**程序设计开发实践报告**

**题 目 中国象棋**

**序 号 03**

**学 号 20232241091**

**姓 名 蔡佳兴**

**任课教师 范晓娅，于鲲鹏**

**成 绩**

大连理工大学软件学院

2024年7月

**目录**

[评分细则及标准 2](#_Toc143947979)

[2 项目开发报告 3](#_Toc143947989)

[2.1 项目简介 3](#_Toc143947990)

[2.2 需求分析 3](#_Toc143947991)

[2.2.1 功能模块 3](#_Toc143947992)

[2.2.2 需求说明 4](#_Toc143947993)

[2.3 系统设计 5](#_Toc143947994)

[2.3.1 界面设计（UI） 5](#_Toc143947995)

[2.3.2 流程设计 7](#_Toc143947996)

[2.4 系统实现 9](#_Toc143947998)

[2.4.1 实现效果 9](#_Toc143947999)

[2.4.2 核心代码 1](#_Toc143948000)1

[2.5 系统测试 15](#_Toc143948001)

[2.6 项目总结 16](#_Toc143948002)

**评分细则及标准**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **考察项目** | **总分** | **评分细则** | **分数** | **得分** |
| **技术调研（第一周）** | 20分 | 调研报告 | 10分 |  |
| 平时表现 | 10分 |
|  |  |  |  |  |
| **技术调研（第二周）** | 20分 | 调研报告 | 10分 |  |
| 平时表现 | 10分 |
|  | | | | |
| **项目开发评分（总分60分）** | | | |  |
| 问题规模 | 10分 | 创新超额完成指定任务，工作量饱满 | 8-10分 |  |
| 基本完成指定任务，工作量一般 | 6-8分 |
| 指定任务未完成，工作量不足 | 0-5分 |
| 技术难度 | 10分 | 模型设计合理优化，采用合适的方法实现 | 8-10分 |  |
| 模型设计基本正确，采用较合适的方法实现 | 6-8分 |
| 模型设计存在问题，采用方法不合理 | 0-5分 |
| 实现程度 | 20分 | 系统实现完整，界面友好，测试全面无误 | 8-10分 |  |
| 系统实现完整，界面友好，存在少许错误 | 6-8分 |
| 系统实现不完整，界面不友好，存在错误 | 0-5分 |
| 报告质量 | 10分 | 报告完整、格式统一、结构清晰、图表正确 | 8-10分 |  |
| 报告较为规范、结构较为清晰、图表基本正确 | 6-8分 |
| 内容不完整不规范、结构不清晰、图表有错误 | 0-5分 |
| 项目汇报 | 10分 | 阐述清晰准确，回答问题准确到位 | 8-10分 |  |
| 阐述基本准确，回答问题基本准确 | 6-8分 |
| 阐述不够清晰完整，回答问题不准确 | 0-5分 |
|  |  |  |  |  |
| 最终得分 | 100分 |  | | |

# 2 项目开发报告

## 2.1 项目简介

我开发的项目为中国象棋人人对弈游戏。其中包含象棋的基础规则、主页面，单人推演、双人博弈、退出游戏。进入游戏会有悔棋和认输，并在游戏结束后设有返回主页面的操作栏。

## 2.2 需求分析

### 2.2.1 功能模块

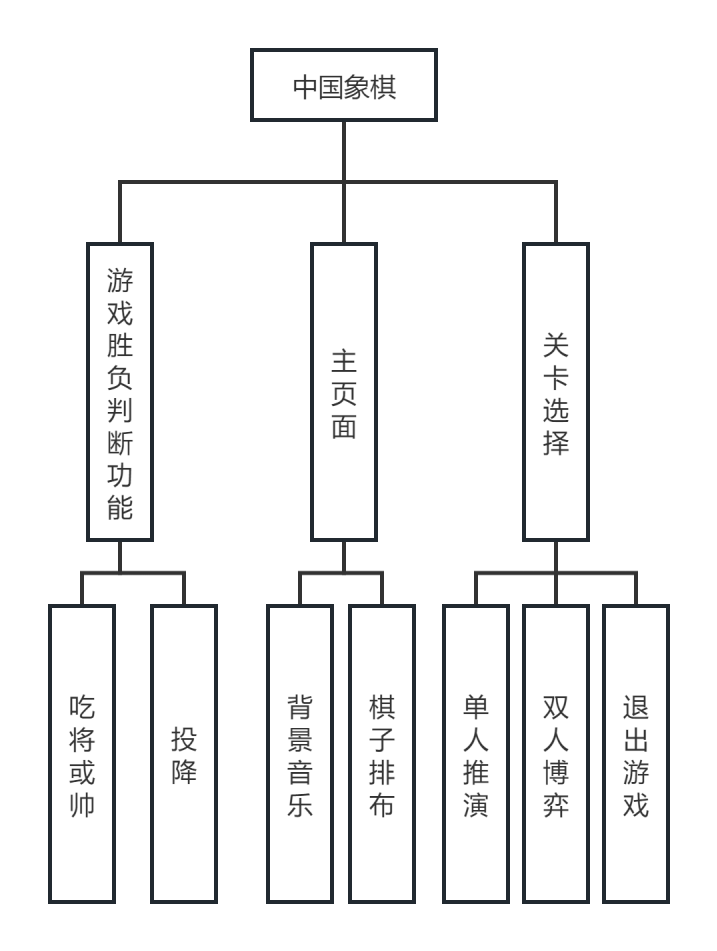


图2.1 功能模块图

### 2.2.2 需求说明

本开发为一款中国象棋游戏，包含中国象棋的基础且准确的玩法。

### 一、本开发游戏象棋的棋盘和棋子

　　象棋的棋盘是一个长方形，由十条横线和九条竖线组成，共有90个交叉点。棋盘的中间有一条分界线，称为“河界”，两边分别为“红方”和“黑方”。红方的棋子放在棋盘的下方，黑方的棋子放在棋盘的上方。

　　象棋的棋子共有32个，分为红黑两色。每方各有16个棋子，分别为：帅、仕、相、车、马、炮、兵(红方)和将、士、象、车、马、炮、兵(黑方)。

### 二、象棋每个棋子的走位规则

　　1. 将/帅：将帅是象棋中最重要的棋子，它们的走位规则是一样的，只能在九宫格内移动，不能出九宫格，每次只能移动一格。将帅不能直接对面，也不能在同一列上移动。

　　2. 士/仕：士仕是将帅的护卫，它们的走位规则也是一样的。士仕只能在九宫格内斜着走，每次只能移动一格，不能出九宫格。

　　3. 象/相：象相是象棋中的进攻棋子，它们的走位规则也是一样的。象相只能斜着走，每次只能走两格，不能越过棋子，且必须在己方半边棋盘内移动。而且，红方的象只能在黑色格子上移动，黑方的相只能在红色格子上移动。

　　4. 马：马是象棋中的特殊棋子，它的走位规则比较复杂。马走“日”字，即先向前或向后走一格，再向左或向右走两格，如果前方有障碍物则不能移动。马腿被挡住时，也不能移动。

　　5. 车：车是象棋中的攻击棋子，它的走位规则比较简单。车可以横着或竖着走，可以走任意格数，但不能斜着走，也不能越过棋子。

　　6. 炮：炮是象棋中的特殊棋子，它的走位规则也比较特殊。炮走法类似于车，可以横着或竖着走，也可以走任意格数，但是炮吃子时必须隔一个棋子，即必须有一个棋子在炮和被吃子之间，否则不能吃子。

　　7. 兵/卒：兵卒是象棋中的兵种，它们的走位规则也是一样的。兵卒只能向前走，每次只能走一格，但是在过了河之后，就可以左右前进了。兵卒过河后，可以横着走，但不能后退。

### 三、中国象棋的基本规则

　　1. 开局：红方先行，双方轮流走棋。

　　2. 棋子的移动：每个棋子都有其特定的走法，如帅只能在九宫内移动，将只能在九宫内移动等。棋子的移动不能超出棋盘范围，也不能穿过其他棋子。

　　3. 吃子：当一方的棋子移动到对方的棋子所在的位置时，就可以将对方的棋子吃掉。但是，有些棋子有特殊的吃子规则，如炮必须隔一个棋子才能吃掉对方的棋子。

　　4. 将军：当一方的将或帅被对方的棋子攻击时，就称为“将军”。被将军的一方必须立即采取措施来解除将军状态。

　　5. 战斗结果：当一方的将被对方吃掉时，就算输了比赛。如果一方的将被将军，但无法解除将军状态，则也算输了比赛。

### 四、象棋的基本技巧

　　1. 先控制中央：在开局阶段，应该尽量控制棋盘的中央，以便更好地展开攻势。

　　2. 防守为主：在比赛中，防守比进攻更为重要。如果没有足够的防守能力，就很容易被对方攻破阵势。

　　3. 利用棋子的特点：每个棋子都有其特定的走法和特点，应该根据棋子的特点来合理地运用棋子。

　　4. 观察对方的走法：在比赛中，应该经常观察对方的走法，以便及时采取反制措施。

　　5. 保持冷静：在比赛中，应该保持冷静，不要轻易地做出决策，否则可能会导致失误。

　　象棋是一种非常有趣的棋类游戏，需要一定的耐心和恒心来学习。通过掌握基本规则和技巧，初学者可以逐渐提高自己的象棋水平，享受到这个古老游戏的乐趣。

## 2.3 系统设计

### 2.3.1 界面设计（UI）



图2.2 界面原型图

### 2.3.2 流程设计

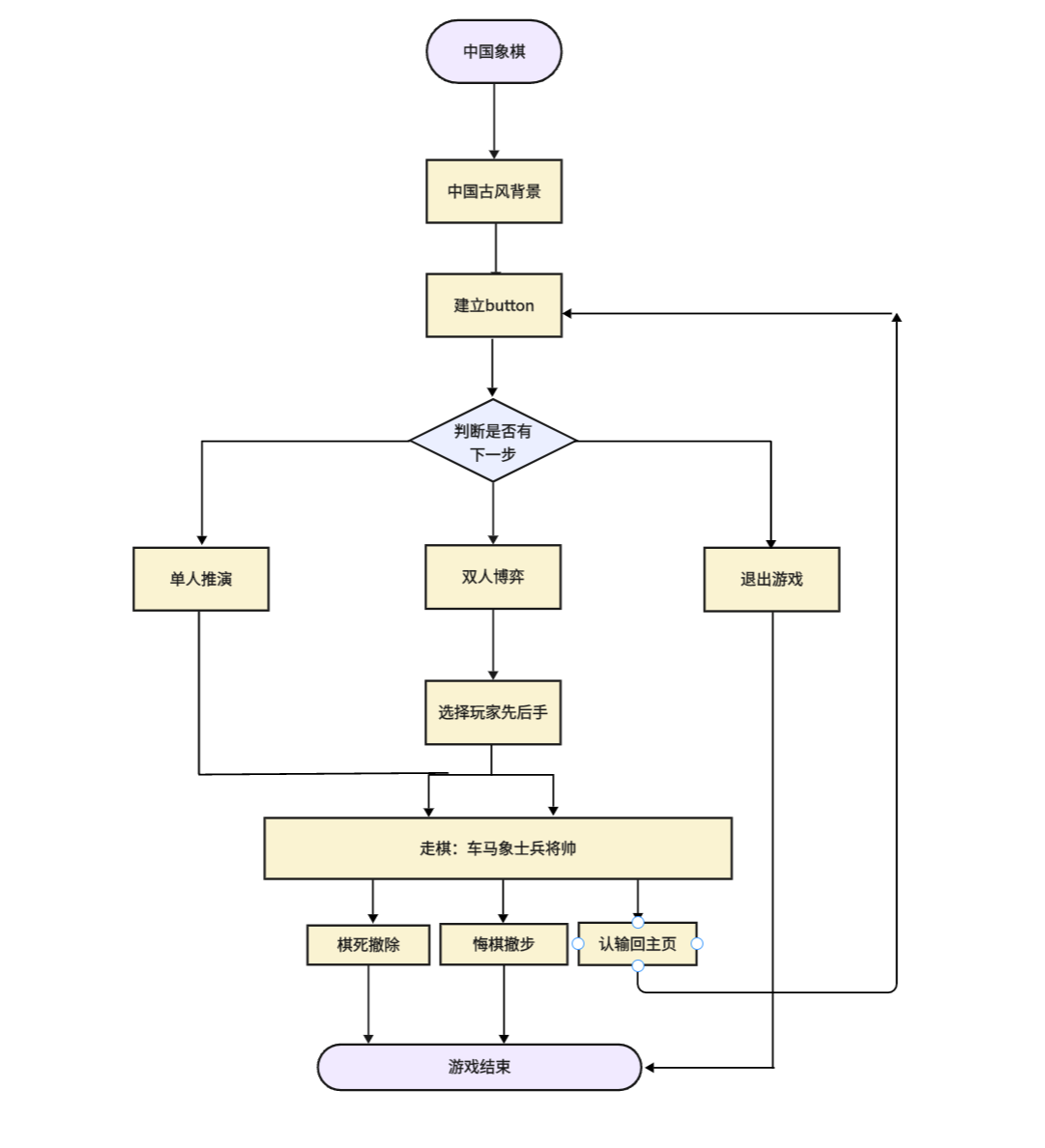


图2.3 系统流程图

绘制开始页面ui、棋盘ui、棋子等基础模型。

1、

void Board::**drawbotton**(QPainter& painter){

painter.setBrush(QBrush(QColor(242,247,247)));

painter.drawRect(d,10.8\*d,d\*2,d\*1);

painter.setPen(Qt::black);

painter.drawText(1.3\*d,11.55\*d, "悔棋");

painter.setBrush(QBrush(QColor(242,247,247)));

2、

if(\_win==0) {

painter.drawRect(7\*d,10.8\*d,d\*2,d\*1);

painter.drawText(7.3\*d,11.55\*d, "认输");

}

else if(\_win==-1) {

painter.drawRect(12\*d,0\*d,d\*3.7,d\*1.2);

painter.drawText(12.2\*d,0.9\*d, "返回主菜单");

}

}

3、

void Board::***isend***(){

if(\_s[8].\_dead){

v.voiceEnd();

QMessageBox::about(NULL, "游戏结束", "黑方胜利！");

\_win=-1;

return;

}

if(\_s[24].\_dead){

v.voiceEnd();

QMessageBox::about(NULL, "游戏结束", "红方胜利！");

\_win=-1;

return;

}

}

设置玩家先后手，P1、P2。

网上联机，cpp代码准备，输入计算机地址即可。

4、

void NetGame::***mouseReleaseEvent***(QMouseEvent \*ev){

QPoint pt = ev->pos();

if(pt.rx()>=12\*d && pt.rx()<=15.7\*d && pt.ry()>=0 && pt.ry()<=1.2\*d){

if(\_win==0) return;

emit this->endd();

}

if(\_win==-1) return;

if(pt.rx()>=d && pt.rx()<=3\*d && pt.ry()>=10.8\*d && pt.ry()<=11.8\*d){

*back*();

}

if(pt.rx()>=7\*d && pt.rx()<=9\*3\*d && pt.ry()>=10.8\*d && pt.ry()<=11.8\*d){

if(redside) NetGame::click(24,8,\_s[8].\_row,\_s[8].\_col);

else NetGame::click(8,24,\_s[24].\_row,\_s[24].\_col);

}

//将pt转化成象棋的行列值

//判断这个行列值上面有没有棋子

//如果有棋子就把它选中

## 2.4 系统实现

### 2.4.1 实现效果

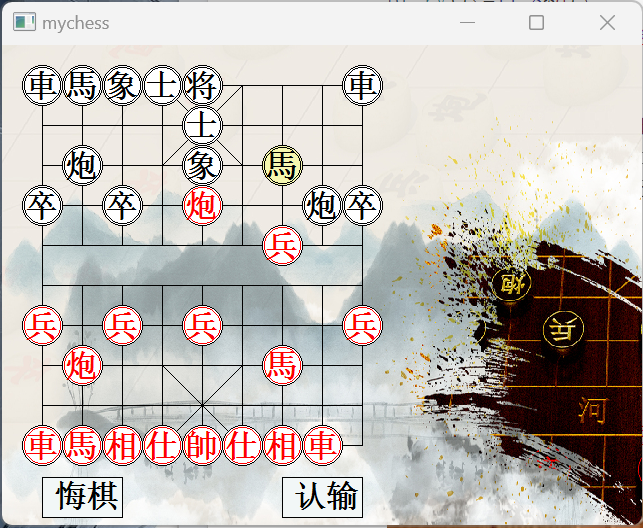


图2.1 游戏开始

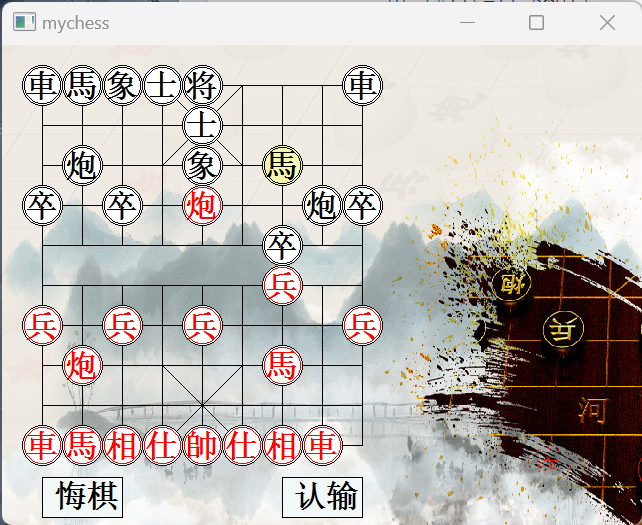


图2.2 游戏悔棋

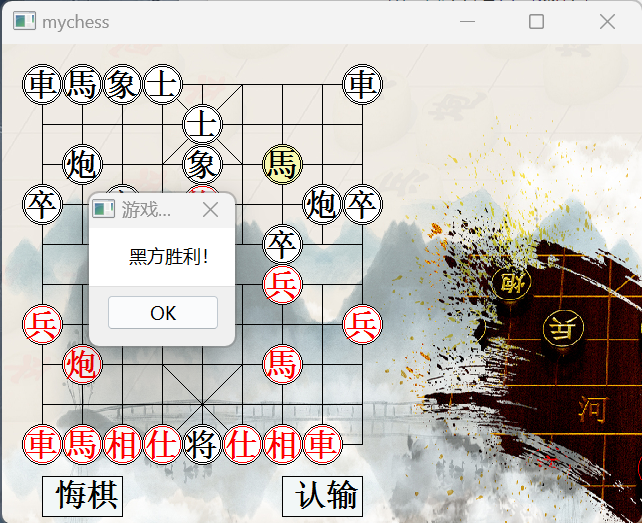


图2.3 胜利结算

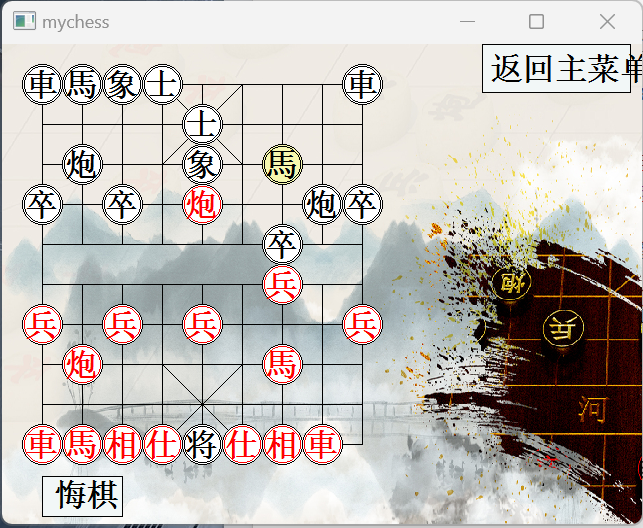


图2.4 可返回主菜单

### 2.4.2 核心代码

void Board::***paintEvent***(QPaintEvent \*){

QPainter painter(this);

painter.drawPixmap(0,0,width(),height(),QPixmap(":/new/prefix1/images/background.jpg"));

//画棋盘

for (int i=1;i<=10;i++){ //10根横线

painter.drawLine(QPoint(d,i\*d),QPoint(9\*d,i\*d));

}

for (int i=1;i<=9;i++){ //9根竖线

if(i==1 || i==9) painter.drawLine(QPoint(i\*d,d),QPoint(i\*d,10\*d));

else {

painter.drawLine(QPoint(i\*d,d),QPoint(i\*d,5\*d));

painter.drawLine(QPoint(i\*d,6\*d),QPoint(i\*d,10\*d));

}

}

painter.drawLine(QPoint(4\*d,d),QPoint(6\*d,3\*d));//老将九宫的斜线

painter.drawLine(QPoint(6\*d,d),QPoint(4\*d,3\*d));

painter.drawLine(QPoint(4\*d,8\*d),QPoint(6\*d,10\*d));

painter.drawLine(QPoint(6\*d,8\*d),QPoint(4\*d,10\*d));

//画棋子

for (int i=0;i < 32; i++){

drawStone(*painter*, i);

}

//QColor color(255, 0, 0);

drawbotton(*painter*);

}

void Board::**drawbotton**(QPainter& painter){

painter.setBrush(QBrush(QColor(242,247,247)));

painter.drawRect(d,10.8\*d,d\*2,d\*1);

painter.setPen(Qt::black);

painter.drawText(1.3\*d,11.55\*d, "悔棋");

painter.setBrush(QBrush(QColor(242,247,247)));

//

if(\_win==0) {

painter.drawRect(7\*d,10.8\*d,d\*2,d\*1);

painter.drawText(7.3\*d,11.55\*d, "认输");

}

else if(\_win==-1) {

painter.drawRect(12\*d,0\*d,d\*3.7,d\*1.2);

painter.drawText(12.2\*d,0.9\*d, "返回主菜单");

}

}

QPoint Board::**center**(int row, int col){

QPoint ret;

ret.rx()=(col+1)\*d;

ret.ry()=(row+1)\*d;

return ret;

}

void Board::**drawStone**(QPainter& painter, int id){

if(\_s[id].\_dead) return;

QPoint c = center(\_s[id].\_row, \_s[id].\_col);

QRect rect = QRect(c.x()-\_r,c.y()-\_r,d,d);

if(id==\_selectid) painter.setBrush(QBrush(QColor(245,247,175)));

else painter.setBrush(QBrush(Qt::white));

painter.setPen(Qt::black);

painter.drawEllipse(center(\_s[id].\_row, \_s[id].\_col), \_r, \_r);

if(\_s[id].\_red) painter.setPen(Qt::red);

painter.drawEllipse(center(\_s[id].\_row, \_s[id].\_col), \_r-2, \_r-2);

painter.setFont(QFont("MingLiU",\_r-4,QFont::Bold));

if(id<16)painter.drawText(rect,\_s[id].gettext1(),QTextOption(Qt::AlignCenter));

else if(id>=16)painter.drawText(rect,\_s[id].gettext2(),QTextOption(Qt::AlignCenter));

}

void Board::***back***(){

if(record[4]==-1){

QMessageBox::about(NULL, "警告！", "不能连续悔棋！");

return;

}

*click*(record[0],record[1],record[2],record[3]);

record[4]=-1;

update();

}

void Board::***mouseReleaseEvent***(QMouseEvent \*ev){

QPoint pt = ev->pos();

if(pt.rx()>=12\*d && pt.rx()<=15.7\*d && pt.ry()>=0 && pt.ry()<=1.2\*d){

if(\_win==0) return;

emit this->endd();

}

if(\_win==-1) return;

if(pt.rx()>=d && pt.rx()<=3\*d && pt.ry()>=10.8\*d && pt.ry()<=11.8\*d){

*back*();

}

if(pt.rx()>=7\*d && pt.rx()<=9\*3\*d && pt.ry()>=10.8\*d && pt.ry()<=11.8\*d){

if(redturn) *click*(24,8,\_s[8].\_row,\_s[8].\_col);

else *click*(8,24,\_s[24].\_row,\_s[24].\_col);

}

//将pt转化成象棋的行列值

//判断这个行列值上面有没有棋子

//如果有棋子就把它选中

int row, col, i, clickid=-1;

bool bRet=getRowCol(pt,*row*,*col*);

if(bRet==false) return;//点到棋盘外了

if(\_win==0) {

painter.drawRect(7\*d,10.8\*d,d\*2,d\*1);

painter.drawText(7.3\*d,11.55\*d, "认输");

}

else if(\_win==-1) {

painter.drawRect(12\*d,0\*d,d\*3.7,d\*1.2);

painter.drawText(12.2\*d,0.9\*d, "返回主菜单");

}

}

void Board::***isend***(){

if(\_s[8].\_dead){

v.voiceEnd();

QMessageBox::about(NULL, "游戏结束", "黑方胜利！");

\_win=-1;

return;

}

if(\_s[24].\_dead){

v.voiceEnd();

QMessageBox::about(NULL, "游戏结束", "红方胜利！");

\_win=-1;

return;

}

}

Voice::**Voice**(QObject \*parent) : QObject(parent){}

void Voice::**voiceSelect**()

{

if(\_select!= nullptr)

this->\_select->play();

}

void Voice:: **voiceMove**()

{

if(\_move!= nullptr)

this->\_move->play();

}

void Voice:: **voiceEat**()

{

if(\_eat!= nullptr)

this->\_eat->play();

}

void Voice:: **voiceEnd**()

{

if(\_end!= nullptr)

this->\_end->play();

}

void Voice:: **voicejiangjun**(){

if(\_jiangjun!= nullptr)

this->\_jiangjun->play();

}

void NetGame::***back***(){ //悔棋

if(record[4]==-1){

QMessageBox::about(NULL, "警告！", "不能连续悔棋！");

return;

}

*click*(record[0],record[1],record[2],record[3]);

record[4]=-1;

update();

}

## 2.5 系统测试

表2.5 系统测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项 | 测试步骤 | 输入信号 | 预期结果 | 测试结果 |
| 1 | 双人博弈 | 点击游戏界面“双人博弈”项 | 鼠标 | 进入游戏，界面中出现P1，P2玩家先后选项 | 通过 |
| 2 | 移动棋子 | 点击棋子，按照棋子的走棋规则 | 鼠标 | 走起成功，棋子按规定路线行进，吃棋正常 | 通过 |
| 3 | 网上联机 | 本机地址，手机热点，校园网 | 无 | 连接成功 | 校园网未通过 |
| 4 | 悔棋 | 点击“悔棋” | 鼠标 | 黑红棋均可毁一步棋 | 通过 |
| 5 | 悔棋 | 连续点击“悔棋” | 鼠标 | 警告，不可悔棋 | 通过 |
| 6 | 认输 | 点击“认输” | 鼠标 | 该黑棋走，黑棋认输 | 通过 |
| 7 | 认输 | 点击“认输” | 鼠标 | 该红旗走，红旗认输 | 未通过 |
| 8 | 退出游戏 | 点击“退出游戏” | 鼠标 | 退出游戏 | 通过 |
| 9 | 返回主菜单 | 点击“返回主菜单” | 鼠标 | 回到游戏主页面 | 通过 |
| 10 | 单人推演 | 点击“单人推演” | 鼠标 | 可自行进行互博 | 通过 |
| 11 | 声音 | 无 | 无 | 主页面，落子均有音效 | 通过 |
| 12 | 声音 | 将军、绝杀时，出现音效 | 无 | 将军或绝杀会出现提示音 | 通过 |

## 2.6 项目总结

中国象棋的整体完成多借鉴网上棋子行走代码，主题背景、音效来源于现学现做。技术上的难点多来源于现学。

位置估值

相同棋子在不同位置时，作用是不同的，比如兵过河后更有价值，越靠近九宫格，对将帅的威胁越大，其位置估值也越大。而炮在本方的价值一般大于对方区域，马在布局后期的价值大于前期的价值。

棋子的灵活性每个棋子的灵活性是不同的，理论上车到达的位置多，但是如果车不从初始位置出来的话，他对棋局的贡献是很小的，需要给评价函数加入棋子灵活性，具体值由棋盘动态变化的过程决定。

威胁与保护在象棋中，每个棋子之间都是牵一发动全身的，他们之间直接或间接存在某种威胁和保护关系。需要明确的是，评价函数只是对某个静态的局面做出评价，而不是对动态的变化过程做出评价。为了寻找某个棋子的保护关系，就要遍历己方所有存在的棋子；同样为了寻找某个棋子的威胁关系，就要遍历对方所有存在的棋子。除了上述的评价关系之外，象棋中还存在特殊的形，例如连环马、马后炮等，也要在评价函数中增加额外的分值，另外还要考虑炮和对方的将的关系等。如果能够考虑到象棋中存才的特殊的形，那对局面的把握更准确。

更多的，在人人对战的基础上，加上TCP通信，用于实现数据的同步。QTcpServer用于创建TCP服务器端。当newCon-nection信号激发时，我们调用指定的槽函数创建一个通信的套接字。QTcpSocket用于创建TCP通信套接字。当connected信号激发时，我们向服务器端发送消息，当readyRead信号激发时，我们就可以读取数据。客户端与服务端通信时用数组buf来发送数据。