

华北电力大学

课程设计报告

(2018--2019 年度第二学期)

名 称: Python 课程设计

题 目: scrapy 框架下爬取 51job 网站

院 系: 控制与计算机工程学院

班 级: 软件 1801

学 号: 120181080223

学生姓名: 姚鹏飞

指导教师: 熊建国

设计周数: 1 周

成 绩: _____

日期: 2020 年 6 月 25 日

一、课程设计(综合实验)的目的与要求

通过课程设计的实践训练,加强学生对 Python 语言基础知识、不同模块的理解运用、利用模块进行程序设计并且达成目标,提高学生综合运用所学知识分析问题、解决问题的能力。

具体如下:

1. 根据题目要求进行需求分析,确定系统的功能需求、数据需求、性能需求等;
2. 根据分析的结果进行软件设计,包括类的设计、数据设计等;
3. 采用 Python 语言实现系统的主要功能;
4. 撰写课程设计报告,要求内容完整、格式规范。

二、设计(实验)正文

1. 需求分析

1.1 功能需求

目标要实现的功能包括:

- (1) 利用 scrapy 框架从 <http://www.51job.com> 网站爬取关于 Python 开发工程师的工作岗位信息,并按照一定结构保存到数据库中。
- (2) 根据爬取的工作信息计算北京地区的 Python 开发工程师的平均薪资。

1.2 数据需求

系统需要获取并保存的数据包括:

- (1) 从 <http://www.51job.com> 网站爬取关于 Python 开发工程师的工作岗位信息

2. 程序总体设计

2.1 数据库的设计

工作岗位数据表:

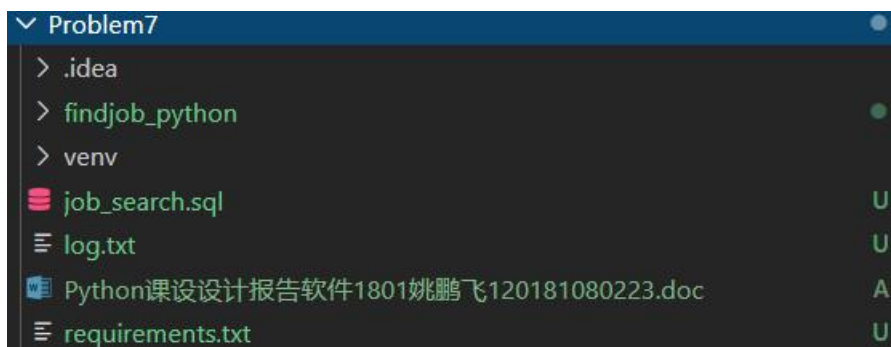
- (1) 数据库名: job_search
- (2) 表名: pythondeveloperjob

数据库表结构说明:

字段名称	字段类型	字段约束	字段说明
id	INT	主键、自增	用户唯一 id
JobTitle	VARCHAR(255)	无	工作名称
CompanyName	VARCHAR(255)	无	公司名称
WorkPosition	VARCHAR(255)	无	工作地点
Salary	FLOAT(20)	无	薪资水平
PublishTime	VARCHAR(255)	无	发布时间

2.2 程序结构

程序位置存放于 exell/Problem7 中。一级目录结构如下:



文件名	文件类型	说明
.idea	文件夹	存放 Pycharm 配置信息
findjob_python	文件夹	用于记录棋子标签在判定矩阵中的列坐标
venv	文件夹	用于记录棋子标签在判定矩阵中的价值
job_search.sql	sql 脚本	
log.txt	txt 文本	
报告.doc	word 文档	

(1) NJLabel 类：表示棋盘中的棋子，包含的主要成员变量和成员方法如表 1 所示。

表 1 NJLabel 类的规格说明

序号	成员变量/成员方法	说明
1	int posx	用于记录棋子标签在判定矩阵中的行坐标
2	int posy	用于记录棋子标签在判定矩阵中的列坐标
3	int value	用于记录棋子标签在判定矩阵中的价值

2.3 界面设计

说明界面设计的思路及所用组件。

界面主要分为两个，一个是登录界面，一个是棋子对弈界面：

(1) 登录界面

登录界面采用了较为简洁的设计，主要就是一句欢迎词，两个提示输入的标签，两个输入框和两个按钮。

设计的思路是密码采取保密形式输入。两个按钮分别是登录和注册：如果点击的是登录按钮，输入的帐号不存在，输入的帐号密码不匹配，输入的帐号已登录都会对应不同的提示，直到输入正确才可以进入到棋子对弈界面；如果是注册界面，如果帐号已经注册，则会有提示重新注册，否则注册成功就跳转到棋子对弈界面。

(2) 棋子对弈界面

棋子对弈界面设计的较为复杂，首先是采用了分层容器的方法，将不同的类中制作好的面板分层次地装载到妻子对弈界面中。按照从下到上的顺序，依次插入了背景图层，棋盘面板，棋子操作层面板，聊天窗口面板，按钮组件面板，我方信息面板，敌方信息面板。

设计思路为通过不同层次的堆叠，真正地使得用户在使用界面时能够感觉到这是一个下棋的界面。并且不同的界面之间也会有交互。例如：点击按钮组件面板中的更换背景能更改背景图层的背景，点击棋子操作层来触发大部分对弈事件，聊天面板的输入框和发送按钮的组合可以进行端口之间的聊天。

3. 系统实现

3.1 开发技术

(1) graphics2D:

利用 graphics2D 画了棋盘，继承 JPanel 的 QIPAN 类重写了 protected void paintComponent(Graphics f)，把 graphics 强制转换成 graphics2D，利用 drawLine(int x, int y, int m, int n), drawRect(int x, int y, int w, int h), fillRect(int x, int y, int w, int h)画出了界面。在使用该技术时一共遇到了两个问题：

第一点是由于画的线像素宽度都很小，只有 1, 2, 4。直线还稍好，如果是斜线，粗细不一，锯齿状的问题就异常严重。查阅资料后，调用了 g.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON)用于抗锯齿较为完美的解决了这个问题，但依然存在一定的粗细不一。而文字的输入开了文字的抗锯齿之后能够很完美地显示；

第二点是由 graphics2D 或者是 graphics 构造的界面在界面上发生操作时容易发生变形，并且变形问题易与 socket 线程互相堵塞从而导致程序白屏卡死。在查阅资料，调用了 g.repaint()后完美解决了线程会卡死白屏以及界面变形的问题。

(2) void javax.swing.JComponent.setOpaque(boolean isOpaque):

所有的面板和标签构成的图层在分层容器中叠加起来时，如果不对不透明度进行设置，那么叠加在一起的图层就只能够显示最上方插入的面板。在这种情况下，Opaque 的值设为 false 时，在没有直接插入组件的面板位置就会呈现出透明的样式，从而透出下面面板的状况。

(3) qizi[i][j].setBorder(BorderFactory.createLineBorder(Color)):

在棋盘上，如果单独点击棋子，在选中时没有特殊标记时会不够显眼，甚至由于 java 的 swing 局限性容易出现没有点击到目标标签的情况，只能通过 console 界面的检查信息来查看。在调用了该方法并且辅之以一定的算法之后，可以较好地反应选中情况和棋子历史操作。

(4) JLayeredPane:

在界面的设计过程中，在加入了棋子操作层没有办法在同样的位置再次插入一个面板，查阅资料之后发现了分层容器。分层容器类似于 PS 中的图层概念，通过不同顺序的堆叠来显示出不同的结果。

(4) Clip clip = AudioSystem.getClip():

利用 clip 类播放音乐。

3.2 系统主界面

界面如图 1 所示。

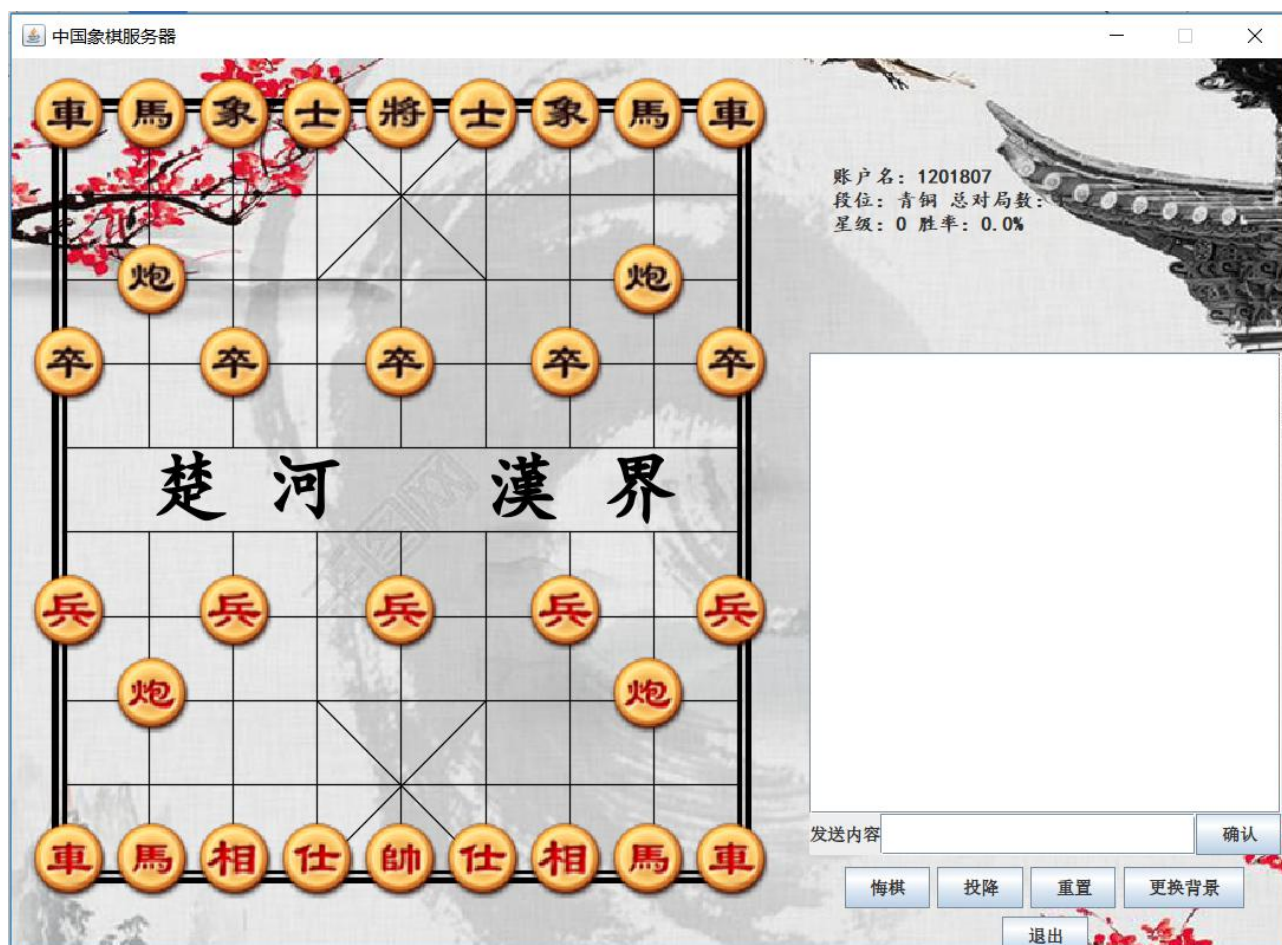


图1 系统主界面

该界面分为背景图层，棋盘面板，棋子操作层面板，聊天窗口面板，按钮组件面板，我方信息面板。而操作具体可分为左半区和右半区。

左半区主要是棋子对弈的操作。当轮到我方下棋时，单击棋子会出现对应持方颜色的选中框（即红方为红色选中框，黑方为黑色选中框），再次单击另一个位置后进行判断：如是合法操作，则执行；如果是点到了友方棋子，则更换选中对象；如果操作不合法，则会重置选中情况。所进行的操作将通过 socket 端口实时传输给另一方，通过读取编码的字符串并解码后反映到棋盘上。

右半区上方可查看自己帐号的基本信息，在下方主要是聊天窗口和各类按钮。确认键用于传输文本框内容来交互聊天信息，双方的消息都会实时地传输到聊天窗口中。而下方的按钮则对应各类操作。悔棋按钮可随时点击，但是连续点击没有效果，可以返回上一步的操作；投降按钮可以随时投降用来认输，但是会受到严厉的分数的惩罚；重置按钮用于一局结束之后点击重置即可开始下一局；更换背景则是在背景屏幕看腻了之后更换来获得新鲜体验，共有四张精修过的背景图可供选择；退出按钮则是用于退出帐号登录以保证帐号安全。

3.3 对弈功能功能的详细设计与实现

对弈功能实现逻辑如下：通过 GUIZE 类的 temp 标签来实时保存上一步触发鼠标事件的棋子。当 temp 为空时，并且操作权在我方时，点击我方棋子即进入选中状态并且保存其到 temp 中。当再次触发鼠标事件时，如果第二次操作的还是友方棋子，则将 temp 更新为刚刚点击的棋子；如果第二次点击的位置是之前棋子可以合法到达的，那么就执行该操作，否则就把 temp 置空。在合法情况时，两个棋子通过保存的坐标对应到 GUIZE 的判定矩阵中，将棋子的操作本质反应到矩阵的变化上，然后再对标签的各个指标进行更新，然后编码发送给端口，

编码字符串最前面加上“//”用于标注操作。发送完之后判定游戏是否结束并进行胜负结算。
发送给端口的信息接收后，直接解码翻译成矩阵坐标变换，再由矩阵变换来更新界面上的棋子。

具体代码如下：

```
//鼠标事件
if(!this.guize.kexing)
{
else
{
    if(this.guize.temp == null)//如果没有上一次操作
    {
        NJLabel t = (NJLabel)e.getSource();
        if(this.guize.judgejuzhen.zuobiao[t.posx][t.posy] > 0)
            //如果点击的是友方棋子，则记录 temp
        {
            this.guize.temp = t; //记录此次操作的标签
            System.out.println( "当前选中坐标: " + t.posx + " , " + t.posy + " 棋子价值为:" + t.value);
            t.setBorder(BorderFactory.createLineBorder(Color.red));
            this.qipan.repaint();
        }
    }
    else if(this.guize.temp != null) //如果已经记录了上一次操作
    {
        NJLabel t = (NJLabel)e.getSource();
        if(this.guize.judgejuzhen.zuobiao[t.posx][t.posy] > 0)//如果点击的是友方棋子，则重置 temp
        {
            System.out.println( "之前选中坐标: " + this.guize.temp.posx + " , " + this.guize.temp.posy
+ " 棋子价值为:" + this.guize.temp.value);
            this.guize.temp.setBorder(null);
            this.guize.temp = t;
            System.out.println( "当前选中坐标: " + this.guize.temp.posx + " , " + this.guize.temp.posy
+ " 棋子价值为:" + this.guize.temp.value);
            t.setBorder(BorderFactory.createLineBorder(Color.red));
            this.qipan.repaint();
        }
    }
    else//如果是其他空白位置或者敌方棋子
    {
        if(this.guize.CanMove(this.guize.temp.posx,this.guize.temp.posy, t.posx, t.posy))
            //判定行为是否合法
        {
            //t 更新为上一个选中的棋子
            System.out.println( "之前选中坐标: " + this.guize.temp.posx + " , " + this.guize.temp.
posy + " 棋子价值为:" + this.guize.temp.value);
            System.out.println( "当前选中坐标: " + t.posx + " , " + t.posy + " 棋子价值为:" + t.v
alue);
            System.out.println("在规则类中，之前判定矩阵值为:" + this.guize.judgejuzhen.zuobiao[t
```

```

his.guize.temp.posx][this.guize.temp.posy]);
    System.out.println("在规则类中，当前判定矩阵值为:" + this.guize.judgejuzhen.zuobiao[t.
posx][t.posy]);
    System.out.println("行为合法");
    //利用覆盖的方式顺便把吃子的事情给干了
    //注意还要传输双方王将的信息
    laozi = (ImageIcon) this.guize.temp.getIcon();
    shangcicaozuo[0] = this.guize.temp.posx;
    shangcicaozuo[1] = this.guize.temp.posy;
    shangcicaozuo[2] = this.guize.temp.value;
    xinzi = (ImageIcon) t.getIcon();
    shangcicaozuo[3] = t.posx;
    shangcicaozuo[4] = t.posy;
    shangcicaozuo[5] = t.value;
    if(this.guize.temp.value == 20)
    {
        this.guize.wfx = this.guize.temp.posx;
        this.guize.wfy = this.guize.temp.posy;
    }
    //把操作信息传给对面
    int oh = 11 - this.guize.temp.posx;
    int ol = 10 - this.guize.temp.posy;
    int nh = 11 - t.posx;
    int nl = 10 - t.posy;
    int ov = this.guize.temp.value;
    String chuanshu = "\\\" + oh + \" \" + ol + \" \" + ov + \" \" + nh + \" \" + nl ; // 构
造传输字符串
    System.out.println("服务器传过去的信息为:" + chuanshu);
    try {
        PrintWriter pWriter=new PrintWriter(socket.getOutputStream());
        pWriter.write(chuanshu+"\n");
        pWriter.flush();
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    this.qizicaozuoceng.fresh();
    t.setBorder(BorderFactory.createLineBorder(Color.red));
    this.guize.temp.setBorder(BorderFactory.createLineBorder(Color.red));
    this.qipan.repaint();
    t.value = this.guize.temp.value;
    t.setIcon((ImageIcon)this.guize.temp.getIcon());
    this.guize.temp.value = 0;
    this.guize.temp.setIcon(null);
    this.guize.judgejuzhen.zuobiao[t.posx][t.posy] = this.guize.judgejuzhen.zuobiao[this.guiz
e.temp.posx][this.guize.temp.posy];

```

```

        this.guize.judgejuzhen.zuobiao[this.guize.temp.posx][this.guize.temp.posy] = 0;
//判断胜负情况，注意之后还需要补充将帅对面的情况
        int sy = this.guize.success(this.guize);//胜负因子
        if(sy == 2)
            System.out.println("存在判定问题，请检查");
        else if(sy == 1)
        {
            int count = guize.shenglijiesuan(guize, sy);
            guize.shujukujiesuan(sy, count, zhxx, zhdw, zhzdjs, zhxj, zhsl);
            JOptionPane.showMessageDialog(this.frame, "你赢了!");
        }
        else if(sy == -1)
        {
            int count = guize.shenglijiesuan(guize, sy);
            guize.shujukujiesuan(sy, count, zhxx, zhdw, zhzdjs, zhxj, zhsl);
            JOptionPane.showMessageDialog(this.frame, "你输了!");
        }
        this.guize.temp = null;
        this.qipan.repaint();
        this.guize.kexing = false;
    }else//操作不合法{
        System.out.println( "之前选中坐标: " + this.guize.temp.posx + " , " + this.guize.temp.
posy + " 棋子价值为:" + this.guize.temp.value);
        System.out.println( "当前选中坐标: " + t.posx + " , " + t.posy + " 棋子价值为:" + t.v
alue);
        System.out.println("在规则类中，之前判定矩阵值为:" + this.guize.judgejuzhen.zuobiao[t
his.guize.temp.posx][this.guize.temp.posy]);
        System.out.println("在规则类中，当前判定矩阵值为:" + this.guize.judgejuzhen.zuobiao[t
posx][t.posy]);
        System.out.println("行为不合法");
        this.guize.temp.setBorder(null);
        this.guize.temp = null;
        this.qipan.repaint();
    }
}

//接收编码字符串的操作
if(zifuchuan.charAt(0) == '\\')
//反斜杠代表棋子操作， “”
//利用\字符来区分棋子操作和聊天内容还有其他按钮
{
    String s = zifuchuan.substring(1);

```



```

System.out.println( s + "客户端接收到的信息为（去第一位）： " );
String[] str = s.split("\\s+");
int[] shuzhishuzu = new int[10];
for(int i = 0;i < str.length; i++) {
    System.out.println(str[i] + "\n");
}
for(int i = 0;i < str.length; i++) {
    shuzhishuzu[i] = Integer.parseInt(str[i]);
}
NJLabel laozi = dl.getjiemian().qizicaozuoceng.qizi[shuzhishuzu[0]][shuzhishuzu[1]];
//之前的那个坐标
NJLabel xinzi = dl.getjiemian().qizicaozuoceng.qizi[shuzhishuzu[3]][shuzhishuzu[4]];
dl.getjiemian().shangcicaozuo[0] = laozi.posx;
dl.getjiemian().shangcicaozuo[1] = laozi.posy;
dl.getjiemian().shangcicaozuo[2] = laozi.value;
dl.getjiemian().laozi = (ImageIcon) laozi.getIcon();
dl.getjiemian().shangcicaozuo[3] = xinzi.posx;
dl.getjiemian().shangcicaozuo[4] = xinzi.posy;
dl.getjiemian().shangcicaozuo[5] = xinzi.value;
dl.getjiemian().xinzi = (ImageIcon) xinzi.getIcon();
ImageIcon icon = (ImageIcon)laozi.getIcon();
xinzi.value = laozi.value;
laozi.value = 0;
xinzi.setIcon(icon);
laozi.setIcon(null);
dl.getjiemian().qipan.repaint();
dl.getjiemian().guize.judgejuzhen.zuobiao[shuzhishuzu[3]][shuzhishuzu[4]] = dl.getjiemian().
guize.judgejuzhen.zuobiao[shuzhishuzu[0]][shuzhishuzu[1]];
dl.getjiemian().guize.judgejuzhen.zuobiao[shuzhishuzu[0]][shuzhishuzu[1]] = 0;
if(shuzhishuzu[2] == -20 || shuzhishuzu[2] == 20) {
    dl.getjiemian().guize.dfx = shuzhishuzu[3];
    dl.getjiemian().guize.dfy = shuzhishuzu[4];
}
dl.getjiemian().qizicaozuoceng.qizi[shuzhishuzu[0]][shuzhishuzu[1]].setBorder(BorderFactory.c
reateLineBorder(Color.red));;
dl.getjiemian().qizicaozuoceng.qizi[shuzhishuzu[3]][shuzhishuzu[4]].setBorder(BorderFactory.crea
teLineBorder(Color.red));;
//记录操作之后判定胜负
int sy = dl.getjiemian().guize.success(dl.getjiemian().guize);//胜负因子
if(sy == 2)
    System.out.println("存在判定问题，请检查");
else if(sy == 1)
{
    int count = dl.getjiemian().guize.shenglijiesuan(dl.getjiemian().guize, sy);
    dl.getjiemian().guize.shujukujiesuan(sy, count, dl.getjiemian().zhxx, dl.getjiemian().zhdw, dl.
getjiemian().zhzdjs, dl.getjiemian().zhxj, dl.getjiemian().zhsl);
}

```

```

JOptionPane.showMessageDialog(dl.getjiemian().frame, "你赢了! ");
}
else if(sy == -1)
{
    int count = dl.getjiemian().guize.shenglijiesuan(dl.getjiemian().guize, sy);
    dl.getjiemian().guize.shujukujiesuan(sy, count, dl.getjiemian().zhxx, dl.getjiemian().zhdw, dl.
getjiemian().zhzdjs, dl.getjiemian().zhxj, dl.getjiemian().zhsl);
    JOptionPane.showMessageDialog(dl.getjiemian().frame, "你输了! ");
}
dl.getjiemian().qipan.repaint();
dl.getjiemian().guize.kexing = true;
}

```

3.4 悔棋功能功能的详细设计与实现

在了解了对弈的实现之后，悔棋也就不难理解了，意味着只需要随时准备变量用于存储上次操作的编码字符串，每次操作之后都更新，一旦触发了悔棋的动作事件之后就立刻解码变量，然后返回到矩阵中，并且注意棋子图标等的变化。传输给对面时，以“GETBACK”标识是悔棋的操作。

//悔棋动作事件

```

else if(e.getSource() == anniuzujian.huiqi)
{
    this.guize.judgejuzhen.zuobiao[shangcicaozuo[0]][shangcicaozuo[1]] = shangcicaozuo[2];
    this.guize.judgejuzhen.zuobiao[shangcicaozuo[3]][shangcicaozuo[4]] = shangcicaozuo[5];
    qizicaozuoceng.qizi[shangcicaozuo[0]][shangcicaozuo[1]].setIcon(laozi);
    qizicaozuoceng.qizi[shangcicaozuo[3]][shangcicaozuo[4]].setIcon(xinzi);
    qizicaozuoceng.fresh();//悔棋就不考虑边框了
    this.guize.chageboolean();
    String s = "GETBACK";
    try {
        PrintWriter pWriter=new PrintWriter(socket.getOutputStream());
        pWriter.write(s + "\n");
        pWriter.flush();
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    for(int i = 0; i < 5; i++)
        shangcicaozuo[i] = 0;//只允许一次悔棋
}

//悔棋事件接受并响应
else if(zifuchuan.equals("GETBACK"))
{
    dl.getjiemian().guize.judgejuzhen.zuobiao[dl.getjiemian().shangcicaozuo[0]][dl.getjiemian().shang
cicaozuo[1]] = dl.getjiemian().shangcicaozuo[2];

```

```

dl.getjiemian().guize.judgejuzhen.zuobiao[dl.getjiemian().shangcicaozuo[3]][dl.getjiemian().shangcicaozuo[4]] = dl.getjiemian().shangcicaozuo[5];
dl.getjiemian().qizicaozuoceng.qizi[dl.getjiemian().shangcicaozuo[0]][dl.getjiemian().shangcicaozuo[1]].setIcon(dl.getjiemian().laozi);
dl.getjiemian().qizicaozuoceng.qizi[dl.getjiemian().shangcicaozuo[3]][dl.getjiemian().shangcicaozuo[4]].setIcon(dl.getjiemian().xinzi);
dl.getjiemian().qizicaozuoceng.fresh();//悔棋就不考虑边框了
dl.getjiemian().guize.chageboolean();
for(int i = 0; i < 5; i++)
    dl.getjiemian().shangcicaozuo[i] = 0;//只允许一次悔棋
}

```

3.5 投降功能功能的详细设计与实现

即跳过正常的操作之后结算的流程，直接将投降方的胜负因子记为-1，扣分 20，然后执行胜负结算。传输标识信息为“YOUWIN”。

//投降按钮动作事件

```
else if(e.getSource() == anniuzujian.touxiang)
```

```
{
```

//设置在投降之后重置，即等同于退出按钮

```
int sy = -1;
```

```
int count = -20;
```

```
guize.shujukujiesuan(sy, count, zhxx, zhdw, zhzdjs, zhxj, zhsl);
```

```
JOptionPane.showMessageDialog(this.frame, "你输了！");
```

```
String s = "YOUWIN";
```

```
try {
```

```
    PrintWriter pWriter=new PrintWriter(socket.getOutputStream());
```

```
    pWriter.write(s + "\n");
```

```
    pWriter.flush();
```

```
} catch (IOException e1) {
```

```
    // TODO Auto-generated catch block
```

```
    e1.printStackTrace();
```

```
}
```

```
guize.reset();
```

```
qizicaozuoceng.reset(0);
```

```
}
```

//投降事件响应

```
else if(zifuchuan.equals("YOUWIN"))
```

```
{
```

```
int sy = 1;
```

```
int count = 10;
```

```
dl.getjiemian().guize.shujukujiesuan(sy, count, dl.getjiemian().zhxx, dl.getjiemian().zhdw, dl.getjiemian().zhzdjs, dl.getjiemian().zhxj, dl.getjiemian().zhsl);
```

```
JOptionPane.showMessageDialog(dl.getjiemian().frame, "你赢了！");
```

```
dl.getjiemian().guize.reset();
```

```
dl.getjiemian().qizicaozuoceng.reset(0);
```

```
}
```

3.6 更换背景功能的详细设计与实现

此方法不需要两边互相通信。采用一个 4 长度的数组和一个模 4 的循环变量来控制界面的背景图层，点击到按钮时刷新至下一张背景图然后更换背景标签。

```
if(e.getSource() == anniuzujian.genghuanbeijing) //更换背景用
```

```
{
    this.background.setIcon(beijingtupian[count]);
    this.qipan.repaint();
    count = (count + 1) % 4;
}
```

3.7 重置功能的详细设计与实现

直接调用各个类的 reset 方法刷新即可。传递标识信息为“RESET”。

//重置按钮动作事件触发

```
else if(e.getSource() == anniuzujian.tuichu)
```

```
{
    guize.reset();
    qizicao zuoceng.reset(O);
    String s = "RESET";
    try {
        PrintWriter pWriter=new PrintWriter(socket.getOutputStream());
        pWriter.write(s +"\n");
        pWriter.flush();
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
```

//重置事件接受响应

```
else if(zifuchuan.equals("RESET"))
```

```
{
    dl.getjiemian().guize.reset();
    dl.getjiemian().qizicao zuoceng.reset(O);
}
```

其中，两个 reset 方法如下：

//GUIZE

```
void reset() //重置
```

```
{
    kexing = false;
    judgejuzhen = yuanshijuzhen;
```

```

wfx = 10;
wfy = 5;
dfx = 1;
dfy = 5;
temp = null;
}
//CAOZUOCENG
void reset(int flag) {
    //图片导入
    ImageIcon iconheiju = new ImageIcon("src//images//stype_1//b_c.png");
    ImageIcon iconheima = new ImageIcon("src//images//stype_1//b_m.png");
    ImageIcon iconheixiang = new ImageIcon("src//images//stype_1//b_x.png");
    ImageIcon iconheishi = new ImageIcon("src//images//stype_1//b_s.png");
    ImageIcon iconheijiang = new ImageIcon("src//images//stype_1//b_j.png");
    ImageIcon iconheipao = new ImageIcon("src//images//stype_1//b_p.png");
    ImageIcon iconheibing = new ImageIcon("src//images//stype_1//b_z.png");
    ImageIcon iconhongju= new ImageIcon("src//images//stype_1//r_c.png");
    ImageIcon iconhongma = new ImageIcon("src//images//stype_1//r_m.png");
    ImageIcon iconhongxiang = new ImageIcon("src//images//stype_1//r_x.png");
    ImageIcon iconhongshi = new ImageIcon("src//images//stype_1//r_s.png");
    ImageIcon iconhongshuai = new ImageIcon("src//images//stype_1//r_j.png");
    ImageIcon iconhongpao = new ImageIcon("src//images//stype_1//r_p.png");
    ImageIcon iconhongbing = new ImageIcon("src//images//stype_1//r_z.png");
    for(int i = 1; i <= 9; i++)
        qizi[2][i].setIcon(null);
    for(int i = 1; i <= 9; i++)
        qizi[5][i].setIcon(null);
    for(int i = 1; i <= 9; i++)
        qizi[6][i].setIcon(null);
    for(int i = 1; i <= 9; i++)
        qizi[9][i].setIcon(null);
    for(int i = 2; i <= 9; i = i + 2)
    {
        qizi[4][i].setIcon(null);
        qizi[7][i].setIcon(null);
    }
    qizi[3][1].setIcon(null);qizi[3][3].setIcon(null);qizi[3][4].setIcon(null);qizi[3][5].setIcon(null);qizi
[3][6].setIcon(null);qizi[3][7].setIcon(null);qizi[3][9].setIcon(null);
    qizi[8][1].setIcon(null);qizi[8][3].setIcon(null);qizi[8][4].setIcon(null);qizi[8][5].setIcon(null);qizi
[8][6].setIcon(null);qizi[8][7].setIcon(null);qizi[8][9].setIcon(null);
    if(flag == 1)//红方为操作方
    {
        qizi[1][1].setIcon(iconheiju);
        qizi[1][9].setIcon(iconheiju);
        qizi[1][2].setIcon(iconheima);
    }
}

```

```

qizi[1][8].setIcon(iconheima);
qizi[1][3].setIcon(iconheixiang);
qizi[1][7].setIcon(iconheixiang);
qizi[1][4].setIcon(iconheishi);
qizi[1][6].setIcon(iconheishi);
qizi[1][5].setIcon(iconheijiang);
qizi[3][2].setIcon(iconheipao);
qizi[3][8].setIcon(iconheipao);
qizi[4][1].setIcon(iconheibing);
qizi[4][3].setIcon(iconheibing);
qizi[4][5].setIcon(iconheibing);
qizi[4][7].setIcon(iconheibing);
qizi[4][9].setIcon(iconheibing);
//红方
qizi[10][1].setIcon(iconhongju);
qizi[10][9].setIcon(iconhongju);
qizi[10][2].setIcon(iconhongma);
qizi[10][8].setIcon(iconhongma);
qizi[10][3].setIcon(iconhongxiang);
qizi[10][7].setIcon(iconhongxiang);
qizi[10][4].setIcon(iconhongshi);
qizi[10][6].setIcon(iconhongshi);
qizi[10][5].setIcon(iconhongshuai);
qizi[8][2].setIcon(iconhongpao);
qizi[8][8].setIcon(iconhongpao);
qizi[7][1].setIcon(iconhongbing);
qizi[7][3].setIcon(iconhongbing);
qizi[7][5].setIcon(iconhongbing);
qizi[7][7].setIcon(iconhongbing);
qizi[7][9].setIcon(iconhongbing);
}
else if(flag == 0)//黑方为操作方
{
    qizi[10][1].setIcon(iconheiju);
    qizi[10][9].setIcon(iconheiju);
    qizi[10][2].setIcon(iconheima);
    qizi[10][8].setIcon(iconheima);
    qizi[10][3].setIcon(iconheixiang);
    qizi[10][7].setIcon(iconheixiang);
    qizi[10][4].setIcon(iconheishi);
    qizi[10][6].setIcon(iconheishi);
    qizi[10][5].setIcon(iconheijiang);
    qizi[8][2].setIcon(iconheipao);
    qizi[8][8].setIcon(iconheipao);
    qizi[7][1].setIcon(iconheibing);

```

```

qizi[7][3].setIcon(iconheibing);
qizi[7][5].setIcon(iconheibing);
qizi[7][7].setIcon(iconheibing);
qizi[7][9].setIcon(iconheibing);
//红方
qizi[1][1].setIcon(iconhongju);
qizi[1][9].setIcon(iconhongju);
qizi[1][2].setIcon(iconhongma);
qizi[1][8].setIcon(iconhongma);
qizi[1][3].setIcon(iconhongxiang);
qizi[1][7].setIcon(iconhongxiang);
qizi[1][4].setIcon(iconhongshi);
qizi[1][6].setIcon(iconhongshi);
qizi[1][5].setIcon(iconhongshuai);
qizi[3][2].setIcon(iconhongpao);
qizi[3][8].setIcon(iconhongpao);
qizi[4][1].setIcon(iconhongbing);
qizi[4][3].setIcon(iconhongbing);
qizi[4][5].setIcon(iconhongbing);
    qizi[4][7].setIcon(iconhongbing);
qizi[4][9].setIcon(iconhongbing);
}
}

```

3.8 退出登录功能功能的详细设计与实现

退出时，需要打开与数据库的连接，将“是否在线”的值更改为“否”，然后才可以关闭程序，不然就需要重新回到数据库去修改值了。

```

else if(e.getSource() == anniuzujian.td)
{
    ResultSet rs = null;
    Statement stmt = null;
    Socket socket;
    Connection conn = null;
    try {
        Class.forName("com.hxxt.sql.access.AccessDriver");
        String url = "jdbc:Access:///d:/中国象棋帐号数据库.accdb";
        conn = DriverManager.getConnection(url, "", "");
        stmt = conn.createStatement();
    }catch(Exception xe) {
        xe.printStackTrace();
    }
    String sql = "update 帐号信息 set 是否在线 = '否' where 账户 = '" + zhxx + "'";
    try {
        stmt.execute(sql);
        if(stmt != null)

```

```

        stmt.close();
        if(conn != null)
            conn.close();
    } catch (SQLException we) {
        // TODO Auto-generated catch block
        we.printStackTrace();
    }
}

```

3.9 聊天功能功能的详细设计与实现

每次点击发送按钮后，会读取输入框中的文本，通过端口传递，辅之以时间追加显示在聊天面板上。传输信号为无标识字符串。

//聊天按钮动作事件

```

else if(e.getSource() == cheat.fasonganniu)
{
    String s = cheat.shurukuang.getText();
    Calendar c = Calendar.getInstance();
    cheat.duihuakuang.append(c.get(Calendar.HOUR_OF_DAY) + ":" + c.get(Calendar.MINUTE) + ":" + c.get(Calendar.SECOND) + "\n");
    cheat.duihuakuang.append("我: " + s);
    cheat.duihuakuang.append("\n");
    qipan.repaint();
    try {
        PrintWriter pWriter=new PrintWriter(socket.getOutputStream());
        pWriter.write(s + "\n");
        pWriter.flush();
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}

```

```

        cheat.shurukuang.setText(null);
    }

```

//聊天事件接受响应

else//先把聊天框的写了，等会再加按钮

```

{
    Calendar c = Calendar.getInstance();
    dl.getjiemian().cheat.duihuakuang.append(c.get(Calendar.HOUR_OF_DAY) + ":" + c.get(Calendar.MINUTE) + ":" + c.get(Calendar.SECOND) + "\n");
    dl.getjiemian().cheat.duihuakuang.append("对手: " + zifuchuan);
    dl.getjiemian().cheat.duihuakuang.append("\n");
    dl.getjiemian().qipan.repaint();
}

```


3.10 登录界面的登录功能功能的详细设计与实现

登录按钮触发时，需要读取文本框和密码框中的字符串，然后与数据库进行查询比较：如果没有该账号，则提示账号不存在；如果密码不匹配，则提示密码输入错误；如果都匹配但是已经登录在线，则提示已登录。登录之后将是否在线更新为“是”。

```

if(e.getSource() == dengluanniu) {
//登录，匹配数据库，匹配失败就一直登录
    String zhxx = zhmk.getText();
    String mmxx = new String(mmk.getPassword());
    String sql = "select * from 帐号信息 where 账户 = '" + zhxx + "' and 密码 = '" + mmxx +
    """;
    try {
        rs = stmt.executeQuery(sql);
        if(rs.next()) {
            if(rs.getString(9).equalsIgnoreCase("是")) {
                JOptionPane.showMessageDialog(denglu, "该账号已登录！");
                zhmk.setText(null);
                mmk.setText(null);
            }else {
                //提取帐号信息，用于棋盘界面
                int zongjushu = rs.getInt(3);
                int shengchangshu = rs.getInt(4);
                int shuchangshu = rs.getInt(5);
                double shenglv = rs.getDouble(6);
                String duanwei = rs.getString(7);
                int xingji = rs.getInt(8);
                String xinxi = rs.getString(1);
                JOptionPane.showMessageDialog(denglu, "登录成功");
                String sql2 = "update 帐号信息 set 是否在线 = '是' where 账户 = '" + rs.getStri
ng(1) + """;
                stmt.execute(sql2);
                denglu.setVisible(false);
               jiemian = new JIEMIAN("中国象棋客户端", socket, xinxi, duanwei, zongjushu, xi
ngji, shenglv);
            }
            if(stmt != null)
                stmt.close();
            if(conn != null)
                conn.close();
        }
    }else {
        JOptionPane.showMessageDialog(denglu, "帐号或密码输入错误！");
        mmk.setText(null);
    }
}
}catch(Exception se) {
    se.printStackTrace();
}

```

```
    }
}
```

3.11 登录界面的注册功能功能的详细设计与实现

注册按钮动作事件触发后，读取文本框和密码框中的字符串，然后与数据库进行查询比较：如果帐号已经存在，则提示已经存在并且注册失败，否则以该账号和密码在数据库中新添一组数据，并将是否在线更改为“是”。

```
if(e.getSource() == zhuceanniu) {
    String zhxx = zhmk.getText();
    String mmxx = new String(mmk.getPassword());
    String sql1 = "select * from 帐号信息";
    String sql = "insert into 帐号信息(账户,密码,总对局数,胜率,段位,星级,是否在线) values('" + zh
xx + "','" + mmxx + "','O,1,'青铜','O,'是')";
    try {
        rs = stmt.executeQuery(sql1);
        //需要考虑是否帐号已经被注册
        while(rs.next()) {
            if(rs.getString(1).equalsIgnoreCase(zhxx)) {
                JOptionPane.showMessageDialog(denglu, "该账号已被注册!");
                zhmk.setText(null);
                mmk.setText(null);
                break;
            }
        }
        if(!rs.next()) {
            stmt.execute(sql);
            JOptionPane.showMessageDialog(denglu, "注册成功");
            denglu.setVisible(false);
            jiemian = new JIEMIAN("中国象棋客户端",socket, zhxx, "青铜", 0,0, 1);
            if(stmt != null)
                stmt.close();
            if(conn != null)
                conn.close();
        }
    } catch (Exception e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
```

3.12 棋子操作合法性功能的详细设计与实现

在确定了操作中的两个坐标之后，首先判断走棋的棋子的坐标对应判定矩阵中的值为何，然后根据棋子价值译码得到对应棋子。然后再根据不同的棋子判断走法是否合理。

```
public boolean CanMove(int x1, int y1, int x2, int y2) {
```

```

int value = judgejuzhen.zuobiao[x1][y1];
if(!kexing)//没有控制权
    return false;
else if(value>=1 && value<=5) //即操作对象为兵
    return BING(x1,y1,x2,y2);
else if(value == 6 || value == 7)
    return SHI(x1,y1,x2,y2);
else if(value == 8 || value == 9)
    return XIANG(x1,y1,x2,y2);
else if(value == 10 || value == 11)
    return MA(x1,y1,x2,y2);
else if(value == 12 || value == 13)
    return PAO(x1,y1,x2,y2);
else if(value == 14 || value == 15)
    return JU(x1,y1,x2,y2);
else if(value == 20)
    return WANG(x1,y1,x2,y2);
else
    return false;
}
//兵的判定规则
public boolean BING(int x1, int y1, int x2, int y2) {
    //考虑过河和不过河的情况
    if(judgejuzhen.zuobiao[x2][y2]>0)//如果终点是自家的子
        return false;
    if(x1>= 6)//没过河
    {
        if(y1 != y2)//水平方向发生了变化，不是向前走的
            return false;
        if(x2 != x1 - 1)//如果走的不只一步
            return false;

    }else//如果已经过河
    {
        if(y1 == y2)//如果走的是竖线
            if(x2 != x1 - 1)//如果走的不是一格
                return false;
        //由于棋盘上没有坐标为 0 和负数的，所以不会出现到头还能选择往前走的，故不考虑该情况
        else
            return true;
        else if(x1 == x2)//走的是横线
            if(y2 - y1 != 1 && y2 - y1 != -1)//如果不是左右横移一格
                return false;
            else
                return true;
    }
}

```

```

        else////斜线
            return false;
    }
    return true;
}
//士的判定规则
public boolean SHI(int x1, int y1, int x2, int y2) {
    if(judgejuzhen.zuobiao[x2][y2]>0)//如果终点是自家的子
        return false;
    if((x2 == 9 && y2 == 5) && ((y1 !=4 && y1 != 6) || (x1 != 8 && x1 != 10)))//如果落
    子在中间而开始不在两边
        return false;
    else if((x1 == 9 && y1 == 5) && ((y2 !=4 && y2 != 6) || (x2 != 8 && x2 != 10)))//如
    果开始在中间落子不在四角
        return false;
    return true;
}
//相的规则
public boolean XIANG(int x1, int y1, int x2, int y2) {
    int dx, dy;
    if(judgejuzhen.zuobiao[x2][y2]>0)//如果终点是自家的子
        return false;
    if(x1-x2>0)
        dx = x1-x2;
    else
        dx = x2-x1;
    if(y1-y2>0)
        dy = y1-y2;
    else
        dy = y2-y1;

    if(x2<=5)//如果相飞到另一半场
        return false;
    else if(dx != 2 || dy != 2)//如果不是斜两格,最初逻辑有误,已修改
        return false;
    else if(judgejuzhen.zuobiao[(x1+x2)/2][(y1+y2)/2] != 0)//如果路线中有子
        return false;
    return true;
}
//马的规则
public boolean MA(int x1, int y1, int x2, int y2) {
    int dx, dy;
    if(judgejuzhen.zuobiao[x2][y2]>0)//如果终点是自家的子
        return false;

```

```

if(x1-x2>0)
    dx = x1-x2;
else
    dx = x2-x1;
if(y1-y2>0)
    dy = y1-y2;
else
    dy = y2-y1;
if(dy == 1 && dx == 2)//如果是竖起来的日字
{
    if(judgejuzhen.zuobiao[(x1 + x2)/2][y1] != 0)//如果蹙马脚
        return false;
    else
        return true;
}
else if(dy == 2 && dx == 1)//如果是横躺着的日字
{
    if(judgejuzhen.zuobiao[x1][(y1+y2)/2] != 0)//如果蹙马脚
        return false;
    else
        return true;
}
else//如果不是日字
    return false;
}
//炮的规则
public boolean PAO(int x1, int y1, int x2, int y2) {
    if(judgejuzhen.zuobiao[x2][y2]>0)//如果终点是自家的子
        return false;
    else if(judgejuzhen.zuobiao[x2][y2] == 0)//如果终点没有子
    {
        if(y1 == y2)//如果是竖着走的
        {
            int Mx = x2>x1?x2:x1;
            int mx = x2>x1?x1:x2;          //记录大小
            for(int i = mx + 1; i < Mx; i++) {
                if(judgejuzhen.zuobiao[i][y1] != 0)//如果有子
                    return false;
            }
        }
        else if(x1 == x2) //如果是横着走的
        {
            int My = y2>y1?y2:y1;
            int my = y2>y1?y1:y2;          //记录大小
            for(int i = my + 1; i < My; i++) {

```

```

        if(judgejuzhen.zuobiao[x1][i] != 0)//如果有子
            return false;
    }
}
else//如果是斜着走的
    return false;
}
else//如果终点有敌方的子
{
    if(y1 == y2)//如果是竖着走的
    {
        int count = 0;
        int Mx = x2>x1?x2:x1;
        int mx = x2>x1?x1:x2;          //记录大小
        for(int i = mx + 1; i < Mx; i++) {
            if(judgejuzhen.zuobiao[i][y1] != 0)//如果有子
                count++;
        }

        if(count != 1)//如果中间不是一个子
            return false;
        else
            return true;
    }
    else if(x1 == x2) //如果是横着走的
    {
        int count = 0;
        int My = y2>y1?y2:y1;
        int my = y2>y1?y1:y2;          //记录大小
        for(int i = my + 1; i < My; i++) {
            if(judgejuzhen.zuobiao[x1][i] != 0)//如果有子
                count++;
        }
        if(count != 1)//如果中间不是一个子
            return false;
        else
            return true;
    }
    else//如果是斜着走的
        return false;
}
return true;
}
//车的规则
public boolean JU(int x1, int y1, int x2, int y2) {

```

```

if(judgejuzhen.zuobiao[x2][y2]>0)//如果终点是自家的子
    return false;
if(y1 == y2)//如果是竖着走的
{
    int Mx = x2>x1?x2:x1;
    int mx = x2>x1?x1:x2;          //记录大小
    for(int i = mx + 1; i < Mx; i++) {
        if(judgejuzhen.zuobiao[i][y1] != 0)//如果有子
            return false;
    }
}
else if(x1 == x2) //如果是横着走的
{
    int My = y2>y1?y2:y1;
    int my = y2>y1?y1:y2;          //记录大小
    for(int i = my + 1; i < My; i++) {
        if(judgejuzhen.zuobiao[x1][i] != 0)//如果有子
            return false;
    }
}
else//如果是斜着走的
    return false;
return true;
}
//王的规则
public boolean WANG(int x1, int y1, int x2, int y2) {
    if(judgejuzhen.zuobiao[x2][y2]>0)//如果终点是自家的子
        return false;
    if(y2 == this.dfy)//如果将帅在同一竖线上
    {
        int count = 0;
        for(int i = this.dfx + 1; i < x2; i++) {          //原有逻辑问题，已修改
            if(judgejuzhen.zuobiao[i][y2] != 0)
                count++;
        }
        if(count == 0)//如果中间没有子，将帅直接见面
            return false;
    }
    if(y1 == y2)//如果走的是竖线
        if(x2-x1!=1 && x2-x1!= -1)//如果上下不是一格(原有逻辑问题，已修改)-
            return false;
    //由于棋盘上没有坐标为 0 和负数的，所以不会出现到头还能选择往前走的，故不考虑该情况
    else
        return true;
    else if(x1 == x2)//走的是横线

```

```
        if(y2 - y1 != 1 && y2 - y1 != -1)//如果不是左右横移一格
            return false;
        else
            return true;
    else//斜线
        return false;
}
```

四、课程设计总结

4.1 总结及体会

课设一直是我认为大学生最该认真做的一门课。如何在短短的一周时间里面，在现成的知识体系上学习新的知识来完成一个新的目标是对代码水平能够有显著提高的。在这一周准备期末考的同时利用更少的时间做出更高质量的东西更可以说是难上加难。虽然最后做出来的成品距离自己嘴还是的预期还是有比较大的差距。但是结果也还是算是让人满意。并且也较好地暴露出了代码编写能力磺酸钠搞的不足：比如说，当代码写到 1600 行以上时，再加上旁边客户端端口也要同时编辑，导致逻辑思考容易出现纰漏。对于上了一定行数的代码在自己脑中的统筹调配的逻辑框架也不够清晰。但是通过这几天的课设作业，我确实地感受到了自己的提高，学习到了不少新的知识，也巩固完善了旧的知识体系。

4.2 难题及解决方案

(1) 难点一是在于 graphics 的运用上，之前没有尝试过用这类画图，第一次尝试时这几个类的知识学了很久才大概弄懂了原理，慢慢上手了之后才画好。

(2) 分层容器即使分了层也不一定能看出来，需要设置 Opaque，这一点在不知情的情况下检查了很久才发现。

(3) 象棋的坐标与二维数组的行列是相反的，没有注意到这一点导致三百多行的规则代码全部出错最后一点一点改回来的。

(4) java 的 swingUI 线程与 socket 线程是互斥阻塞的，所以不能够线程混写，需要单独把接收数据写成一个死循环放在主线程里方可读取数据。

参考文献

[1]王素琴. Java 语言程序设计实用教程 [M]. 北京: 中国电力出版社, 2017.2. 1-276.

五. 意见与建议

平时作业过于简单，在实际课设的时候有许许多多的问题存在，都是在课堂上没有遇到过的，作业里没有过的。导致花费了大量的时间在检查一些很小却完全没有了解到过的问题上。希望平时的时候可以增加作业难度，多多拓展课外知识，而不是作业仅仅是把书上代码敲一遍。

本次由于学期较短课设与期末考冲突，回头需要麻烦各位任课老师合理沟通，不然确实太过吃力。