

面向对象程序设计 课程大作业

五子棋程序开发报告

—— 浙江大学 ——

课程时间 | 2017 年 · 春学期

指导 教师 | 陈 奇

—— 组 员 ——

夏 涵
赫 炎
宋斯亮

目录

一、引言.....	3
1.1 编写目的.....	3
1.1.1 编写目的说明.....	3
1.1.2 目的.....	3
1.2 背景.....	3
二、需求分析.....	3
2.1 用户注册与登陆功能.....	3
2.2 游戏功能.....	4
2.2.1 人机对战.....	4
2.2.2 双人对战.....	4
2.3 排名系统.....	4
三、状态图.....	5
3.1 总状态图.....	5
四、类图及模块功能.....	5
4.1 Mainwindow	5
4.2 ChessBoard.....	7
4.3 Chessman	7
4.4 enroll	8
4.5 GLOBALA	9
4.6 Login	10
4.7 myTimer	12
4.8 Sky_Ladder	12
4.9 User	13
4.10 Welcome	13
4.11 Win1/2_show	14
五、测试.....	15
5.1 测试概要.....	15
5.2 测试环境.....	15
5.3 部分测试结果.....	15
5.4 最终版测试.....	16
六、任务分工.....	19
6.1 代码部分.....	19
6.2 文档	19
七、感想.....	20

一、引言

1.1 编写目的

1.1.1 编写目的说明

本报告的编写主旨在于规范具有双人对战和人机对战两种对战模式的五子棋游戏的编写与实现。本册说明了五子棋游戏的主要功能,明确表示各功能模块的实现过程以及最终的实现要求,阐明了为不同用户提供的权限,模拟了游戏玩家的功能需求,为软件的开发与使用提供了一个规格标准。对于开发人员来说,本册提高了软件开发过程的能见度,便于软件开发人员进行管理与监控,同时也可作为开发者与用户间的沟通手段。

1.1.2 目的

- 1) 作为五子棋游戏,为开发设计提供参考与规范。
- 2) 规范游戏开发各个模块间的接口,便于开发者之间的整合。
- 3) 为游戏的开发设计提供参考与规范。
- 4) 作为测试与验收的依据。

1.2 背景

软件名称	五子棋
任务提出者	浙江大学面向对象程序设计课程任课老师——陈奇
开发者	夏涵, 宋斯亮, 赫炎
用户	游戏玩家
系统架设环境	带有Qt Creator软件的PC机
其它	玩家在游戏中可以选择与AI进行对战,来提升自己的五子棋能力;也可以注册并登录账号,与好友进行本地双人对战,并查看自己的积分名次。

二、需求分析

2.1 用户注册与登陆功能

本软件是一个包含人机对战与双人对战两种功能的游戏,因此,需要包含用户的注册和登陆功能。在用户发送一个注册请求时,程序要及时响应并且确认注册成功与否,并发送返

回值。而当用户发送了一个登陆请求时要能够及时的对用户名及密码做出判断，并发送成功与否的标记和当前登陆用户的信息。

2.2 游戏功能

本游戏包含了人机对战和双人对战两种游戏功能，在主界面可以选择要进行的对战方式。在对战结束后，程序会弹出一个窗口，其中包含了胜负信息，并且玩家可以选择重新开始或者退出对

2.2.1 人机对战

在本五子棋游戏中，玩家可以选择与AI进行对战，来提升自己的水平。在人机对战中，AI默认为黑子先手，且第一子会落在棋盘的正中央。在此界面中，没有玩家的信息，因为进入此界面不需要登录；而且人机对战界面没有倒计时，玩家可以花无限时间来进行思考，与AI博弈。

2.2.2 双人对战

在五子棋游戏中，最重要的一个部分就是人和人之间的博弈了。在进行对战之前，首先进入的是一个登陆界面，要求对战双方都进行登陆。登陆成功后，即进入了双人对战界面。界面的左边是与人机对战界面相同的棋盘，右边包含了双方用户的基本信息，倒计时以及悔棋和认输功能。双方都有有限次悔棋机会，用光后无法继续悔棋。使用认输功能，就会立刻判定对方获胜。

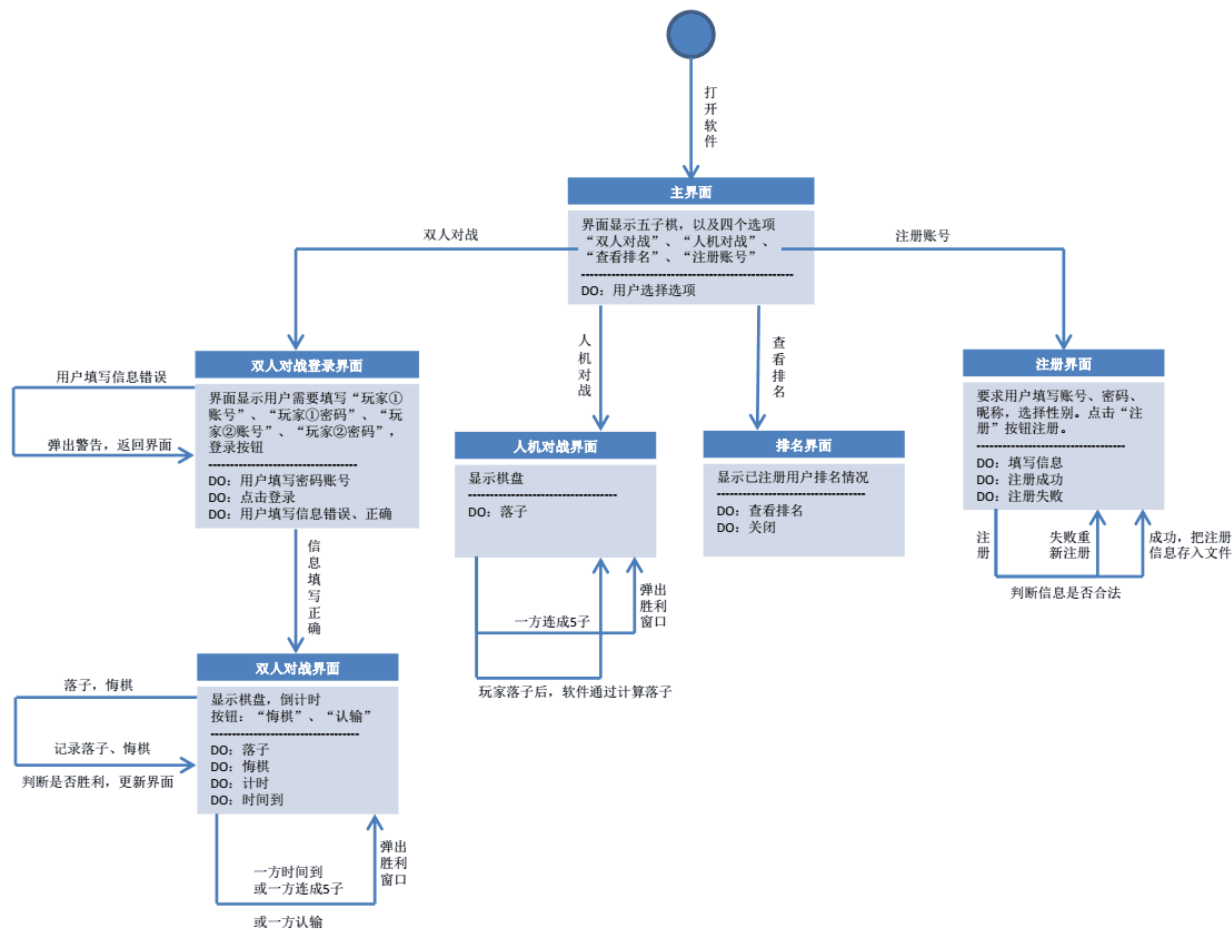
在一局结束后，双方的信息会发生相应的变化，其中包括积分的加减，总游戏局数的增加以及胜利局数的改变。

2.3 排名系统

为了反映玩家之间的强弱，以增加此游戏的竞技性，一个排名系统是很有必要的。本游戏提供了一个天梯排行，按积分顺序显示前五名玩家的昵称和积分。

三、状态图

3.1 总状态图



四、类图及模块功能

4.1 Mainwindow

下棋主界面（人机，人人对战共用）

MainWindow

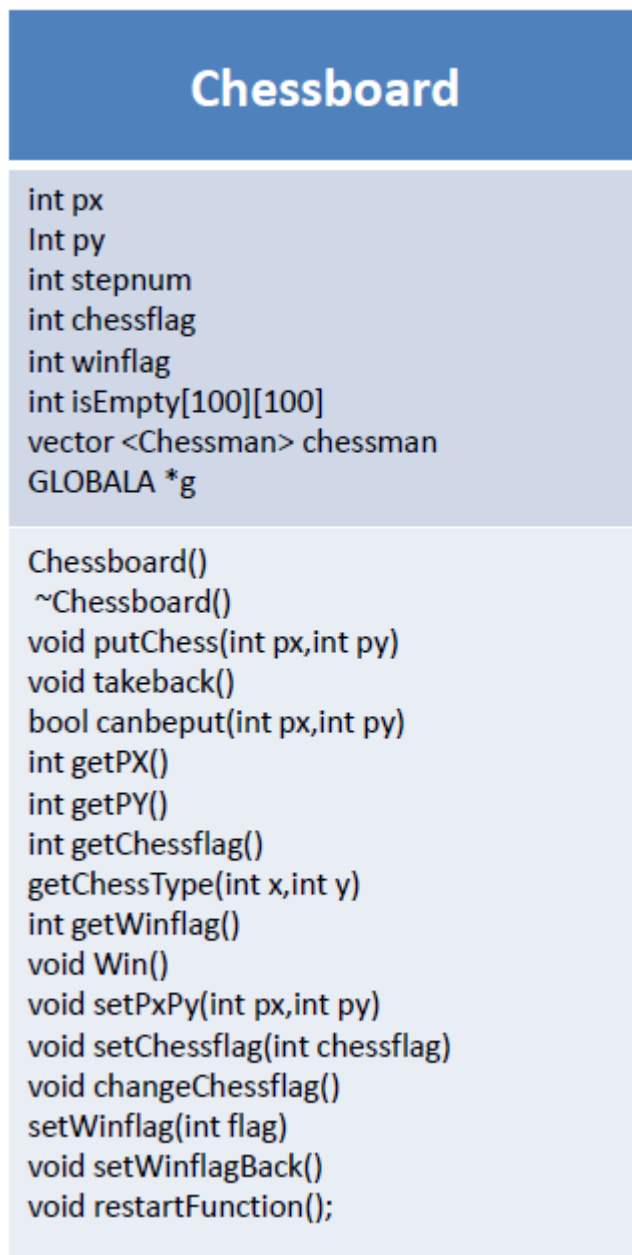
```
Int flag
GLOBALA *g
GLOBALA *g
win1_show *win1
win2_show *win2
User *u
QString UserName1
QString UserName2
mytimer *blackTime
mytimer *whiteTime
mytimer *blackRoundTime
mytimer *whiteRoundTime
QTimer *WhTimer
QTimer *BlTimer
QTimer *WhRoTimer
QTimer *BlRoTimer

explicit MainWindow(QWidget *parent = 0)
~MainWindow()
void paintEvent(QPaintEvent*)
void mouseReleaseEvent(QMouseEvent
*event)
void on_Takeback_Player2_clicked();
void on_Takeback_Player1_clicked();
void on_Giveup_Player2_clicked();
void on_Giveup_Player1_clicked();
void quit_Slot()
void restart_Slot()
void WhTimer_Slot()
void BlTimer_Slot()
void WhRoTimer_Slot()
void BlRoTimer_Slot()
```

通过全局变量（后面全局类介绍），控制同一个下棋主界面的锁定窗口大小展示不同的下棋模式（人机，人人）界面，控制程序运行的模式（人人，人机对战）。两个 win1/2_show 通过各种胜利判断条件（超时判负，连成 5 子）来弹出提示胜利信息。timer 相关是四个部分的计时器（黑/白棋的总/本轮剩余时间）。使用 mousetoreleaseevent 函数找到落子位置。通过 quit/restart SLOTS 进行重置/退出对战。两个 Username 是从全局类得来的，用来作为参数传入 User 类中，获得现在登录的双方玩家的昵称，积分，场次等信息。

4.2 ChessBoard

棋盘类，存储棋盘状态



通过一个二维数组 `isEmpty` 去存储棋子类型和位置，`chessflag` 0, -1, 1 分别代表没有棋子，黑子，白子状态。`winflag` 通过判断胜利的函数更改，表明胜利状态。成员函数用来实现落子，判断胜利条件是否满足，悔棋，重置棋盘等功能。需要说明的是，我们在当前以及接下来的类图中，会有部分的成员变量或者成员函数没有被使用过。这些变量或者函数在将来如果要添加需求时，可能会起到作用，在这里由于没有影响程序运行，我们就保留在了代码中。

4.3 Chessman

棋子类

Chessman
<pre> int chessflag2 int x int y int No_step </pre>
<pre> Chessman(int x,int y,int chessflag2,int No_step) ~Chessman() int getX() int getY(); </pre>

这个类设计出来之后并没有在本次程序中起到作用，但是作为以后扩展，还是可以添加很多功能的。

4.4 enroll

注册界面

enroll : QDialog
<pre> Ui::enroll *ui GLOBALA * User *u=new User enroll_success *es enroll_success *er enroll_success *er2 enroll_success *er3 enroll_success *er4 QString UserName QString Password QString NickName QString gender QString id_for_user_image QString total_score_earned QString total_game_played QString total_wins </pre>
<pre> void paintEvent(QPaintEvent*) explicit enroll(QWidget *parent = 0) ~enroll() void on_Register_clicked() void on_Boy_clicked() void on_Girl_clicked() void quit_slot() </pre>

enroll_success : QDialog

Ui::enroll_success *ui

```
explicit enroll_success(QWidget *parent = 0)
~enroll_success()
void on_pushButton_clicked()
quit_signal()
```

enroll_warning1/2/3/4: QDialog

Ui::enroll_warning1/2/3/4*ui

```
explicit enroll_warning1/2/3/4(QWidget
*parent = 0)
~enroll_warning1/2/3/4()
void on_Bcak_clicked()
```

注册界面主要包含了一些判断条件，通过调用 `user` 类中相关函数，判断注册的用户名或者昵称是否重复，是否包括一些非法字符（比如我们在存储数据的时候用逗号分隔），是否有信息未填写，并发出相应的弹窗警告。若成功，则调用 `Uer` 成员函数进行注册信息写入。

4.5 GLOBALA

全局变量存储类

末尾的 `A` 为了区分关键字

GLOBALA

```
static QString UserName1
static QString UserName2
static int modeflag

explicit enroll_warning3(QWidget *parent = 0)
~enroll_warning3()
static void setUsername1(QString)
static QString getUsername1()
static void setUsername2(QString)
static QString getUsername2()
static void setModeflag(int)
static int getModeflag()
```

两个 username 是为了存储登陆成功时候双方玩家的用户名，这样在 mainwindow 中调用 user 的函数将 username 信息传入则会获得相应的用户信息返回值。modeflag 是为了判断用户选择的 game 模式。

4.6 Login

登录界面

login : QDialog

```

Ui::login *ui; void
paintEvent(QPaintEvent*)
User *u
MainWindow *mw
login_warning *lw
login_warning2 *lw2
login_warning3 *lw3
login_warning4 *lw4
GLOBALA *g
QString UserName1
QString UserName2
QString Password1
QString Password2

explicit login(QWidget *parent = 0)
~login()
static void setUserName1(QString)
static QString getUserName1()
static void setUserName2(QString)
static QString getUserName2()
static void setModelflag(int)
static int getModelflag()
void on_LoginConfirm_clicked();

```

Login_warning1/2/3/4 : Qdialog

```

Ui::login_warning 1/2/3/4*ui

explicit login_warning1/2/3/4(QWidget
*parent = 0)
~login_warning1/2/3/4()
on_Back_clicked()

```

登录界面有四个文本框输入窗口，分别用来输入两个玩家的用户名和密码。进行如下检查：判断是否所有信息填写完毕，判断用户名密码是否正确，判断是否两个玩家信息属于同一个人，弹出相应警告。

4.7 myTimer

计时相关

mytimer
<pre>int min; int sec QString minStr QString secStr int cnt</pre>
<pre>void setStrLength(QString *str, int length) mytimer() void reset(int min,int sec) QString getjointTime() int getSEC() void internalCalulateSlot()</pre>

计时器函数实现，主要功能实现计时，补零，字符串拼接功能；

4.8 Sky_Ladder

天梯界面

Sky_Ladder : QWidget
<pre>Ui::Sky_Ladder *ui User *u=new User QStringList SkyLadder</pre>
<pre>explicit Sky_Ladder(QWidget *parent = 0) ~Sky_Ladder(); void paintEvent(QPaintEvent*)</pre>

显示天梯排名（注意这个的内部实现只能显示运行 exe 那是的排名，比如你刚下完一盘棋时候得分并不会马上显示在天梯上）

4.9 User

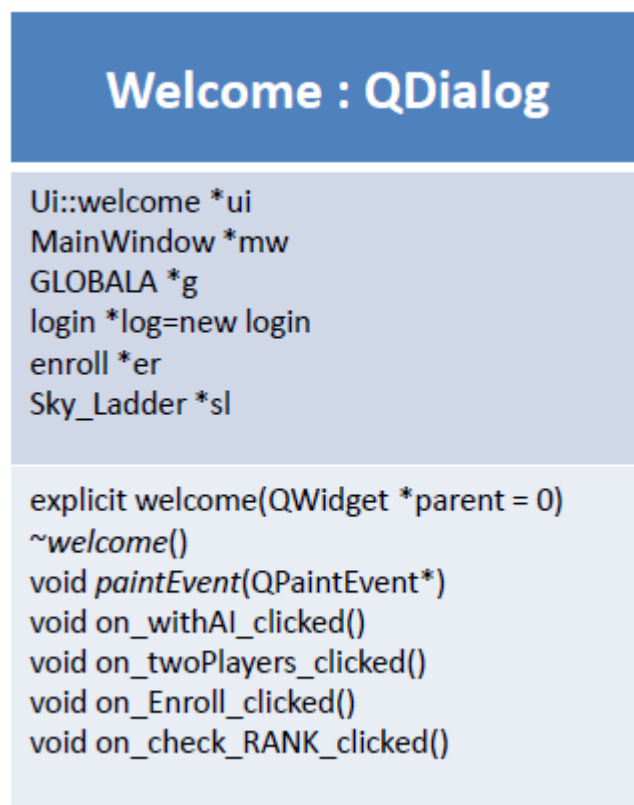
用户类

User	
<pre>int rtime int checkflag QString Filename QString WriteLine QString Line QString Usertemp QString Passwordtemp QString Username QString Password QString nickname QString gender QString id_for_user_image int total_score_earned int total_game_played int total_wins QString C_Username QString C_Password QString C_nickname</pre>	<pre>QString C_gender QString C_id_for_user_image int C_total_score_earned int C_total_game_played int C_total_wins int Score int Games int Wins QString data_added QString zero QString no QString strAll QStringList strList QStringList RegistList; QStringList sortedfile QStringList unsortfile QStringList sky_ladder_list;</pre>
<pre>User() ~User() int regrettime() void add() QString Check_nickname(QString) QString Check_gender(QString) QString Check_id_for_user_image(QString) int Check_total_score_earned(QString) int Check_total_game_played(QString) int Check_total_wins(QString) QStringList sky_ladder() bool Login(QString,QString) void regester(QString Username, QString</pre>	<pre>Password, QString nickname, QString gender, QString qqid, QString score_earned, QString game_played, QString wins) void addWinTimes(QString,int) void addTotalGames(QString,int) void addScore(QString,int) void Readallfile() void check_change_file(QString) int isRegistered(QString Username, QString nickname) int includecomma(QString username, QString password, QString nickname) QStringList sortfile()</pre>

用户类，实现跟用户相关的操作，比如，注册，检查登录信息，排序，根据用户名返回相应用户信息等。这里面还有悔棋记录次数变量和函数，但是最后由于时间问题虽然已经在类中实现了，但是并没有将悔棋次数限定加到 `mainwindow` 中去真正实现。

4.10 Welcome

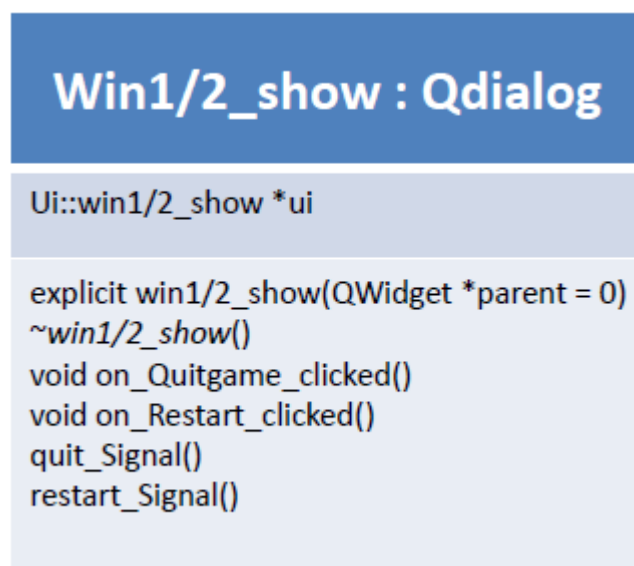
初始界面



初始进入的界面。导航包括，双人对战，人机对战，天梯排名，注册用户。

4.11 Win1/2_show

弹窗显示获胜信息



在 mainwindow 中根据获胜情况弹出相应获胜信息。并且提供了退出游戏

五、测试

5.1 测试概要

测试范围
软件的注册、登录、天梯排名、人机对战、双人对战
测试方法
功能测试：软件开发者针对某一特性功能而进行的测试
随机测试：软件开发者和用户没有针对性地反复使用本软件

5.2 测试环境

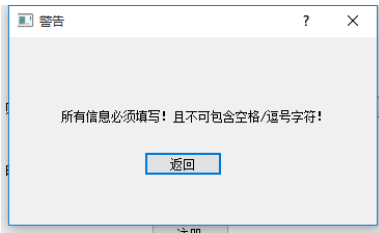
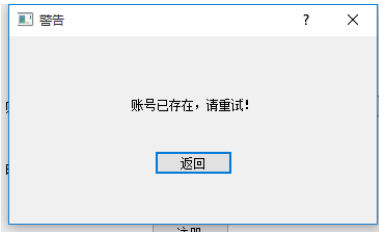
计算机配置			
计算机名(IP)	CPU	内存	软件环境(操作系统，应用软件)
MyPC	I7	8G	Windows10, Qt Creator 4.2.1

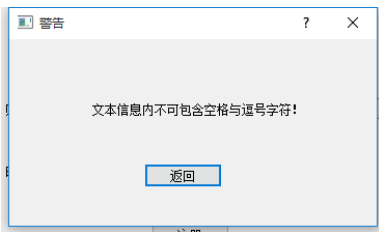
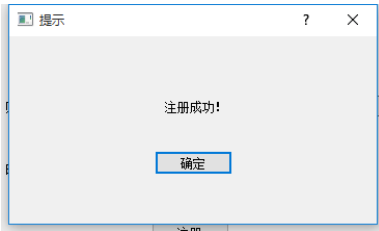
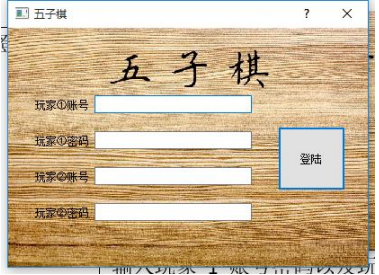

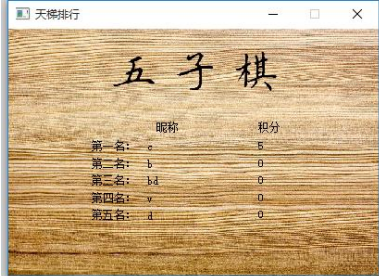
5.3 部分测试结果


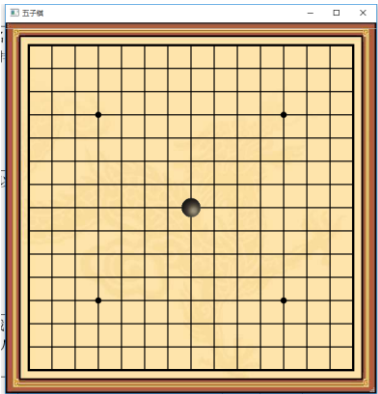
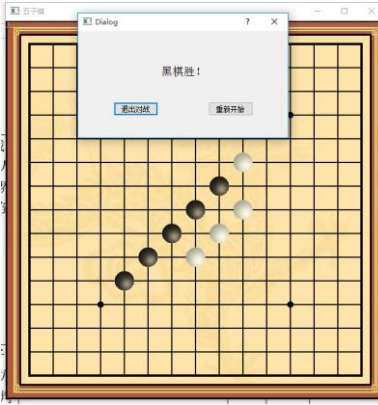
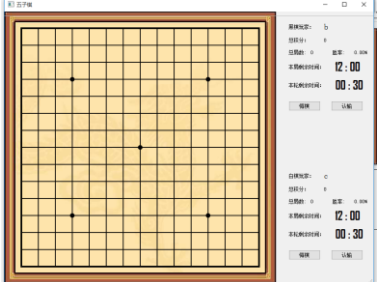
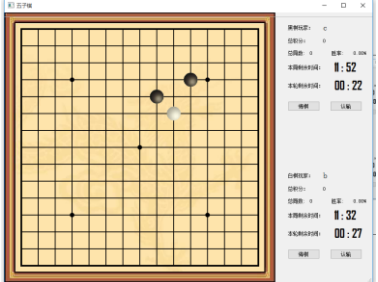
模块名称	编号	简要描述	分析结果	解决方法	最终结果	
					已解决	未解决
注册	01	如果注册信息中包含英文逗号，此用户信息登录的时候会查不到	数据库文件中是以逗号作为分隔符的，如果用户加入逗号，在读取信息的时候就会乱套。	如果用户输入的信息中包含逗号，则认为输入信息无效并弹出提示。	<input checked="" type="checkbox"/>	<input type="checkbox"/>
登录	02	如果注册用户的账号昵称为“2”，再注册一个昵称为“3”的账户会提示已被注册。	在判定是否有账户时，要判断“2，”后面的语句是否相同，因为数字加逗号是作为分隔符用的。所以如果有一个账号昵称为“2，”后面跟着的必然是“3”，所以程序会判定账户	修改判断程序，不用数字来判断，而只用“，”作为判断符。	<input checked="" type="checkbox"/>	<input type="checkbox"/>


			已存在。			
查看排名	03	如果注册用户不到 5 个人，则程序会崩溃。	给用户进行排名的程序会把排好名的 QStringList 传给显示函数，而如果此列表数目小于 5，那么当显示函数要求调出列表的第五个值的时候，就会出现溢出异常。	在排名程序中，如果总数小于 5，就会加入“无”作为填充。	<input checked="" type="checkbox"/>	<input type="checkbox"/>
人机对战、双人对战	04	当一局对战结束后，如果选择退出对战，则会回到初始界面；此时如果再次进入对战，棋局会是上一局结束时的样子，没有重置。	忘记在 win1/2_show 弹窗的退出游戏按钮的 slot 里更新棋盘了。	在 quit_SLOT 里加上更新棋盘函数。	<input checked="" type="checkbox"/>	<input type="checkbox"/>

5.4 最终版测试

功能	基本要求	最终结果	测试通过	
			是	否
注册	在初始页面单击“注册账号”按钮，进入注册界面；输入账号、密码、昵称，选择性别。如果有至少一个没有填写，则出现提示“所有名称必须填写”；		<input checked="" type="checkbox"/>	<input type="checkbox"/>
	如果全部填写，但输入的账号或昵称与数据库中已有的信息重复，则提示“账号已存在”或“昵称已存在”；		<input checked="" type="checkbox"/>	<input type="checkbox"/>

	如果填写的内容中包含英文逗号或空格，则提示“文本信息内不能包含空格或逗号字符”；		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	如果输入的信息没有以上情况，则输入有效，单击“注册”按钮后，提示“注册成功”，用户数据被存入数据库文件中，单击“确定”后返回初始页面。		<input checked="" type="checkbox"/>	<input type="checkbox"/>
登 录	在初始页面单击“双人对战”按钮，进入登录界面。		<input checked="" type="checkbox"/>	<input type="checkbox"/>
	输入玩家 1 账号密码以及玩家 2 账号密码。如果数据库中的信息无法与其中一个玩家的账号与密码相匹配，则提示“玩家 1（或 2）的信息不存在 或 密码错误”。		<input checked="" type="checkbox"/>	<input type="checkbox"/>
	如果玩家 1 和玩家 2 输入的信息相同，则提示“想和自己下棋么？别做梦了~”。		<input checked="" type="checkbox"/>	<input type="checkbox"/>
	如果玩家 1 和玩家 2 的信息均有效，则进入双人对战界面。	正常进入双人对战界面。	<input checked="" type="checkbox"/>	<input type="checkbox"/>
查 看 排 名	在初始界面单击“查看排名”按钮，进入排名界面，显示排名前五的用户信息。		<input checked="" type="checkbox"/>	<input type="checkbox"/>

	如果总用户数不足 5 个，则以“无”来补全。		☒	☐
人 机 对 战	在初始界面单击“人机对战”按钮，无需登录即可进入人机对战界面。在人机对战界面中，只有一个棋盘，没有倒计时等信息。		☒	☐
	与 AI 进行对战。AI 先手黑子，第一子落在正中央。当一方的 5 个棋子连成一线，系统判定胜负，并弹出提示，可以选择退出对战或是重新开始。若选择“退出对战”，则返回初始界面；若选择“重新开始”，则回到 AI 先手黑子，第一子落在正中央的棋局。		☒	☐
双 人 对 战	在初始界面中单击“双人对战”按钮，进入登录界面；输入正确的两个用户信息，进入双人对战界面。 双人对战界面的左边为棋盘，右边为用户信息以及倒计时，悔棋和认输功能。规定任意一方一手的时间为 30 秒，一局的时间为 12 分钟。		☒	☐
	默认登录时的用户 1 为黑棋；黑棋先下，当下第一子以后，白棋的倒计时开始计时。白棋再走一子后，白棋的计时器停止，黑棋的计时器开始计时。黑棋再走一子，白棋的“本轮剩余时间”继续倒计时，而“本轮剩余时间”则重置到 30 秒重新开始计时。		☒	☐

	<p>当我方首先连出 5 子，或对方时间走光后，程序会判定我方获胜，在用户信息中会有以下改动：积分+5，总局数+1，获胜局数+1。</p> <p>同时，对方的信息会有以下改动：积分-5，最低到 0；总局数+1。</p> <p>界面中的胜率=获胜局数/总局数*100%。</p>		<input checked="" type="checkbox"/>	<input type="checkbox"/>
--	--	--	-------------------------------------	--------------------------

六、任务分工

6.1 代码部分

MainWindow: 夏涵 宋斯亮;
myTimer: 赫炎 夏涵;
AI: 宋斯亮;
User: 赫炎;
Login: 夏涵;
enroll: 夏涵;
chessboard: 夏涵;
chessman: 夏涵;
GLOBALA: 夏涵;
sky_ladder: 赫炎;
welcome: 夏涵;
界面设计: 宋斯亮 夏涵;
整合: 夏涵;
系统整体设计: 宋斯亮 夏涵;
代码注释: 全体成员
软件测试: 全体成员

6.2 文档

引言: 赫炎;
需求分析: 赫炎;

状态图：宋斯亮；

类图及模块功能：宋斯亮 夏涵；

测试：赫炎；

文档整合：夏涵

七、感想

夏涵：

本来没准备加上感想这一个环节的，但是做了这么久还是想说些什么。我本身是生物专业，由于对计算机有着浓厚的兴趣，大三下（这学期）开始学习计算机相关课程，准备将来往计算机方向发展。所以说本身底子不是很好，学期刚开始数据库系统，软件工程，面向对象，数据结构，一个个任务像山一样压下来。尤其是面向对象的第一个 `string` 大作业，对于一个连 `C` 都写不好的小白，真的是鸭梨山大。

刷书，码代码，刷书……这样地重复着，庆幸的是我坚持下来了，当然也少不了小伙伴的帮助。从真正独立写完了面向对象第二个稍微简单的 `shape` 程序之后，码代码的乐趣一直激励着我。春学期的数据库系统大作业，一个简易图书管理系统我也带着小伙伴们顺利完成。夏学期，刚不久和小伙伴们完成了数据库系统的 `miniSQL` 大程序，今天又算是真正完成了面向对象的大程序。自己在这两个大作业中扮演着领导者的角色，使我对整个开发过程有了更深入的理解，收获颇丰。这两个大作业总代码量应该超过 1600 行了吧嘻嘻~超级满足的一个学期。

尤其是五子棋程序，一开始上软件工程课的时候觉得没啥用，都是文科的东西感觉。但是到自己要领导开发的时候一下子慌了神，从软件工程这门课里找到了不少理论依据。由于一开始没有一个完整的思路，在要求组员实现的时候总是要提出一些修改传入参数啊，返回值啊，或者增加功能的一些事。整个过程中也正是通过这种试错学习，才让自己成长。

最后，首先要感谢组员们的辛苦工作~再次，要感谢这个学期帮我的小伙伴们。最后要感谢助教群里耐心回答问题，以及陈老师的人生之道编程方面的教导，谢谢！【鞠躬】