

国际象棋开发

作者：付强、唐婷婷、陈熙蕊



2024-7-9

[国际象棋项目开发](#_Toc171445340)

[目录](#_Toc171445340)

[国际象棋项目开发 2](#_Toc171445607)

[一、 项目介绍 2](#_Toc171445608)

[一、项目背景 2](#_Toc171445609)

[二、需求分析 2](#_Toc171445610)

[二、 UI设计 3](#_Toc171445611)

[一、主页面 3](#_Toc171445612)

[二、游戏页面 4](#_Toc171445613)

[三、设置页面 5](#_Toc171445614)

[四、帮助页面 6](#_Toc171445615)

[三、 棋盘层次结构 6](#_Toc171445616)

[棋盘层次分为三层 6](#_Toc171445617)

[四、 详细设计 7](#_Toc171445618)

[一、走棋与悔棋实现 7](#_Toc171445619)

[二、联机实现 8](#_Toc171445620)

[三、页面相关实现 10](#_Toc171445621)

[五、 成员及项目任务大致分配 12](#_Toc171445622)

[六、问题分析及处理 12](#_Toc171445623)

[1](#_Toc171445340)

# 国际象棋项目开发

## 项目介绍

### **一、项目背景**

国际象棋（Chess）是一种历史悠久、在世界范围内广泛普及的二人对弈的智力游戏。它起源于亚洲，后由阿拉伯人传入欧洲，成为国际通行的棋种，也是一项智力竞技运动，曾一度被列为奥林匹克运动会正式比赛项目。

国际象棋棋盘为正方形，由横纵各8格、颜色一深一浅交错排列的64个小方格组成。棋子分为黑白两组，共32个，每方各16个，兵种分为王、后、车、象、马、兵6种。国际象棋的游戏规则简单明了，双方轮流走棋，以把对方的王“将死”为胜。

### 二、需求分析

#### 功能需求

1.提供标准的国际象棋规则，包括棋子的移动方式和比赛目标。

2.允许玩家进行单机双人对战，局域网联机对战

3.提供悔棋功能，允许玩家撤销上一步或几步的移动。

4.提供保存和加载游戏的功能，以便玩家可以在不同的时间继续未完成的比赛。

5.提供帮助功能，包括基本规则介绍、开局策略、中局战术和残局技巧。

6.提供游戏轮次提醒功能

7.提供设置功能，可以更换界面主题背景颜色，控制声音大小

#### 2.用户界面需求

1.提供直观的图形用户界面，包括棋盘、棋子和控制按钮。

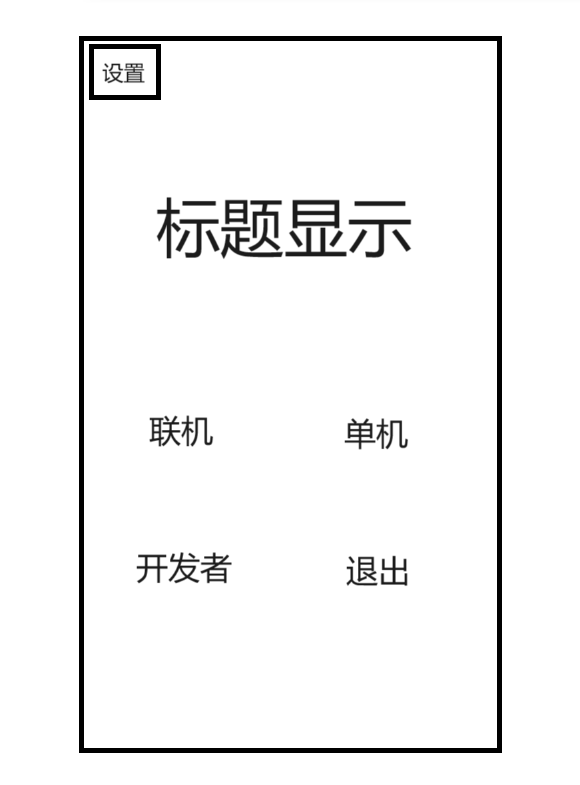
2.允许玩家通过鼠标点击或触摸屏幕来移动棋子。

3.显示当前玩家和回合的状态。

4.提供视觉和声音效果以增强游戏体验。

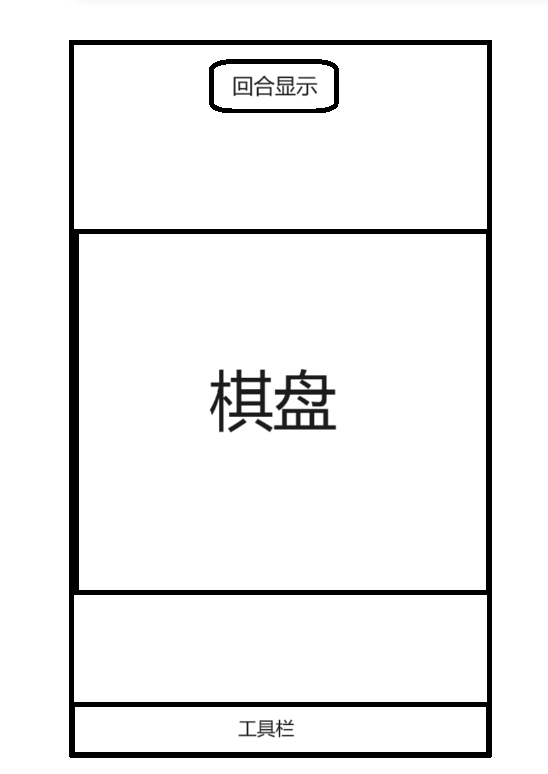
## UI设计

### 一、主页面



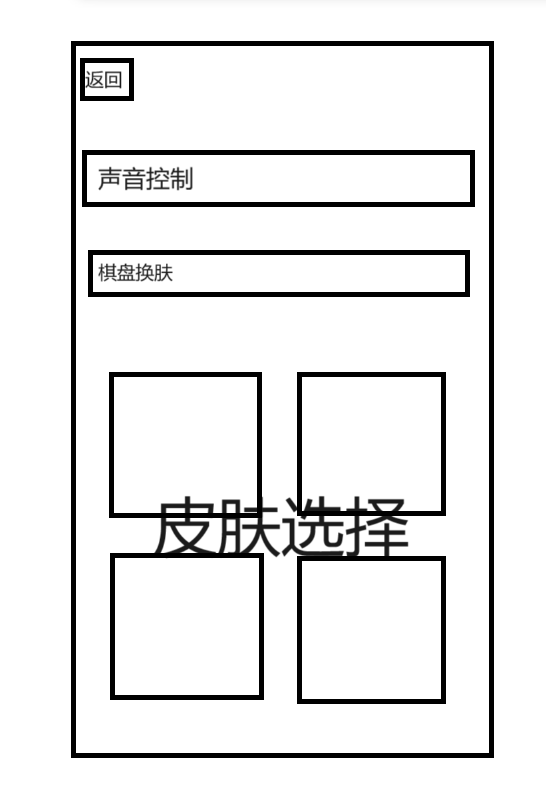
主页面设置了五个按钮，分别为联机模式、单机模式、开发者模式、以及退出。左上角为设置。

### 二、游戏页面



单机模式下，工具栏具有退出、重开、撤回、帮助功能，没有联机模式是因为通知我不要管（免责声明）

### 三、设置页面



声音控制可以通过滑动条控制声音大小，棋盘换肤提供四个皮肤，同时会改变回合显示颜色和工具栏颜色

### 四、帮助页面



帮助页面含有国际象棋教程，包括棋子认识，棋子走法与特殊规则。

## 棋盘层次结构

### 棋盘层次分为三层

1.最底层，隐藏层，z值为-2

2.底层，棋盘。Z值为-1

3.上层，棋子，及升变、胜利的对话框。z值为1

## 详细设计

### 一、走棋与悔棋实现

#### 1.移动相关

possibleMoves(): 得到棋子可走的位置

分不同类型的棋子，每种棋子普通的走法如下

pawn兵：第一次移动可以走两格，其余时候只能一次向前走一格，只能斜着一格吃子。查看前方一格的位置为空才可移动，判断左上右上有敌方棋子可吃子

bishop象：斜着走任意格子，吃子同理。分别循环遍历左上右上左下右下的位置是否为空，如果为空可移动。不为空，是己方棋子，不可再往此方向移动，退出循环停止遍历；是敌方棋子则可吃子，然后退出循环停止遍历

knight马：马只能走八个位置，x方向走两格后再向y方向走一格；y方向走两格后再向x方向走一格，即最多8种可能的移动

rook车：直着走任意格子，吃子同理。分别循环遍历上下左右的位置是否为空，如果为空可移动。不为空，是己方棋子，不可再往此方向移动，退出循环停止遍历；是敌方棋子则可吃子，然后退出循环停止遍历

queen后：结合象和车的走法，即可以走上下左右，左上右上左下的任意格子

king王：走"米"字型，即可以走上下左右，左上右上左下的一格

#### 2.特殊规则

兵升变: 己方兵走到对方的最后排位置上，必须晋升。可以选择把兵变成象、马、车、后的任意一种

吃过路兵: 白方兵到第三排，黑方兵到第六排，且左方一格或者右方一格有敌方的兵，则可以向该方向往前斜着移动一格，并吃掉这个敌方兵

王车易位: 前提条件是王没有移动过，且易位的王车之间没有别的棋子

短易位: 最右边的车没有移动过，则王可以向右移动两格，然后最右边的车自动移动到王的左边

长易位: 最左边的车没有移动过，则王可以向左移动两格，然后最左边的车自动移动到王的右边

**move()**: 移动后更新数据

判断移动后的位置为空，则更新该棋子视图

判断移动后的位置为不为空，即吃了敌方棋子，则移除被吃的敌方棋子，并更新该棋子视图

特殊：

吃过路兵的情况下，兵移动到空白位置后，敌方的兵自动消失

王车易位的情况下，王移动了两格后，易位的车自动变到王的旁边

#### 3.悔棋相关

**regretChess()**

QVector<QSharedPointer<Pieces>> m\_record 里面存储所有棋子的每一步位置。当一个棋子移动到空白位置，记录它移动前的位置。若移动到有敌方棋子的位置，即吃子，则先记录敌方棋子的位置，再记录它移动前的位置。

点击悔棋时，返回到m\_record的最后一条记录即可。若有吃子行为，m\_record倒数第二条记录中记录着被吃的棋子，将其插入到m\_pieces中(恢复被吃的棋子)，再返回m\_record的最后一条记录

### 二、联机实现

#### 1.概述

通过mynetwork.cpp实现数据传递。具体数据由五个数字组成。

#### 2.联机逻辑

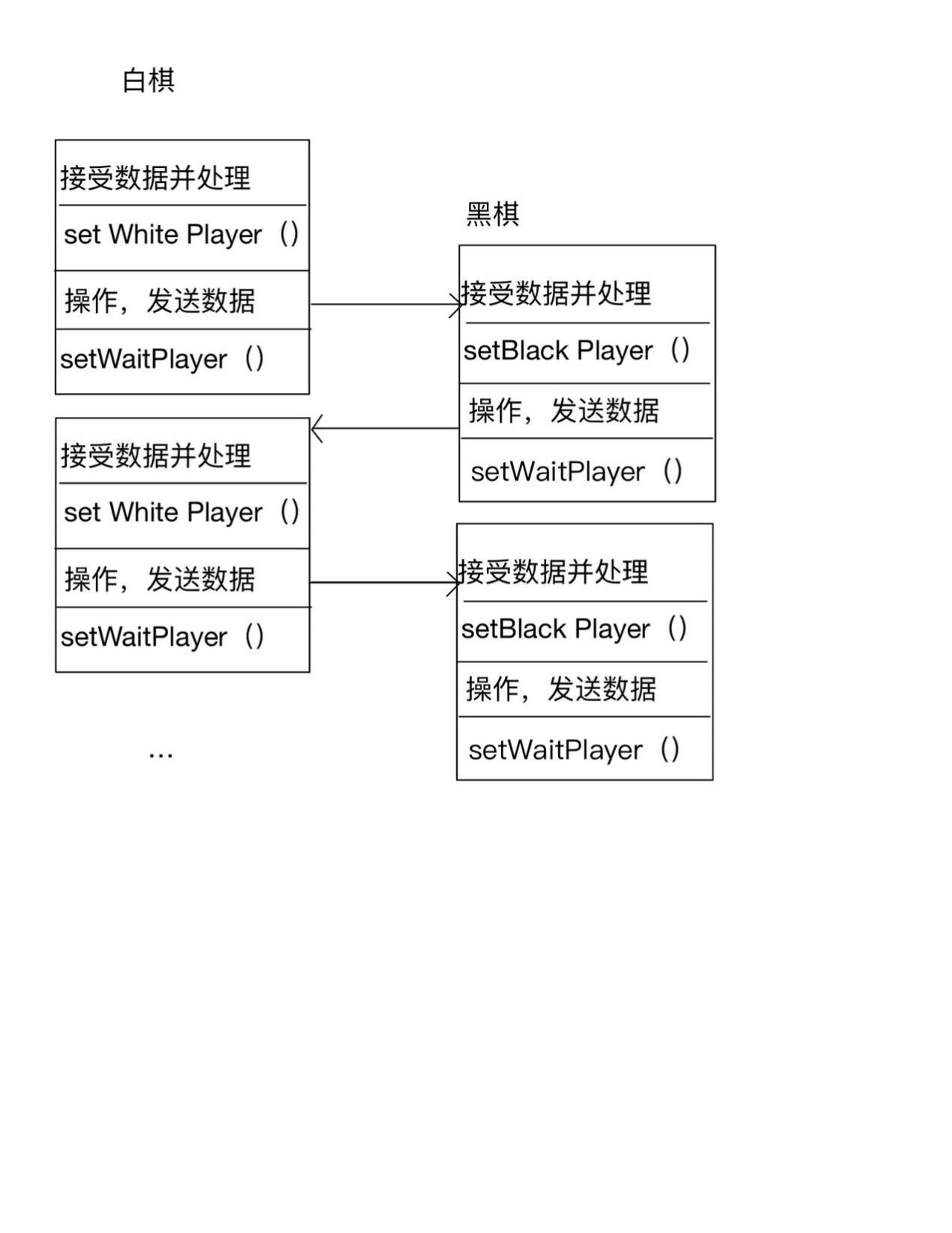
由于黑白两方应该是对等的，即不明确存在服务端、客户端。更类似一个打电话，相互倾听，相互讲话的过程。因此设计了NetWorkBlack.qml和NetWorkWhite.qml两个组件，如果把他们比作两部电话则mynetwork.cpp好比电话线。我们在一部电话发出声音，在另一部电话就会响起对应的声音，同理，我们在自己的棋盘上进行一步操作，会通过mynetwork.cpp在对方的棋盘上得到相同的操作，这样我们就好比是在同一个棋盘，联机对战的逻辑就转为了单机对战的逻辑。我们可以简要概述一下其内部流程为（己方操作，传递己方操作到对方。对方接受数据，对方操作后传递数据到己方。以此循环）。

通过qudp可以实现数据传递，在cpp文件与相应的qml中，通过myip4传递对方的ip地址（尽管myip4这个名字似乎有点不规范），通过字符串send和字符串recive分别传递要发送的数据和接受到的数据。

#### 3.关于操作轮次

1.我们用setWhitePlayer()、setBlackPlayer()、setWaitPlayer()三个函数控制对局时的轮次，来防止在白棋的一端去操纵了黑棋的棋，或者在黑棋的一端操作白棋的棋子。

2.开始时，白棋端设置为setWhitePlayer()即可以操作，黑棋端设置为setWaitPlayer()即不可操作或等待对方操作。

3.一方操作之后，将发送一次数据，并且被设置为setWaitPlayer()。另一方接收到数据处理后会由setWaitPlayer()变成对应的setWhitePlayer()或setBlackPlayer()，这样另一方又可以操纵一次。

#### 4.关于被传递的数据

1.字符串send和recive分别由5个数字组成。他们的初始值都是“-1 -1 -1 -1 0”。

2.前两位是被选择棋子移动前的xy坐标，第三四位是被选择棋子移动后的xy坐标。

3.特别的，在黑棋中第一位数字最开始用来判断连接是否成功，而在白棋中第二位数字最开始用来判断连接是否成功。

4.第五位是控制位，它分别代表决定了1悔棋、2重开游戏、3退出对战、5小兵升变为后、6小兵升变为车、7小兵升变为马、8小兵升变为象。

5.在qml中字符串send和recive的值分别暂时存储在id分别为\_inToSend、\_inToRecive的两个Text中。

#### 5.接收数据后的处理

接收数据后，我们对数据的处理和发送方的处理基本一样，但是要省略其可视化的过程来增强用户体验（比如对方选择小兵升变时，接收方只需要得到升变后的结果即可，而不需要同时弹出升变选择对话框）

### 三、页面相关实现

#### 1.页面运行顺序

启动后，Main.qml文件相关组件被创建，主窗口创建后，启动startTimer，播放启动动画，显示logo与文字。启动动画结束后，准备加载manage组件，之后加载manage组件，出现开始页面。

#### 2.页面跳转的实现

1.manage.qml，modes.qml,mode.qml为实现页面跳转的三大核心组件，其中，

manage是顶层组件，集成了 Modes 和多个 Loader 实例，用于管理和显示不同的模式及其对应的 QML 文件。

2.Modes是模式管理的核心，负责管理和切换不同的 Mode 实例。

3.Mode 是具体的模式实例，每个 Mode 都有一个唯一的名称，并包含激活和停用时的处理逻辑。

4.它们之间的运行逻辑可以比喻为：Modes作为管理员记录下每一个Mode，在实行页面跳转时，管理员Modes会帮助用户找到希望被激活的Mode，激活的Mode就会出现，如果你想回到之前的 Mode，就告诉 Manage 总指

挥，它会帮你回到上一个 Mode。

#### 3.页面返回的方法

在Manage.qml中，设置了onBack方法和goBack方法。onBack方法作用类似于书签，用于记录用户离开的页面（mode），goBack方法会根据书签标记进行返回处理。

##### 1.Modes中的方法

Modes组件完成加载后，调用reload()方法，整理所有添加的模式，同时调用clear()方法，忘记之前记住的模式。

之后，调用add()方法，加入需要管理的所有模式，其中，第一个有效的mode(start)会被设置为当前正在使用的mode。

之后，调用 mode.enter(noMode)启动第一个mode。

如果需要切换mode，调用set(mode)方法可选择指定mode。

##### 2.Mode中的方法

Mode组件完成初始化时，将 enter 信号连接到 onEnterHandler 方法，

将 leave 信号连接到 onLeaveHandler 方法。

当 when 属性变为 true 时，触发 onWhenChanged 信号处理器，记录当前模式并调用 mode.parent.set(name)。

mode.parent.set(name) 触发 enter 信号，调用 onEnterHandler 方法，输出激活信息。

当 when 属性变为 false 时，触发 onWhenChanged 信号处理器，调用 mode.parent.set(internal.backWhen, name)。

mode.parent.set(internal.backWhen, name) 触发 leave 信号，调用 onLeaveHandler 方法，输出停用信息。

#### 4.棋盘换肤的实现

Setting.qml文件中设置初始属性lightcolor与darkcolor，在选择按钮中，选择相应颜色，会将颜色值赋值。

在Manage.qml中，会判断Setting.qml是否成功加载。

如果 item 存在，连接 item的lightcolorChanged和darkcolorChanged信号到相应的处理函数。

item.lightcolorChanged信号在Setting.qml中的lightcolor属性发生变化时触发。

item.darkcolorChanged信号在Setting.qml中的darkcolor属性发生变化时触发。

通过 connect 方法，将这两个信号连接到处理函数，处理函数会将 Setting.qml 中的 lightcolor 和 darkcolor 属性值赋给 manage 对象的相应属性。

#### 5.音量控制的实现

在musicplayer.cpp中，存在setVolume(qreal volume)方法设置音量，并在音量改变时发出 volumeChanged 信号。

在Setting.qml中的Slider控件中，onValueChanged 信号处理器在滑块值改变时调用，将 musicPlayer.volume设置为滑块的当前值。

## 成员及项目任务大致分配

|  |
| --- |
| 付强：负责联机相关所有内容 |
| 唐婷婷：负责移动，悔棋单机对战 |
| 陈熙蕊：负责图形界面，所有UI |

## 六、问题分析及处理

|  |  |
| --- | --- |
| 遇到的问题 | 解决方法 |
| cpp数据和qml显示的连接 | board继承QAbstractListModel类，实现数据模型 |
| 点击棋子显示可走的位置 | qml中调用cpp的possibleMoves()方法，得到可走的位置，相应位置矩形变色 |
| 棋子移动图像更新 | 使用继承QAbstractListModel类的dataChanged()方法。只要数据发生更改，就会发出信号 |
| 悔棋逻辑实现 | m\_record存储所有棋子走过的每一步的位置 |
| 通过stackview实现多页面跳转时导致闪退 | 改用Loader实现页面之间的转换 |
| 设立背景音乐导致闪退 | 不通过QML语言实现播放，仅通过其实现音量控制功能，播放功能通过<QMediaPlayer>实现 |
| 换肤功能 选择默认色时无法变回默认色 | 将默认色改成了相似色 |

1. 可能出现的问题

悔棋记录的位置可能顺序有问题，恢复被吃的棋子是插入m\_pieces的末尾，而不是被吃之前所在的m\_pieces索引位置，这可能导致棋子数据出现混乱

在某种情况下，单机模式下返回键会突然无效，同时伴随回合显示异常，触发原因未知。

联机时，双方连接时间差异过长可能导致连接失败。