下吃子最多的路线。

我使用DFS来解决这个问题。从某一个己方棋子出发，区分该棋子是王棋还是普通棋子，然后朝四个对角线的方向搜索，不断更新吃子最多的路线。大家可能会有更加优秀的算法，但我的第一反应就是DFS，而且写起来也比较熟悉。

3.socket通信

这周学习的主要内容就是socket通信。我使用了老师课上提供的ChatClient和ChatServer例子的源码，然后通信问题就解决了，剩下的就是双方通信的协议了。我是对QString形式的通信内容进行操作。在QString的开头用字母来区分对方的指令类别，后面加上数字来讲指令的内容具体化。

例如：m03,1416,2507 表示从第0行第3列出发，吃掉第1行第4列和第1行第6列的棋子，经过第2行第5列和第0行第7列。翻译指令的时候当然要注意双方的棋盘朝向不同，需要将对方的第0行对应到自己的第9行，对方的第9列对应到自己的第0列的问题。

4.其他

在移动棋子的过程中，我希望在跳吃的过程中在每个位置停留一段时间，以显示棋子的移动路径。

我使用的方法是在移动棋子的代码之后加入一段开始局部时间循环的代码：

```
QEventLoop eventLoop;  
QTimer::singleShot(200,&eventLoop,SLOT(quit()));  
eventLoop.exec();
```

这样可以避免程序界面不响应用户的操作。