开封大学

毕业设计

题 目 基于C++的五子棋对战系统的设计与实现

学生姓名 赵帅印

专业班级 21网络2

学 号 2021101035

院 （系） 信息工程学院

指导教师(职称) 付红杰（讲师）

完成时间 2023 年10月21日

目 录

[论 文 摘 要 2](#_Toc2458)

[一、绪论 3](#_Toc1848)

[（一）选题的目的与意义 3](#_Toc1490)

[二、相关技术介绍 6](#_Toc27605)

[（一）Qt简介 6](#_Toc31683)

[（二）C/S架构的优点 6](#_Toc4037)

[（三）MySQL 数据库 7](#_Toc11084)

[三、可行性分析及需求分析 8](#_Toc10561)

[（一）可行性分析 8](#_Toc30767)

[1. 经济可行性 8](#_Toc16818)

[2. 技术可行性 8](#_Toc20536)

[3. 运行可行性 8](#_Toc31141)

[（二） 需求分析 8](#_Toc31693)

[1. 功能性需求分析 9](#_Toc16825)

[2. 非功能性需求分析 10](#_Toc29621)

[（三）运行环境 10](#_Toc22451)

[四、系统总体设计 10](#_Toc28245)

[（一）系统的总体功能结构设计 10](#_Toc4429)

[（二）数据库设计 11](#_Toc10175)

[1.概念结构设计E-R图 11](#_Toc2697)

[2.逻辑结构设计 12](#_Toc17216)

[3.物理结构设计（数据库表结构） 12](#_Toc15241)

[（三）系统工作流程 13](#_Toc23122)

[五、系统总体实现 14](#_Toc23101)

[（一）服务端的功能实现 14](#_Toc11287)

[（二）客户端的功能实现 20](#_Toc20070)

[六、系统测试 33](#_Toc14802)

[（一）软件测试的定义 33](#_Toc29369)

[（三）测试用例 34](#_Toc16369)

[总结与展望 35](#_Toc29744)

[致谢 36](#_Toc14506)

[参考文献 37](#_Toc26825)

# 论 文 摘 要

在现在如此发达并且得到广泛应用的现代网络技术下，集趣味性，娱乐性，益智性，并且包含网络功能的休闲类游戏以及成为了人们选择网络休闲游戏的要求。

系统采用当今广为流行的五子棋游戏为模版，利用C++的第三方GUI设计工具Qt为程序设计界面，并结合软件工程的思想开发一款基于网络的五子棋游戏对弈软件。此系统共分为登录模块，本地游戏，网络游戏等三个模块，其中登录模式连接数据库，通过注册功能向数据库输入账号，并登录功能来成功登录。本地游戏实现了两个人在一台计算机上互相攻略的功能，而本软件网络游戏采用P2P的模式，利用一个服务端来辅助各个客户端之间的查找和链接，服务端用于记录当前各个客户端的游戏状态，并将实时更新的各个客户端的状态发送到其他客户端，方便客户端加入其他客户端建立的游戏主机，并建立P2P链接，进行游戏。客户端是玩家用于建立游戏主机或者加入游戏进行五子棋对弈的主程序，具有当今五子棋游戏常见的聊天功能、悔棋功能、认输功能等。

玩家只需运行客户端，连接到指定的服务端，就可以加入其他的游戏主机进行游戏，或者建立游戏主机，等待其他客户端的加入进行游戏对弈。

关键词：五子棋 C/S架构 Qt MySQL

**基于C++的五子棋游戏的设计与实现**

赵帅印

（开封大学 信息工程学院 计算机网络技术）

# 一、绪论

## （一）选题的目的与意义

五子棋在我国的历史可谓是源远流长，喜爱它的玩家也是大有人在。但目前的五子棋软件一般都是单机版的，游戏无法在异地的玩家之间进行，因此为了方便异地的玩家在一起进行五子棋游戏，设计出一款基于网络版的五子棋游戏，玩家可以通过网络挑选自己喜爱的对手并同其进行比赛，比赛过程中可以进行聊天，达到相互交流经验的作用，同时能让身处各地的玩家连接到服务器进行五子棋对弈。

五子棋[1]是起源于中国古代的传统黑白棋种之一。现代五子棋日文称之为“连珠”，英文称之为“Renju”、“Gobang”或“FIR”（Five in a Row 的缩写），亦有“ 连五子”、“五子连”、“串珠”、“五目”、“五目碰”、“五格”等多种称谓。

五子棋不仅能增强思维能力，提高智力，而且富含哲理，有助于修身养性。五子棋既有现代休闲的明显特征“短、平、快”，又有古典哲学的高深学问“阴阳易理”；它既有简单易学的特性，为广大人民群众所喜闻乐见，又有深奥的技巧和高水平的国际性比赛；它的棋文化源渊流长，具有东方的神秘和西方的直观；既有“场”的概念，亦有“点”的连接。它是中西文化的交流点，是古今哲理的结晶。

五子棋起源于古代中国，发展于日本，风靡于欧洲。对于它与围棋的关系有两种说法，一说早于围棋，早在“尧造围棋”之前，民间就已有五子棋游戏；一说源于围棋，是围棋发展的一个分支。在中国的文化里，倍受人们的青睐。古代的五子棋的棋具与围棋相同。五子棋大约随围棋一起在我国南北朝时先后传入朝鲜、日本等地。从此，五子棋经过了不断的改良，例如棋盘由原来的纵横各十七道改为现行的纵横各十五道的五子棋专用棋盘等等。二十世纪初，五子棋传入欧洲并迅速风靡了全欧洲。通过一系列的变化，使五子棋这一简单的游戏系统化、规范化，最终成为今天的职业连珠五子棋，同时也成为一种国际比赛棋。

在网络发达的今天，对于二人对弈的五子棋，开发一款带有网络功能的五子棋游戏，能够更加方便身在异地的玩家进行游戏。

1. 研究现状分析

目前常见的单机类五子棋游戏可分为“人机模式”与“双人模式”。“人机模式”指人与按照事先编写好的五子棋落子算法程序来进行游戏的计算机进行对弈，“双人模式”指两个人通过使用同一台计算机，交替落子来进行游戏。这两种方式，都需要玩家双方在同一台计算机面前进行完成，因此局限性很大，玩家不能再异地的不同的计算机上来进行游戏。

随着现代计算机网络的迅速发展，网络上的计算机之间的通信已经变得十分便捷。这也为身在异地间的用户利用网络来进行游戏提供了途径。

网络版的五子棋软件利用网络连接协议TCP/IP，在不同地方的玩家之间建立一个TCP连接，并用这个连接来交换数据，进行通信。因此，网络版的五子棋软件在游戏之前必须确定对网络参数设置的正确性，设置正确之后，才能进行玩家双方的TCP成功连接。连接成功后，双方轮流下棋，同时将每一步落子的信息通过网络连接传送给对方，使双方棋盘上的状态保持一致。

与单机版的五子棋软件相比，网络版的五子棋软件也提供了所有单机版的五子棋软件所具有的功能，例如：倒计时器、“认输”、“和棋”和“悔棋”等。但对于网络版的五子棋来说，这些功能的同步都是通过网络信息交互来实现的，因此，网络版的五子棋游戏实现过程较单机版要复杂一些。

在胜负判断方面，单机版的五子棋软件与网络版的五子棋软件可以采用完全相同的算法（如果不考虑先手禁手等）。一方玩家下完一步棋后，系统会自动判断当前落子之后，玩家是否胜利，如果没有胜利，再将落子的信息传送给对方，如果判断结果为胜利，则直接发送胜利消息给对方。每次落子之后，都要进行这样一次胜负判断定的过程。另外，由于网络版的五子棋的游戏双方可能不在同一个地点，因此，提供聊天功能很有必要。

目前常见的网络五子棋的游戏形式又可分为两种，一种是以QQ五子棋（如图1）为代表的桌面应用程序的游戏形式，一种是网页flash（如图2）的游戏形式

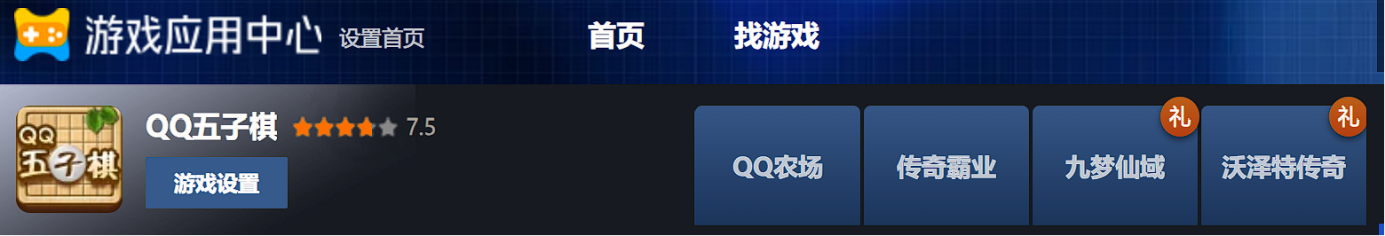


图1 QQ五子棋游戏

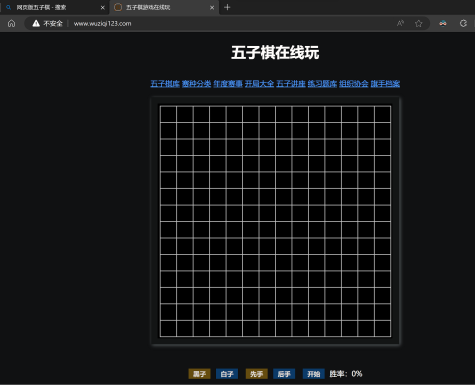


图2 网页版五子棋

系统结构方面，QQ五子棋游戏采用了C/S结构，而网页flash的五子棋采用的是B/S结构。

功能方面，二者均提供了常见的悔棋、计分、计时、聊天、认输等功能。均未设计五子棋中“先手禁手”等规则。不同之处在于，常见的网页五子棋对于一方断网的情况不能及时发现，发现后也只是做出提醒另外一方的操作。而QQ五子棋能够对一方断网的情况及时发现，并且采用电脑托管或者直接结束比赛并计算相应分数的操作。

游戏流畅度方面，网页flash的游戏形式受限于浏览器解析网页代码的速度限制，游戏载入时间明显过长。而QQ五子棋只需要基本的windows操作系统即可，运行环境要求较低，运行流畅。

# 二、相关技术介绍

## （一）Qt简介

QT是Trolltech公司的标志性产品,是一个跨平台的C++图形用户界面(GUI)工具包,QT应用程序接口与工具兼容于所有支持平台,让开发员们掌握一个应用程序接口,便可执行与平台非相关的应用开发与配置(QT/windows对不同平台的API进行了进行了专门的封装(文件处理,网络)。Qt是一个相对完整的应用程序开发软件,包括一个类库和一些跨平台、国际化的工具。它是挪威Troll Tech公司的标志产品。Qt工具箱有三种不同的版本,一是专业版,另一种是免费版,还有一种是嵌入式免费版。专业版用于创造成功的商业贸易方面的跨平台软件;而自由版已经成为用C++GUI工具包在 Linux上进行由软件开发的主流,它是Linux 上流行的KDE桌面环境的基础。由于Qt是一个跨平台的GUI工具箱,它确保了软件非常方便的应用于绝大多数操作系统。Qt 的良好封装机制使得 Qt 的模块化程度非常高,可重用性较好,对于用户开发来说是非常方便的。Qtcallback,这使得各个元件 之间的协同工作变得十分简单。

Qt包括多达 250个以上的 C++类,还提供基于模板的collections, serialization,file, I/Odevice, directory management, date/time 类。甚至还包括正则表达式的处理功能。支持 2D/3D 图形渲染,支持 OpenGL,大量的开发文档,XML支持,Webkit 引擎的集成,可以实现本地界面与Web内容的无缝集成。但是真正使得Qt在自由软件界的众多Widgets(如 Lesstif, Gtk,EZWGL, Xforms, fltk 等等)中脱颖而出的还是基于 Qt的重量级软件 KDE。

## （二）C/S架构的优点

C/S即Client/Server（客户机/服务器）结构，C/S结构在技能上非常成熟，它的重要特征就是交互性强、拥有安全的存取形式、网络通信数量低、响应速度快、利于处置大量数据。可是这个结构的程序就是针对性开发，变更不够灵活，维护与管理的难度较大。常常只局限在小型局域网，不利于扩展。而且，因为这个结构的每台客户机全部须要安装相对应的客户端程序，分布功能弱并且兼容性差，不可以完成迅速部署安装与配置，因为这样缺少通用性，拥有比较大的局限性。请求拥有肯定专业水准的技能人员去结束。

C/S架构：

（1）优点：

●可以足够表现客户端PC的处置才能，很多工作能够在客户端处置以后再提交给服务器，于是CS客户端响应速度快。

●操作界面漂亮、形式多样，能够足够满足客户自己的个性化要求。

●C/S结构的管理信息系统拥有比较强的事务处置才能，可以完成复杂的业务过程。

●安全性能能够非常容易确保，C/S通常面向相比固定的用户群，程序越发注重过程，它能够对权限实行多层次校验，提供了更安全的存取形式，对信息安全的控制才能非常强。通常高度机密的信息系统选用C/S结构适宜。

## （三）MySQL 数据库

MySQL是一种开放源代码的关系型数据库管理系统（RDBMS），MySQL数据库系统使用最常用的数据库管理语言--结构化查询语言（SQL）进行数据库管理。由于MySQL是开放源代码的，因此任何人都可以在General Public License的许可下下载并根据个性化的需要对其进行修改。MySQL因为其速度、可靠性和适应性而备受关注。大多数人都认为在不需要事务化处理的情况下，MySQL是管理内容最好的选择。MySQL[关系型数据库](http://www.so.com/s?q=%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)于1998年1月发行第一个版本。它使用系统[核心](http://www.so.com/s?q=%E6%A0%B8%E5%BF%83&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)提供的[多线程](http://www.so.com/s?q=%E5%A4%9A%E7%BA%BF%E7%A8%8B&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)[机制](http://www.so.com/s?q=%E6%9C%BA%E5%88%B6&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)提供完全的多线程[运行模式](http://www.so.com/s?q=%E8%BF%90%E8%A1%8C%E6%A8%A1%E5%BC%8F&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)，提供了面向C、C++、Eiffel、Java、Perl、PHP、Python等编程语言的编程接口（APIs），支持多种字段类型并且提供了完整的[操作符](http://www.so.com/s?q=%E6%93%8D%E4%BD%9C%E7%AC%A6&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)支持查询中的SELECT和WHERE操作，速度快，可以用于小型[项目](http://www.so.com/s?q=%E9%A1%B9%E7%9B%AE&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)，**所以中小型的公司都喜欢使用它作为网站数据库。**

MySQL特点：

**1.性能卓越服务稳定，很少出现异常。**

**2.开放源代码且无版权制约，自主性强，使用成本低。**

**3.历史悠久，社区及用户非常活跃，遇到问题可以很快获取到帮助。**

4.软件体积小，安装使用简单，并且用于维护，安装及维护成本较低。

5.支持多种操作系统，提供多种API接口，支持多种开发语言。

# 三、可行性分析及需求分析

只有满足了客户对本系统功能的要求，我们才能够分析和寻求解决方案，最终实现这些具体的系统功能，本节将分别从可行性分析及用户需求两个方面进行具体的研究。

## （一）可行性分析

可行性分析是通过对系统的初步的调研和分析，提出可行性方案并进行论证的过程。本节从经济可行性、技术可行性、运行可行性进行分析。

### 经济可行性

非常简洁的界面设计,可以让玩游戏的人很明白的操作。而且游戏系统的开发成本非常低,在现在所有的计算机上可以很轻松简单的完成实现整个环境的需求。

### 技术可行性

运用云服务器平台或者Qt客户端,以Qt为平台开发界面,开发功能完整的五子棋

游戏。

### 运行可行性

新的系统运行后对现行旧的系统带来包括(工作环境、管理方式、组织机构等)的后果以及影响来进行评判和估计。同时更需要考虑到的是:对现有的管理人员进行培训,补充、分析在给出的时间里是不是能完成预定开发系统的任务等系统的开发,是典型的Mis开发,主要是对数据的处理,包括数据的收集,数据的变换,及数据的各种报表形式的输出.只需要鼠标的点击就可以完成真的玩游戏。

## 需求分析

五子棋游戏是一款很符合现代人的休闲特点的益智类游戏，其是我国古代围棋的的衍生物，本系统主要分为两部分：联机对战和人机对战。

联机对战模式实现的功能：选择联机对战，进入联机游戏界面，连接服务器，进入或创建房间，点击准备，选择先后手，开始对战，判断输赢，重新开始游戏。

人机对战模式实现的功能：选择人机对战，设置先手，实现玩家与电脑轮流落子，棋盘动态信息显示，判断输赢，重新开始游戏。

### 功能性需求分析

系统可分为服务端和客户端部分，客户端又分为登录模块，网络游戏模块，本机游戏模块。而实现这些部分，模块的功能如下所示

1、服务器开始、退出

服务端程序能够开启、关闭五子棋游戏的服务。关闭服务时，能够给当前已连接的客户端发送关闭服务消息。

2、登录，注册

启动客户端的代码时，客户端会首先打开登陆功能，玩家需要输入在数据库中的账号和密码才能进入主页面。

3、玩家开始、退出

玩家进入游戏主机之后，需要点击准备才能开始游戏，若玩家在游戏时退出游戏，需要向对手发送退出游戏的信息。

4、聊天功能

玩家在对弈过程中可以发送信息进行交流。

5、悔棋

玩家在对弈过程中，能够进行悔棋请求，如果对方答应悔棋请求，则系统撤销双方玩家最近一次的落子；若对方不答应悔棋请求，则游戏继续。

6、认输

玩家在对弈过程中，可以直接选择认输，不需要经过对方同意。

7、计分

程序能保存当前两个玩家从开始第一句对弈，到其中一方离开游戏这期间的战绩情况。

8、断网处理

当其中一方玩家意外断网时，程序能通过超时相应判断出对方丢失了网络连接并作出处理。

### 非功能性需求分析

1、界面友好、美观

2、界面简洁

3、操作界面方便、易懂

## （三）运行环境

1、装有windows或者linux的操作系统的计算机两台

2、计算机的CPU不低于P4，内存不小于64M

3、10/10Mbps自适应网卡每台计算机各一块

4、每台计算机都应配置有TCP/IP协议

# 四、系统总体设计

## （一）系统的总体功能结构设计

该系统有两大模块，分为服务端模块和客户端模块。服务端模块的具体功能如下：

处理客户端刷新战局的请求功能、处理客户端创建房间的请求、处理客户端退出房间的请求、处理客户端加入房间的请求和处理客户端更新对手准备状态的请求。客户端模块的具体功能如下：登陆、注册模块：该模块与数据库相连，当玩家打开客户端时，会第一时间打开登录模块，只有输入数据库中的账号与密码才能进入到主页面。而点击注册则可以向数据库中输入账号和密码。网络游戏模块：该模块是系统的主要功能模块，用来处理玩家与玩家之间进行对弈，通过连接服务器将玩家与玩家连接到一起。本地游戏模块：该模块用来处理玩家与人机之间的对弈。如图6所示，为本系统的总体结构图：

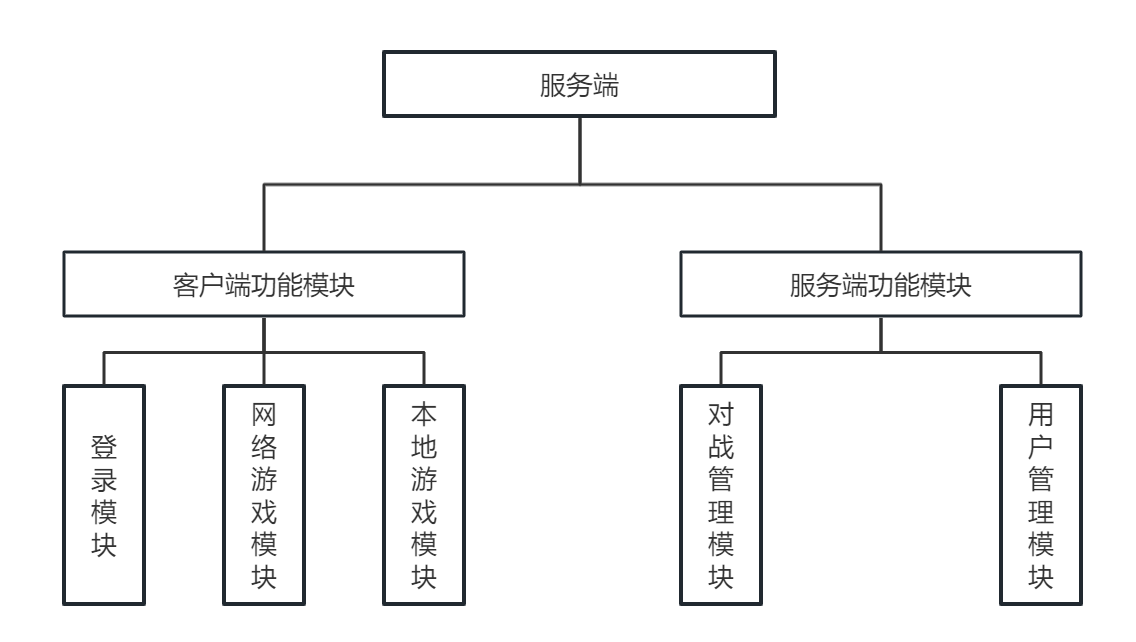


图6系统总体结构图

## （二）数据库设计

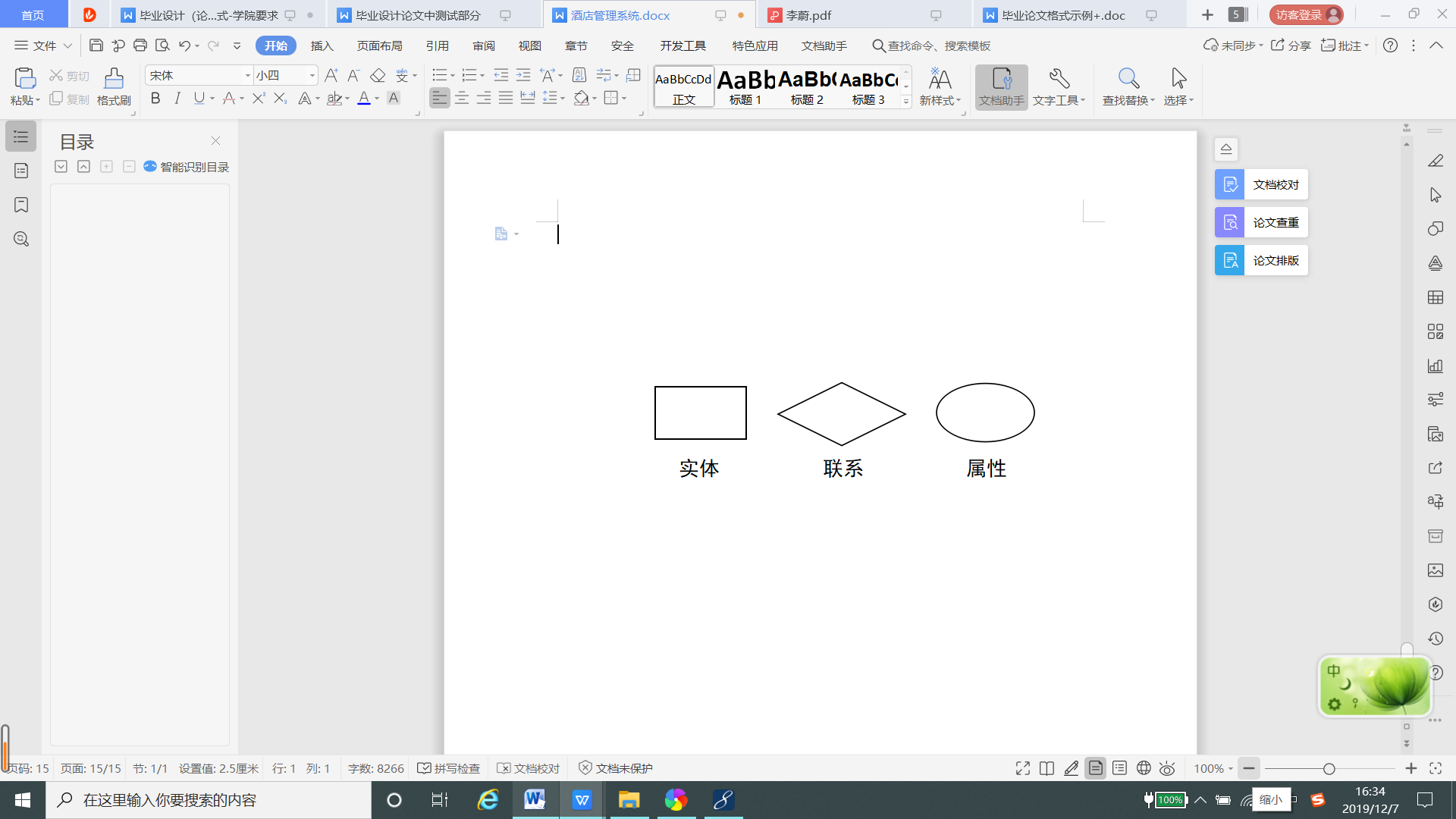
### 1.概念结构设计E-R图

概念模型中最常用的方法为实体—关系（E-R）方法，主要概念有实体、联系和属性。该方法直接从现实世界中抽象出实体和实体间的联系。E-R 图的三个要素：

实体：使用矩形表示，框内标注实体名称。

属性：使用椭圆表示，并用无向边将其与相应的实体连接起来。

实体之间的联系：使用菱形框表示，框内注明联系名称，并用无向边分别与相关的实体连接起来，同时在无向边旁标上联系的类型（1∶1、1∶n 或 m∶n）。如果一个联系具有属性，则这些属性也要用无向边与该联系连接起来。实体、联系、属性的标识符如下：



基于本应用需求建立对应 E-R 模型如下。

1. 玩家信息实体属性图如图7所示。

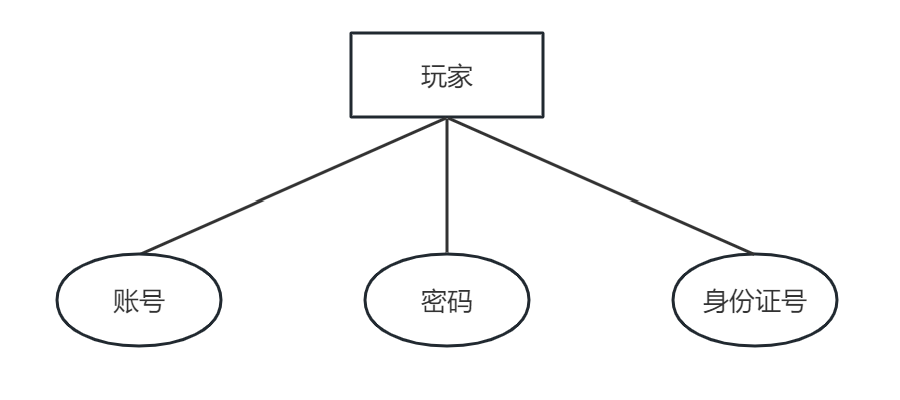


图7玩家信息实体属性图

### 2.逻辑结构设计

数据库的逻辑结构设计是将在数据库设计的E-R模型进行转换成关系型数据库系统支持的数据模型，本课题酒店管理系统的实体及之间的关系可以确定本系统所需的数据库逻辑结构。

1. 用户表：（账号、密码，姓名）

### 3.物理结构设计（数据库表结构）

根据逻辑结构设计，按照表转换，可以得到系统一共有一张表，是：用户表（user\_table）下面就是每张表的具体结构：

（1）用户表：主要包括：账号、姓名、密码等字段 ，管理员信息表的具体结构如表1所示。

表 1 用户表（new\_tables）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 是否为主键 | 说明 |
| username | varchar | 12 | 是 | 账号 |
| passward | varchar | 18 | 否 | 密码 |
| id | varchar | 45 | 否 | 身份证号 |

## （三）系统工作流程

本系统的能够实现一个在网络上供玩家进行对战的网络版的五子棋游戏，玩家只要登陆到服务器上，然后选择其它已创建好的游戏房间，进行五子棋对战游戏。或者自己建立一个游戏房间，等待对方的加入。

在C/S模式游戏中，服务端一般提供所有用户的全局信息，并能提供客户之间的信息转发，客户之间的通讯必须通过服务端进行。因为在多个客户能够连接到同一台服务端上，所以服务端必须用线程负责每个用户的通讯和消息处理。

但是考虑到如果采用这种思路，当客户端达到一定的数量之后，会增加服务端的负荷，而且当两个玩家开始游戏之后，服务端只需知道玩家双方是否退出游戏以及双方的游戏结果即可，而玩家之间的对弈信息、聊天等信息大可不必经过服务端。设计思想的结构流程图如图7：

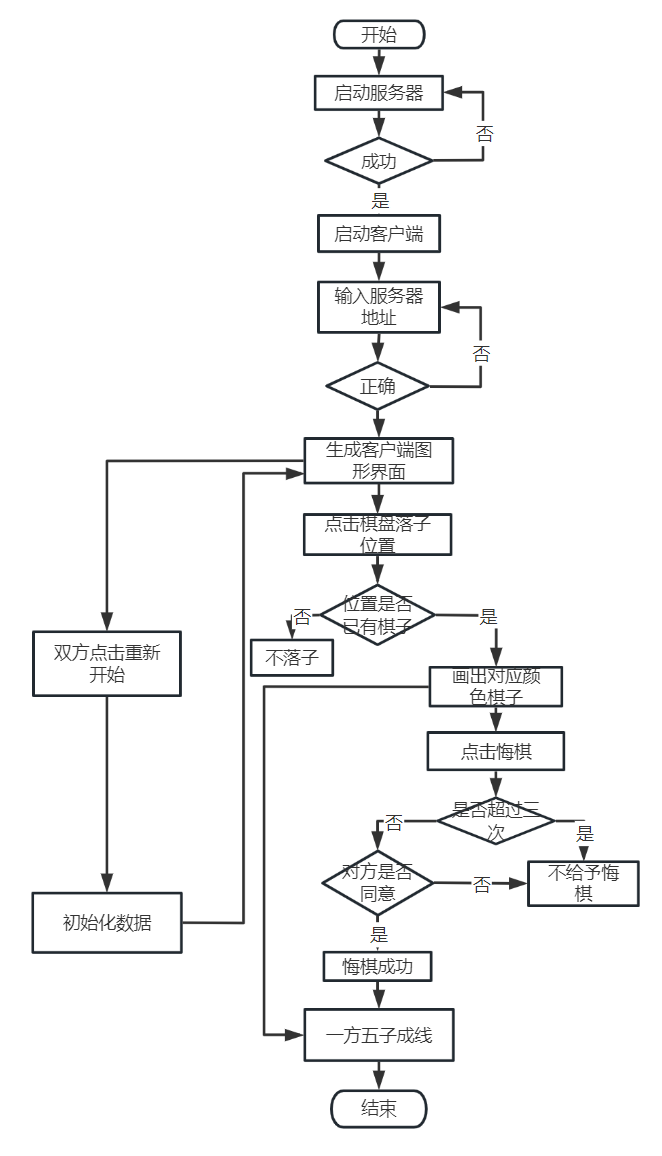


图7系统流程图

系统使用Socket技术与Qt多线程机制结合，使用C/S模式，在进行客户端与服务端以及各个客户端之间进行信息的交互。

例如某个客户端在连接服务器时，详细流程如图8：

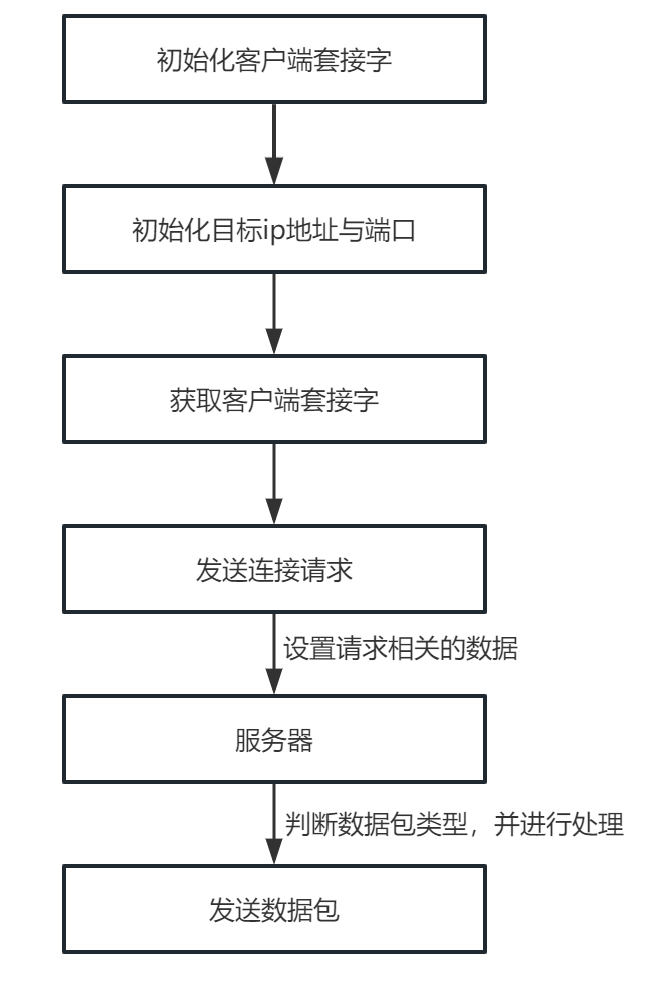


图8客户端连接服务端流程图

# 五、系统总体实现

## （一）服务端的功能实现

系统使用Socket技术与Qt多线程机制结合，使用C/S模式，在进行客户与服务端以及各个客户端之间进行信息的交互。

其中，对客户端进行的各个操作请求产生的数据包发送的对象如表2：

表 2 用户程序协议表

|  |  |
| --- | --- |
| 客户端操作类型 | 数据包发送对象 |
| 用户登录请求 | 服务器 |
| 用户建立主机请求 | 服务器 |
| 用户加入游戏请求 | 服务器 |
| 网络丢失确认 | 服务器 |
| 对方网络丢失信息通知 | 服务器 |
| 发送认输通知 | 客户端 |
| 发送落子信息 | 客户端 |
| 发送聊天内容 | 客户端 |
| 发送悔棋请求 | 客户端 |

其中，“数据类型”字段如表3：

表 3 数据类型表

|  |  |
| --- | --- |
| 消息类型 | 说明 |
| COMM\_SERVER\_CONN\_FAILED | 连接服务器失败 |
| COMM\_SERVER\_CONN\_SUCCESSFUL | 连接服务器成功 |
| COMM\_SERVER\_CLOSE | 服务器关闭 |
| COMM\_SERVER\_GAMEINFO | 服务器发送大厅信息 |
| COMM\_CLIENT\_QUITGAME | 客户端退出游戏 |
| COMM\_CLIENT\_DISCONN | 客户端请求断开服务器 |
| COMM\_CLIENT\_CONN | 客户端请求连接服务器 |
| COMM\_CLIENT\_CREATE | 客户端请求创建主机 |
| COMM\_CLIENT\_JOIN | 客户端请求加入游戏 |
| COMM\_CLIENT\_GAMESTART | 双方准备完毕，游戏开始 |
| COMM\_CLIENT\_GAMEOVER | 某方玩家胜利，游戏结束 |
| COMM\_CLIENT\_GAMEOP | 玩家游戏操作：落子 |
| COMM\_CLIENT\_CHAT | 玩家游戏操作：发送聊天信息 |
| COMM\_CLIENT\_UNDO | 玩家游戏操作：悔棋 |
| COMM\_CLIENT\_UNDO\_YES | 玩家游戏操作：悔棋回复yes |
| COMM\_CLIENT\_UNDO\_NO | 玩家游戏操作：悔棋回复no |
| COMM\_CLIENT\_LOSE | 玩家游戏操作：认输 |
| COMM\_CLIENT\_LOSTCONN | 玩家发送消息：对手超时，判定为掉线 |

请求方需要发出请求时，便把消息类型设置为如上对应的消息类型（例如：服务器连接请求），并设置请求相关的数据（例如：需要连接的服务器的IP以及端口），最后发送此数据包。

数据字段根据第一个字段的不同设置不同的数据，有可能为包含五子棋落子位置的信息；有可能为空数据（例如：连接服务器成功则数据字段为空，连接服务器失败则数据字段为连接服务器失败的原因）。

接收方在接受到此数据包之后，首先判断出数据包的类型什么，然后利用接受到的数据（例如：服务器的IP以及端口）以及数据包类型（例如：服务器连接请求）来处理数据（例如：是否接受此客户端的连接），处理完毕之后，设置响应数据包类型为具体的响应类型（例如：连接成功），并且设置响应相关的数据（例如：连接成功则没有数据，连接失败则数据为连接失败的原因），最后发送此数据包。

服务端只是作为后台支撑，并无管理界面，服务端大致分为五种功能，分别为：处理客户端刷新战局的请求、处理客户端创建房间的请求、处理客户端退出房间的请求、处理客户端加入房间的请求和处理客户端更新对手准备状态的请求。

1.处理客户端刷新战局的请求功能主要是通过memset函数将结构体清零，sprintf格式化输出函数，write函数向文件中写入内容实现的，具体代码如下所示：

void R\_signal(int client\_fd)

{

int sum=0;

char msg\_[1024];

for(auto x:rooms)

{

if(x.client\_fd==-1)

sum++;

}

memset(msg\_,0,sizeof(msg\_));

sprintf(msg\_,"/S%ld/S%d",client\_fds.size(),sum);

//printf("[%d]%d\n",\_\_LINE\_\_,sum);

write(client\_fd,msg\_,strlen(msg\_));

for(auto x:rooms)

{

if(x.client\_fd==-1)

{

memset(msg\_,0,sizeof(msg\_));

sprintf(msg\_,"/N%s",x.room\_name.c\_str());

//printf("[%d]%s\n",\_\_LINE\_\_,msg\_);

write(client\_fd,msg\_,strlen(msg\_));

memset(msg\_,0,sizeof(msg\_));

sprintf(msg\_,"/I%s",inet\_ntoa(client\_addrs[x.master\_fd].sin\_addr));

//printf("[%d]%s\n",\_\_LINE\_\_,msg\_);

write(client\_fd,msg\_,strlen(msg\_));

memset(msg\_,0,sizeof(msg\_));

sprintf(msg\_,"/F%d",x.master\_fd);

//printf("[%d]%s\n",\_\_LINE\_\_,msg\_);

write(client\_fd,msg\_,strlen(msg\_));

}

}

}

2.处理客户端创建房间的请求功能主要是通过定义room\_information room(buf,fd),并通过push\_back函数将创建的房间名展示出来，hash\_client[fd].master=true来表明房间的主人，具体代码如下所示：

void C\_signal(char\* msg,int fd)

{

char buf[1024];

memset(buf,0,sizeof(buf));

for(int i=2;msg[i]!='\0';i++)

buf[i-2]=msg[i];

//printf("[%d]%s\n",\_\_LINE\_\_,buf);

room\_information room(buf,fd);

rooms.push\_back(room);

hash\_client[fd].room\_num=rooms.size()-1;

hash\_client[fd].master=true;

}

3.处理客户端退出房间的请求功能主要是通过分为三种情况实现的，分别为1主人的套接字为假，2为真时，但房间有对手时，3为真时，房间没有对手时。具体代码如下所示：

hash\_client[client\_fd].opponent\_fd=0;

hash\_client[client\_fd].master=true;

rooms[room\_num].master\_fd=client\_fd;

rooms[room\_num].client\_fd=-1;

}

else

{

int r=hash\_client[fd].room\_num;

rooms.erase(rooms.begin()+r,rooms.begin()+r+1);

for(int i=r;i<rooms.size();i++)

{

if(rooms[i].client\_fd>0)

hash\_client[rooms[i].client\_fd].room\_num=i;

hash\_client[rooms[i].master\_fd].room\_num=i;

}

}

hash\_client[fd]=client\_information();

}

}

4.处理客户端加入房间的请求功能主要是通过对比房间号是否一致，及有无对手等功能实现的，具体代码如下所示：

void J\_signal(int fd,char\* msg)

{

int sum=0;

for(int i=1;msg[i]!='\0';i++)

sum=sum\*10+msg[i]-'0';

if(sum<=0||hash\_client[sum].room\_num<0||hash\_client[sum].opponent\_fd>0)

{

write(fd,"/Zerror",strlen("/Zerror"));

return;

}

hash\_client[sum].opponent\_fd=fd;

rooms[hash\_client[sum].room\_num].client\_fd=fd;

hash\_client[fd].opponent\_fd=sum;

hash\_client[fd].room\_num=hash\_client[sum].room\_num;

write(fd,"/Zsuccess",strlen("/Zsuccess"));

}

1. 处理客户端更新对手准备状态的请求主要是通过先用memset将结构体清零，再用if语句判断房间中有无对手实现的，具体代码如下所示：

void U\_signal(int fd)

{

char msg[1024];

memset(msg,0,sizeof(msg));

if(hash\_client[fd].opponent\_fd>0)

sprintf(msg,"/Z1/Z%d/Z%s/Z%d",hash\_client[hash\_client[fd].opponent\_fd].prepare,

inet\_ntoa(client\_addrs[hash\_client[fd].opponent\_fd].sin\_addr),hash\_client[fd].opponent\_fd);

else

sprintf(msg,"/Z0/Z /Z /Z ");

write(fd,msg,strlen(msg));

}

服务端与客户端之间的信息交互采用C/S模式，因此当有客户端更新并向服务端发送信息（例如建立主机，加入游戏，退出游戏等）之后，服务端应及时将更新之后的信息广播出去，使得其他客户端及时更新大厅信息。

## （二）客户端的功能实现

客户端具有一个登录界面，登录界面与MySQL数据库相连接，只有输入与相应数据库中表的账号和密码，才能登陆成功，反之，失败。而点击注册，则可往数据库中的表输入信息，该信息会被记录到表中。登录界面如图9所示：

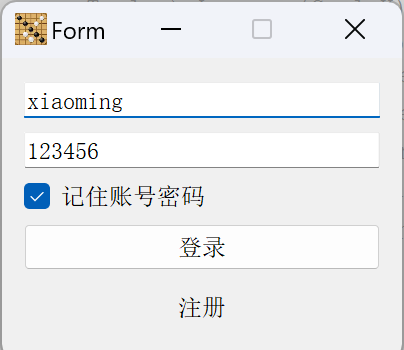


图9登录界面

登录界面的核心代码如下：

//用户登录：查询数据

connect(ui->Loginbtn, &QPushButton::clicked, this, [=]()

{

//接收用户输入：

QString account = ui->username->text(); //用户账号

QString passwd = ui->password->text();

//查询数据：

QString qs = QString("select \* from new\_table where username = '%1' and passward = '%2'").arg(account).arg(passwd);

if (!m\_queryInfo->exec(qs)) //如果没有查到记录

{

qDebug() << m\_queryInfo->lastError().text(); //输出错误信息

return ;

}

//获取查询的数据：

if (m\_queryInfo->next()) //获取到数据

{

//QMessageBox::information(this, "登录" ,"登录成功");

//跳转窗口：

this->hide(); //隐藏主窗口

Menu \*tw = new Menu(); //创建新窗口

tw->show(); //显示子窗口

//在配置文件中记录用户账号密码：

if (ui->checkBox->isChecked()) //选中

{

//创建配置文件:

QSettings setting("config.ini", QSettings::IniFormat); //配置文件在工程目录下

//把用户账号密码写到配置文件中：

setting.beginGroup("section"); //节开始

setting.setValue("username", account);

setting.setValue("passward",passwd);

setting.endGroup(); //结束

}

}

而登录界面需要实现，连接数据库必不可少，则连接数据库的核心代码如下所示：

QSqlDatabase m\_mysqldb;

QSqlQuery \*m\_queryInfo;

m\_mysqldb = QSqlDatabase::addDatabase("QMYSQL","chess");

m\_mysqldb.setDatabaseName("chess");

m\_mysqldb.setHostName("182.43.99.171");

m\_mysqldb.setPassword("Zhao123456!");

m\_mysqldb.setPort(9004);

m\_mysqldb.setUserName("zhao");

qDebug()<<"conect name = "<<m\_mysqldb.connectionName();

if(!m\_mysqldb.isOpen())

{

if(!m\_mysqldb.open())

{

qDebug()<< "db " <<m\_mysqldb.userName()<<" last error"<<m\_mysqldb.lastError().text();

qWarning("m\_mysqldb open failed!");

return ;

}

else

{

qDebug().noquote().nospace()<<"m\_mysqldb open sucess ";

}

}

m\_mysqldb.exec("SET NAMES utf8;");

m\_queryInfo = new QSqlQuery(m\_mysqldb);

if(!m\_queryInfo -> exec("SELECT username,passward FROM chess.new\_table;"))

{

return ;

}

while(m\_queryInfo -> next())

{

qDebug().noquote().nospace()<< m\_queryInfo->value(0).toString() <<"\t"

<< m\_queryInfo->value(1).toUInt();

}

登录成功后，则会进入主页面，主页面中分为“本地游戏”“网络游戏”“退出游戏”三个功能。主页面如图10所示：



图10主页面界面

进入主页面之后，点击“网络游戏”，进入“网络游戏”界面，用户这时可以选择建立游戏主机等待别人加入，或者是加入别人的主机进行对弈。网络游戏界面如图所示：

图11网络游戏界面

因为想要在网络游戏界面与玩家对战必须连接服务端，则客户端连接服务端的核心代码如下所示：

//初始化客户端套接字与目标ip地址与端口

client\_net::client\_net()

{

WSAStartup(MAKEWORD(2, 2), &wsadata);

client\_fd = socket(PF\_INET, SOCK\_STREAM, 0); //[BUG]建立tcp连接(重要\*)

client\_addr.sin\_family = AF\_INET;

client\_addr.sin\_addr.s\_addr = inet\_addr("182.43.99.171"); //你的服务器的公网IP

client\_addr.sin\_port = htons(7030); //目的端口号

connected = false;

connect\_thread\_running = false;

received = false;

}

client\_net::~client\_net()

{

qDebug() << "client\_net 析构..." << endl;

closesocket(client\_fd);

WSACleanup();

}

}

//获取客户端套接字

SOCKET client\_net::get\_socket\_fd()

{

return client\_fd;

}

bool client\_net::isConnected()

{

return connected;

}

//连接服务器

bool client\_net::connect()

{

//SOCKET fd = get\_socket\_fd();

if(isConnected())

return true;

connect\_thread\_running = true; //正在连接

client\_fd = socket(PF\_INET, SOCK\_STREAM, 0); //[BUG] 重新建立套接字 否则重新连接不上

for(int i = 0; i < 10 && connect\_thread\_running; i++)

{

//尝试连接

int ret = ::connect(client\_fd, (struct sockaddr\*)&client\_addr, sizeof (client\_addr));

//连接成功并接收服务器数据

if(ret == 0)

{

connected = true;

received = true;

connect\_thread\_running = false;

qDebug() << "连接服务器成功" << endl;

\_beginthreadex(NULL, 0, recv\_msg, this, 0, NULL);

return true;

}

qDebug() << "请求连接中:" << i << endl;

}

connect\_thread\_running = false;

return false;

}

//断开连接

void client\_net::disconnect()

{

if(!connected)

return;

connected = false;

received = false;

qDebug() << "客户端断开连接" << endl;

closesocket(client\_fd); //关闭套接字

}

则连接服务端后，网络游戏界面的核心代码如下所示：

//构造函数 初始化所有属性

internet\_game::internet\_game(int x,int y,client\_net \*client\_,QString room\_name,QWidget \*parent) :

client(client\_), //初始化client网络通信对象

QWidget(parent),

ui(new Ui::internet\_game)

{

ui->setupUi(this);

this->setFixedSize(1000,800);

this->move(x,y);

ui->Title->setText(room\_name); //展示房间名

setWindowTitle("五子棋网络对战");

setWindowIcon(QPixmap(":new/prefix1/img/Title1.png"));

ui->stackedWidget->setCurrentIndex(0);

ui->button\_black->hide();

ui->button\_white->hide();

ui->label\_victory->hide();

ui->Label\_your\_color->hide();

ui->button\_agree->hide();

ui->button\_refuse->hide();

ui->label\_wait\_answer->hide();

ui->label\_anwser->hide();

ui->label\_msg->hide();

white\_chess.load(":/new/prefix1/img/shiro.png");

black\_chess.load(":/new/prefix1/img/kuro.png");

board\_bg.load(":/new/prefix1/img/menubg1.png");

wait=false;

ban\_mouse=false;

prepare=false;

color=-1;

running=false;

turn=false;

timerId1=startTimer(500);

back.resize(0); //记录棋盘信息初始化

square = 800 / (chessboard\_size + 1); //格子边长赋值

//保存每个点的信息 [BUG]下标必须从0开始

for(int i = 0; i < chessboard\_size; i++)

{

chess\_info.push\_back(QVector<QPair<QRect, int>>());

for(int j = 0; j < chessboard\_size; j++)

{

//设置每个点的点击范围

chess\_info[i].push\_back(QPair<QRect, int>(QRect((i+1)\*square - square \* 1.25 / 3, (j+1)\*square - square \* 1.25 / 3, square/3\*2.5, square/3\*2.5), -1));

}

}

}

在“网络游戏”中点击返回主页，在点击“本地游戏”，进入“本地游戏”界面。本地游戏界面如上图12所示：



图12本地游戏界面

本地游戏界面的核心代码如下所示：

GameWin::GameWin(QWidget \*parent) :

QWidget(parent),

ui(new Ui::GameWin)

{

ui->setupUi(this);

setWindowIcon(QPixmap(":new/prefix1/img/Title1.png"));

setWindowTitle("五子棋本地对战");

white\_chess.load(":/new/prefix1/img/shiro.png");

black\_chess.load(":/new/prefix1/img/kuro.png");

board\_bg.load(":/new/prefix1/img/btnbg.jpg");

square = 800 / (chessboard\_size + 1); //格子边长赋值

//保存每个点的信息 [BUG]下标必须从0开始

for(int i = 0; i < chessboard\_size; i++)

{

chess\_info.push\_back(QVector<QPair<QRect, int>>());

for(int j = 0; j < chessboard\_size; j++)

{

//设置每个点的点击范围

chess\_info[i].push\_back(QPair<QRect, int>(QRect((i+1)\*square - square\*1.25/3, (j+1)\*square - square\*1.25/3, square/3\*2.5, square/3\*2.5), -1));

}

}

initialization();

}

而客户端大致分为以上这几大模块，其中还有一些具体的功能：

例如网络游戏中玩家准备功能。创建房间或者加入房间后，玩家可以点击准备按钮，当玩家双方都点击准备按钮后，即可进入游戏，准备功能界面如图13所示：



图13玩家准备界面

//准备按钮点击

void internet\_game::on\_Button\_prepare\_clicked()

{

int ret;

//Sleep(500);

ret = client->send\_msg("prepare"); //发送准备请求

if(ret==SOCKET\_ERROR)

return;

//此时的prepare是点击前的状态

else if(prepare)

{

ui->label\_prepare->setText("请准备...");

ui->label\_prepare->setStyleSheet("QLabel{color:rgba(255,0,0,0.6);}");

}

else

{

ui->label\_prepare->setText("等待对手准备...");

ui->label\_prepare->setStyleSheet("QLabel{color:rgba(0,255,0,0.6);}");

}

prepare=!prepare; //更新准备状态

//更改按钮文字

if(prepare)

{

ui->Button\_prepare->setText("取消准备");

}

else

{

ui->Button\_prepare->setText("准备");

}

}

进入游戏后，玩家会进行选择，选择谁是先手，谁为后手。“黑棋”为先手，“白棋”为后手。玩家先后手如图14所示：

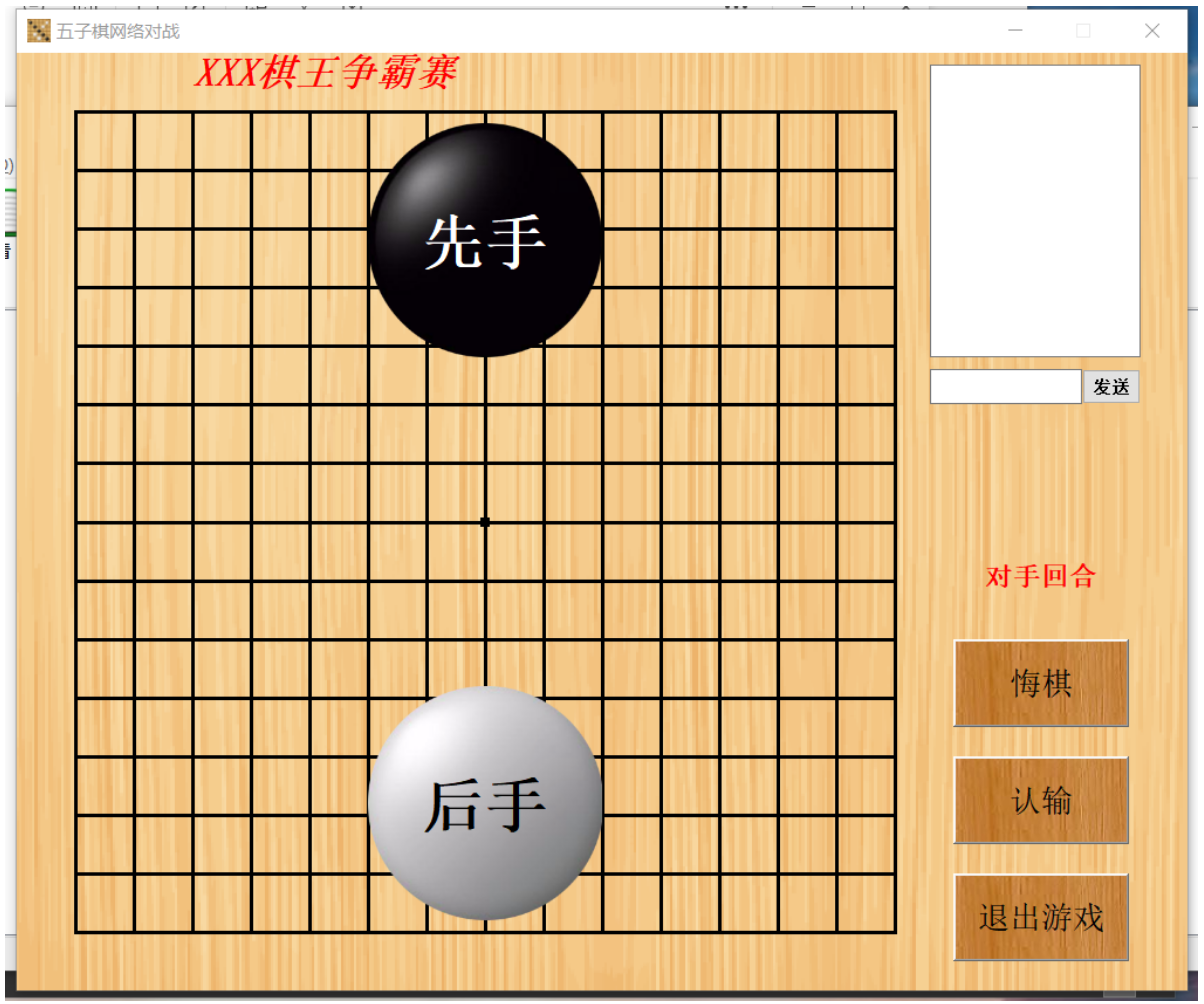


图14玩家先后手界面

玩家先后手核心代码如下所示：

//先手按钮点击

void internet\_game::on\_button\_black\_clicked()

{

if(color != -1)

return;

client->send\_msg("color1");

}

//后手按钮点击

void internet\_game::on\_button\_white\_clicked()

{

if(color != -1)

return;

client->send\_msg("color0");

}

在对局过程中，玩家若点击错误，则可以点击悔棋功能。若有急切的事情，则可点击人数功能，悔棋、认输功能如图15所示：



图15悔棋、认输界面

认输，悔棋核心代码如下所示：

//悔棋按钮点击

void internet\_game::on\_btn\_back\_clicked()

{

if(!running || color == -1 || back.empty())

return;

client->send\_msg("OB"); //向服务器发送悔棋请求

ui->label\_wait\_answer->setText("请等待对手做出回应");

ui->label\_wait\_answer->show();

ban\_mouse = true; //禁用鼠标

wait = true; //等待状态

}

//拒绝悔棋按钮

void internet\_game::on\_button\_refuse\_clicked()

{

client->send\_msg("OB0"); //向服务器发送悔棋拒绝信息

wait\_over();

}

//同意悔棋按钮

void internet\_game::on\_button\_agree\_clicked()

{

client->send\_msg("OB1"); //向服务器发送悔棋同意信息

if(!turn) //如果不是己方回合 则向前退回两步

{

go\_back();

go\_back();

}

else

go\_back();

wait\_over();

}

当玩家结束游戏之后，则可点击退出游戏。即可结束对局，退出房间。退出游戏核心代码如下所示：

void internet\_game::closeEvent(QCloseEvent \*event)

{

qDebug() << "网络游戏页面关闭..." << endl;

if(running) //游戏运行中退出,通知对手已退出游戏

{

client->send\_msg("OR"); //通过服务器将退出消息转发给对手

}

emit gameOver(); //发送游戏结束信号，触发网络页面析构

}

而五子棋判断胜负的核心代码如下所示：

for(int j = 0; j <= 4; j++)

{

a += dir2[i][0];

b += dir2[i][1];

//[BUG] chess\_info[a][b].second == black 表示该点上的棋子颜色与当前落子方的颜色一致

if(a >= 0 && b >= 0 && a < chess\_info.size() && b < chess\_info.size() && chess\_info[a][b].second == black)

sum++;

else

break;

}ui->label\_opponent->setStyleSheet("QLabel{""border-image:url(:/new/prefix1/img/none.png);""}");

if(sum >= 4) //五子相连 游戏结束

{

if(black)

ui->label\_victory->setText("黑方胜利");

else

ui->label\_victory->setText("白方胜利");

on\_Button\_prepare\_clicked(); //(游戏结束)调用准备按钮函数 相当于取消准备

ui->label\_prepare->show();

ui->label\_victory->show();

ui->stackedWidget->setCurrentIndex(0);

color=-1;

running=false;

return ;

}

客户端的主要成员变量有：

QVector<QVector<QPair<QRect, int>>> chess\_info ：棋盘信息 记录每个点的落子颜色与落子范围等

QStack<QPair<int, int>> back ：所有落子信息 悔棋用

bool wait ：用于游戏运行中，一方发出悔棋、新游戏的请求后发出方持续的状态，这个状态下发出方将只等待处理对方的回应信息

bool turn ：用于游戏运行中，你的回合，为你的回合时才能下棋，但此时，依然可以点击悔棋、新游戏等按钮

# 六、系统测试

坚持下五子棋，不仅能提高智力活动的速度，而且在逻辑推理思维中能提高从不同的角度、有独创性地解决问题的能力。 这一点对于开发学龄前儿童及中小学生的智力亦是很有帮助的。 下五子棋则有助于提高注意力。 下棋时，“一心不能二用”，聚精会神可使脑细胞活动的有序化程度增强，同时促进人的思维、心理活动以及循环系统、呼吸系统产生良性调节。 总之，五子棋是一种有助于身心健康的智力运动。

## （一）软件测试的定义

软件测试是描述一种用来促进鉴定软件的正确性、完整性、安全性的过程。换句话说，软件测试是一种实际输出与预期输出之间的审核或者比较过程。软件测试的定义是：在正规的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。软件测试是伴随着软件的产生而产生。早期的软件开发过程中软件规模都很小、复杂程度低，软件开发的过程混乱无序、相当随意，测试的含义比较狭窄，开发人员将测试等同于“调试”，目的是纠正软件中已经知道的故障，常常有开发人员自己完成这部分工作。

（二）软件测试的方法

软件测试方法有两种方法：黑盒测试和白盒测试。

1.白盒测试法

白盒测试方法也称结构测试或逻辑驱动测试，是一种结构化测试、逻辑驱动测试或基于过程细节的测试方法，即对系统软件的一种细节测试方法。白盒测试法是事先设定特定的条件或设计循环的用例进行测试，它主要针对系统的内部结构采用穷举路径进行测试，因此测试人员必须充分了解系统程序的逻辑结构，通过测试用例获取测试结果。

2.黑盒测试法

黑盒测试也称为功能测试，它是指是基于已知程序软件的功能结构，通过试验的方法来检测所有的功能是否都满足要求。利用黑盒测试法对系统程进行测试时，把系统看作是一个整体，不去理解其内部的是采用什么结构，只是把它当成一个无法打开的盒子。通过黑盒测试方法，可以判断系统是否达到分析阶段的用户要求、系统各项功能是否能正常使用、系统是否能正确接收数据和输出结果等。

## （三）测试用例

1.用户注册测试

表3 用户注册测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例 | 输入 | 输出 | 结果 |
| 1 | 不输入用户名和密码，点击注册 | 注册失败 | 通过 |
| 2 | 输入用户名，不输入密码 | 注册失败 | 通过 |
| 3 | 输入重复的用户名 | 注册失败 | 通过 |
| 4 | 输入新的用户名和密码 | 显示注册成功 | 通过 |

2. 登录测试

表4 登录测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 数据 | 输出 | 结果 |
| 用户登录 | 输入错误的用户名或密码，点击登录 | 用户名：xiaoming  密码：12345 | 登陆失败 | 通过 |
| 输入正确的用户名和密码 | 用户名：xiaoming  密码：123456 | 登录登陆成功 | 通过 |

1. 连接服务端测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例 | 服务端打开状态 | 输出 | 结果 |
| 连接服务端 | 未打开服务端 | 请连接服务器 | 通过 |
| 打开服务端 | 连接服务器-《创建或加入房间》 | 通过 |

1. 下棋先后手测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例 | 点击 | 显示 | 结果 |
| 先后手 | 黑棋图片 | 你为先手，所掌为黑棋 | 通过 |
| 白棋图片 | 你为后手，所掌为白棋 | 通过 |

1. 主页面点击测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例 | 点击 | 显示 | 结果 |
| 1 | 本地游戏 | 切换到本地游戏界面 | 通过 |
| 2 | 网络游戏 | 切换到网络游戏界面 | 通过 |
| 3 | 退出游戏 | 游戏结束，界面消失 | 通过 |
| 4 | 关于项目 | 制作人：小明 |  |

# 总结与展望

总结

经过一个多月地忙碌，我的毕业设计课题也终将告一段落。点击运行，也基本达到预期的效果，虚荣的成就感在没人的时候也总会冒上心头。但由于能力和时间的关系，总是觉得有很多不尽人意的地方，譬如功能不全、底层代码的不合理等问题数不胜数。这些都有待进一步完善。

通过对Socket和C++框架的研究，做出本系统总体设计和详细设计后，利用QT和MySQL数据库做出了一个基于C++的五子棋游戏的设计与实现。根据自己对游戏和五子棋的熟悉，虽然我的技术不是很清大，甚至可以说很弱，但是经过我这么长的努力和学习，并在付红杰老师的培养下，也算是勉强完成了这个五子棋系统，不过本系统的个别功能和模块还不是很完善，有待在以后的工作和学习中逐渐的改进；也希望在以后的工作和学习中做的越来越好。

展望

本课题所开发的就基于C++的五子棋游戏的设计与实现，虽然实现了C++下 MySQL数据库的数据管理和资源共享，但由于能力有限，所以系统还存在一些不足，还需今后进一步完善与改进，比如：没有对数据进行进一步的处理和优化，如何提高运行速度，以及如何完善五子棋的联机界面经常与对方玩家的界面不一致等方面需要进一步加强和完善。对于五子棋复杂的体系，目前所实现的功能模块相对独立，需要进一步将各个子模块，进一步整合成一个有机的整体。其主要完善与改进的方面如下：

（1）需要将单机模式的自己一个人下棋，修改成人机对战。

（2）项目分析阶段，一定要预测系统可能存在的问题，早发现问题早解决，避免系统开发工作的无效性。

（3）系统开发过程中，要充分分析与评估系统安全隐患，系统开发中，设计系统的安全体系统，保障系统投入使用的安全性。

（4）系统在运行时，要处理大量的数据，因此在数据库存设计中，要尽可能考虑数据的紧密性与联系性，增强数据的在逻辑上的关联。

# 致谢

本文是在付红杰老师的悉心指导下完成的。在整个课题阶段，无论是论文的选题、开展还是论文的撰写，付红杰老师都给予极大的关怀和悉心的指导。在论文具体写作阶段，导师付红杰老师多次对我论文的结构与提纲提出许多建议，从论文内容到形式都提出了严格的要求，让我体会到了学术的严谨性与严肃性。导师付红杰老师严谨的治学态度、开拓创新的工作作风、高昂的工作热情以及对学术执著的追求精神使我受益终生，同时严于律己、宽以待人、淡泊名利的优秀品德以及任劳任怨和诲人不倦的精神不断地教育我、激励我，使我受益匪浅。在此论文完稿之际，谨向付红杰老师的辛勤工作和给予我的培养教育致以最衷心的感谢！ 在此我还要深深感谢支持我、关心我的家人，感谢我的父母对我的养育之情，在学习和生活上给予的支持与理解，是你们鼓励我不断地上进，在我学习时给我支持和力量。感谢在这大学生活中身边的同学和朋友，他们在我论文写作期间给予我无私的帮助。最后，借此机会向所有支持、帮助和关心我的各位领导、老师和同学表示深深的谢意！

# 参考文献

[1] 谭浩强,C程序设计.3版.北京:清华大学出版社.2005.

[2] 谭浩强,C++程序设计.北京:清华大学出版社,2004.

[3]郑阿奇、丁有和,Visual C++应用教程.北京:人民邮电出版社.

[4] 潘爱民:《Microsoft Visual C++.NET技术内幕(第6版)》,清华大学出版社,1999年

10月.

[5] 汪晓平、钟军:《Visual C++网络通信协议分析与应用实现》,人民邮电出版社,2001年1

月.

[6]刘宗田、邢大红、孙慧杰:《C++编程思想》,机械工业出版社,2002年10月.

[7]Jesse Liberty著:《21天学通C++》,人民邮电出版社,2003年5月.

[8] 侯捷、孟岩:《C++标准程序库》,华中科技大学出版社,2006年11月.

[9] Charles Petzold:《Windows程序设计》,北京大学出版社,2002年1月.

[10] 艾朱斯特(Alan Ezust)、艾朱斯待(Paul Ezust)。C++QT设计模式。北京:电子工业出版社,2012.7

[11] 萨默菲尔徳(Mark Summerfield).QT高级编程。北京:电子工业出版社

[12]kennethA.ReekC和指针人民邮电出版社。北京2008

[13]杰普软件嵌入式软件课程体系标准及培训计划