**程序设计思维大作业**

**姓名：张浩威**

**授课老师：邓习峰**

**目录**

[1 大作业选题设计 2](#_Toc516170138)

[2 简要项目说明 2](#_Toc516170139)

[3 项目分析 2](#_Toc516170140)

[4 重点难点分析 3](#_Toc516170141)

[4.1 重点 3](#_Toc516170142)

[4.1.1 五子棋棋盘布局及判断胜负 3](#_Toc516170143)

[4.1.2 界面布局 4](#_Toc516170144)

[4.2 难点 5](#_Toc516170145)

[4.2.1 客户端与服务器之间的交互 5](#_Toc516170146)

[5 操作流程及代码分析 6](#_Toc516170147)

[6 总结与展望 11](#_Toc516170148)

# 大作业选题设计

五子棋是一款老少皆宜的休闲活动！还记得从小学一年级开始，我就学会了下五子棋，并且常和父母亲戚下五子棋。由于这款棋类非常简单，极易上手，并且短时间内能够下完，常成为上学期间下课时风靡班级的娱乐活动之一。

五子棋这款游戏，在十五年之前也早已出现在互联网的各个角落，小时候我经常会思考，是什么让我们双方在两个可能相差几千里的地方，能够共享同一块棋盘。现在，我知道了，是服务器！

于是趁着这次大作业的机会，我很渴望能自己实现一次五子棋。感谢这次大作业，逼着我学习客户端与服务器之间的交互模式，并最终通过30KB的代码完成了这个互联网对战版五子棋。

# 简要项目说明

名字：五子棋局域网对战版

创作者：张浩威

游戏简介：这款游戏由两名玩家组成。一开始有一个15\*15的棋盘，先手执黑，后手执白，在棋盘下轮流放下自己的棋子。当有一方率先出现连续五颗自己的棋子形成一条竖、横或者正斜时获得胜利。

# 项目分析

首先我们得先确定这个项目的流程。

初始时应当有一个登录界面，因为这是一个互联网对战，如果不通过用户名，我们只能通过ip地址来获取对手的信息，这件事显然不靠谱。为了做这个用户系统，我们需要在服务器端使用数据库来维护所有用户的信息。

这个登录界面需要有两个功能：登录和注册。

首先注册需要输入用户名与密码，此时在后端我们需要检验是否已存在用户名，并且将注册成功或者用户名已重复的信息告知给客户端。另外登录时我们需要核对是否存在该用户，密码是否正确等，并告知用户相应的信息。

一旦登录成功，我们需要进入邀请界面。在这个界面中，我们可以查看到已登录的玩家的情况，并邀请它们进入对战。此时会出现三种情况：被邀请，对方同意，对方拒绝。针对这三种情况，服务器都需要分别告知给客户端，并且客户端根据每种情况来执行相应的策略。

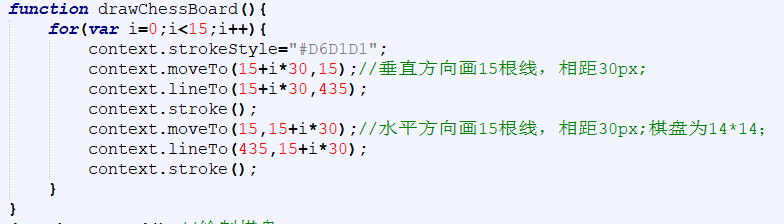
之后一旦被邀请后同意，或者对方同意自己的邀请，将会进入第三个界面：对战界面。在这个界面中，双方轮流下棋，我们需要告知给客户端他所执的棋子的颜色，以及当前回合。这里由于两个客户端之间是不能互相通信的。每次客户端需要告知给服务器它的下棋位置，以及每次服务器需要告知给客户端上一步对方下在了哪里。在这一界面中，我们还需要维护当前棋盘的情况，如果出现连续的五颗颜色相同的棋子，则认为游戏结束。这里我们采用在客户端中判断的方法。（事实上我们也可以在服务器里判断，但这边我的实现是在客户端中操作的）另外为了让用户体验更加友好，还需要加入“投降”功能。

以上便是整个项目的流程，除此之外，需尽可能通过css与jquery设计出美观的画面。

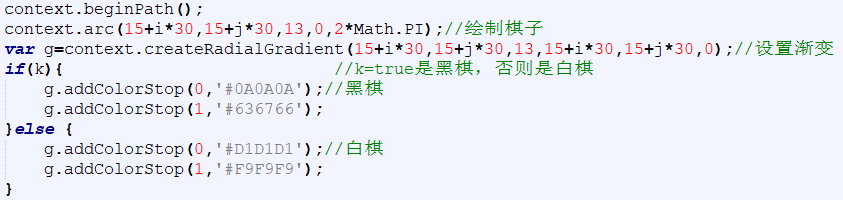
# 重点难点分析

## 重点

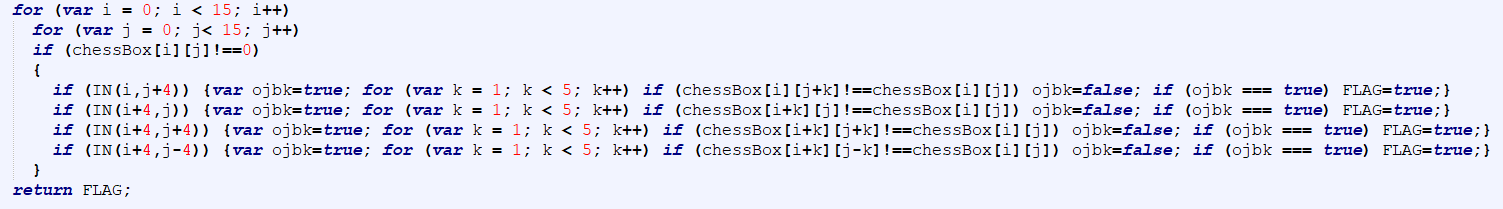
### 五子棋棋盘布局及判断胜负

我们采用canvas来布置我们的五子棋局，首先我们需要制作15条间距相同的横线与竖线来绘制这个棋盘。这里我们用如下代码来实现，其中context是基于canvas来实现的：

另外我们需要制作一些棋子，棋子显然是正圆形的，以及颜色要么纯黑要么纯白。我们采用如下代码来实现：



在双方下棋的过程中，每下一步我们都需要判断游戏是否终止，这里终止条件非常显然。判断终止的方法我们可以通过枚举棋子的起点，以及连续接下来连续4个棋子是否和当前棋子颜色一样来判断。代码如下：



这里IN表示是否落在棋盘内，chessBox存的是棋子的颜色，0表示没棋子。

另外，由于本游戏有投降功能，我们可以在程序中规定一旦下出了边界则视为投降。也就是在服务器和客户端通信时，一旦客户端收到对方落在了棋牌外的信号，则视为投降。

### 界面布局

如果我们不断地通过链接来修改页面，虽然也能达到效果，但运行时间及用户体验上肯定是非常差的。经过不断尝试，我最终将页面设定为两个：初始界面与对战界面。

初始界面包括登录界面以及用户邀请界面。一开始隐藏用户邀请界面，一旦登录则隐藏登录界面。

这一点在处理的过程中是通过js和css来做到的。通过js调整在css中的显示状态就可以做到这一点。另外我还加入了逐渐出现等动画效果，能更加符合现代人的审美。

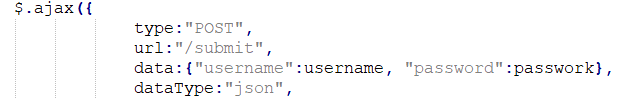
## 难点

### 客户端与服务器之间的交互

一开始我一直在思考客户端之间如何进行交互。直到后来我才发现这个出发点是有问题的。所有客户端都只连接着服务器。也就是说只有服务器能与客户端进行通信。客户端之间没有办法来传递信息，但可以先传递到服务器，再由服务器传递到相应的客户端中去。

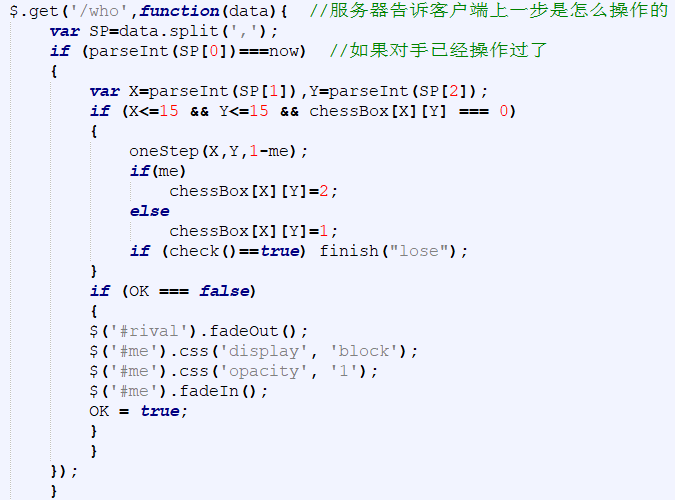
这里我用ajax来实现它们的传递过程。

其中客户端到服务器是通过POST来传送的，以登录界面举例，代码如下：



其中username和passwork表示用户和密码。（因为password不让用）

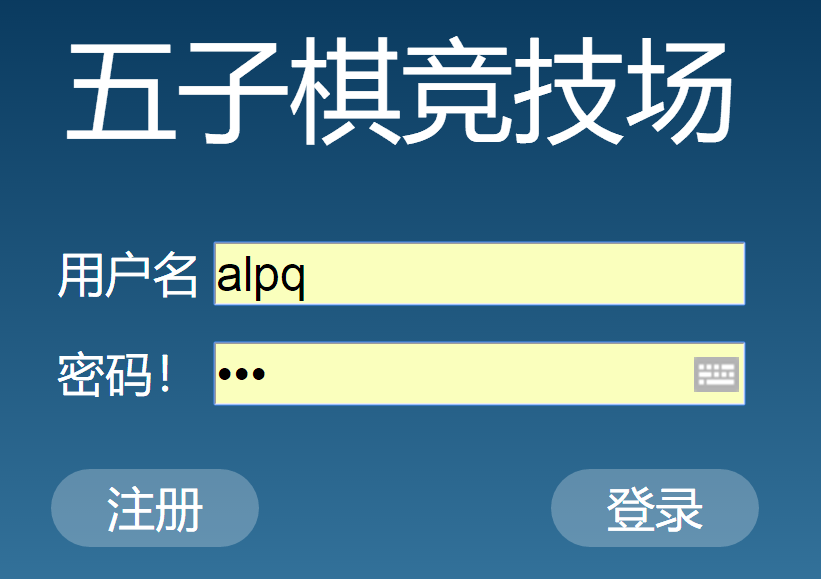
从服务器到客户端是通过get来传递的，以在对战中接受上一步对手的操作为例，代码如下：



# 操作流程及代码分析

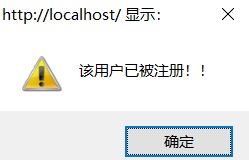
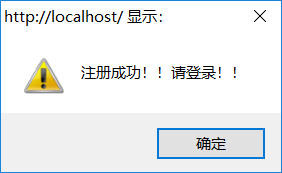
由于调字体调格式调动画都是些工程活，我们略去那些代码分析，只分析一些交互以及有技术含量的代码。

在一开始登录时会出现如下界面：



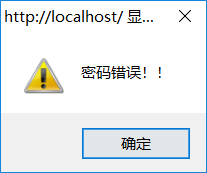
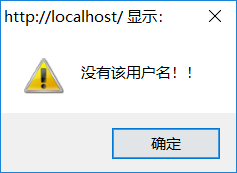
注意这里都是以动画形式出现的，这里我只截取了最终的样式。

点击注册会出现两种情况：



都是通过后端返回给前端的信息得到的。

登录有三种情况：

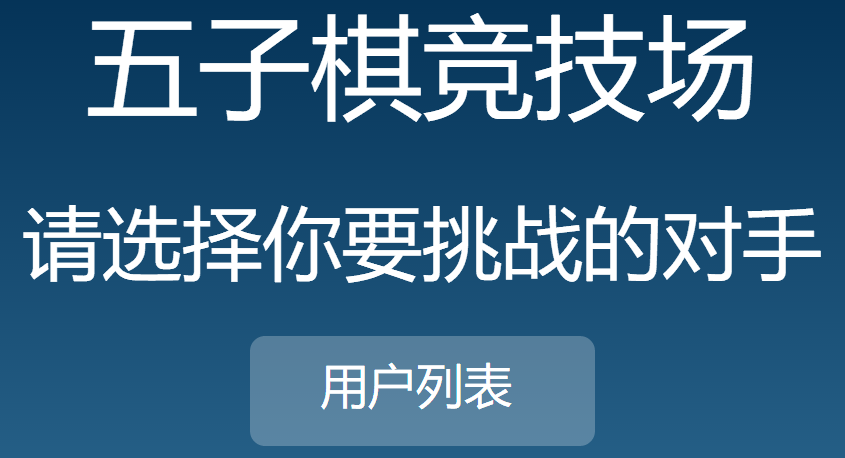
以及登录成功跳转下一界面。实现原理同上。

代码实现如下（以注册为例）：



首先读取客户端发送的username和password，检查数据库中的信息，并返回相应的两个操作。

登录成功后的界面如下：



当出现其余用户时，则会以动画形式出现在用户列表中出现：

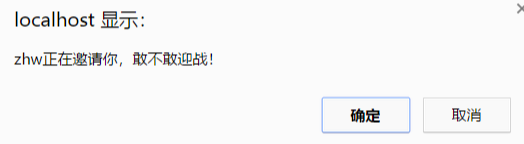
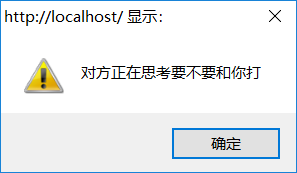


并列出了ID和邀请按钮。

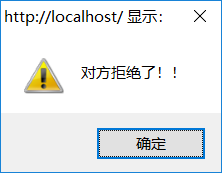
注意这里的邀请按钮和之前的注册和登录在鼠标移动到它时都有边框高亮效果，如下：



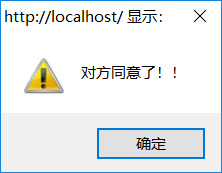
点击邀请功能，两名用户会分别弹出：



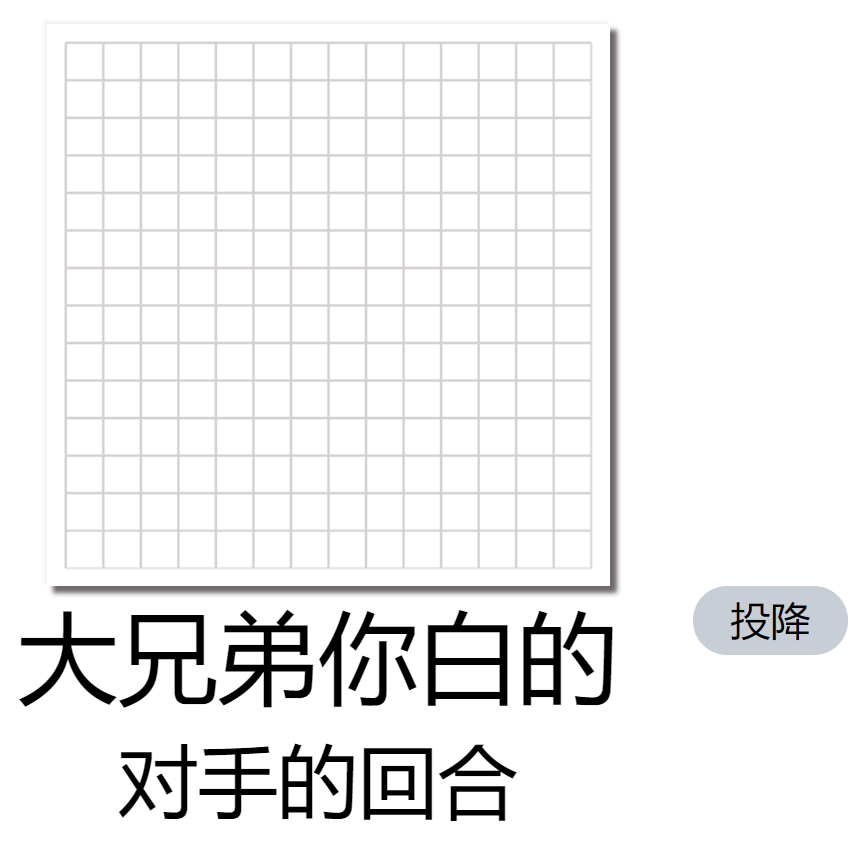
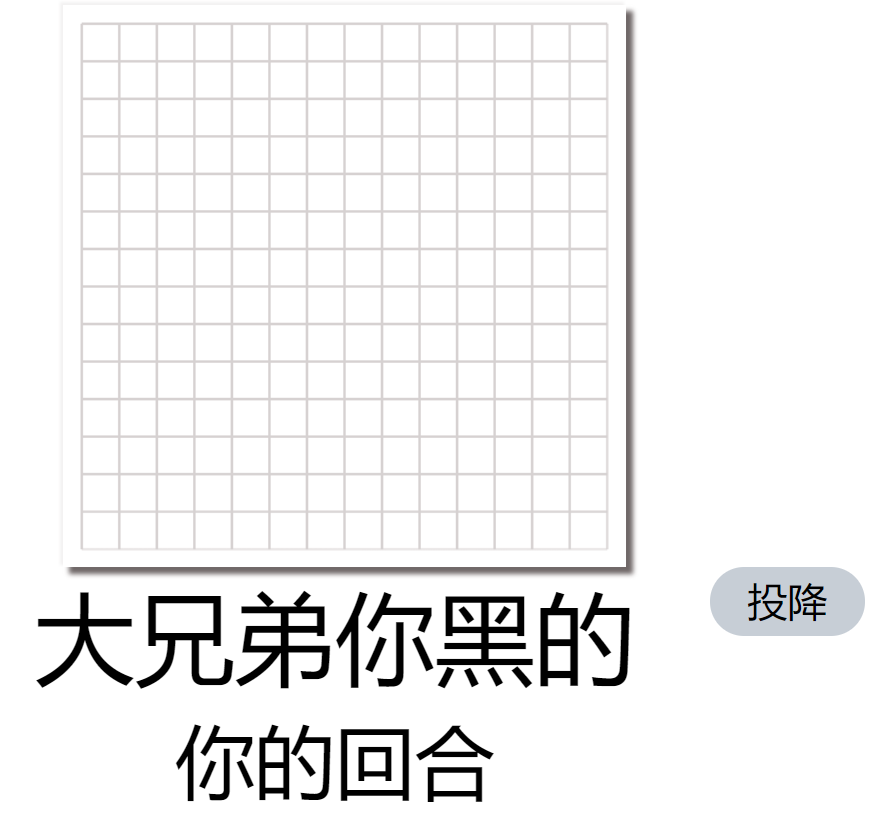
如果点取消，则会弹出：



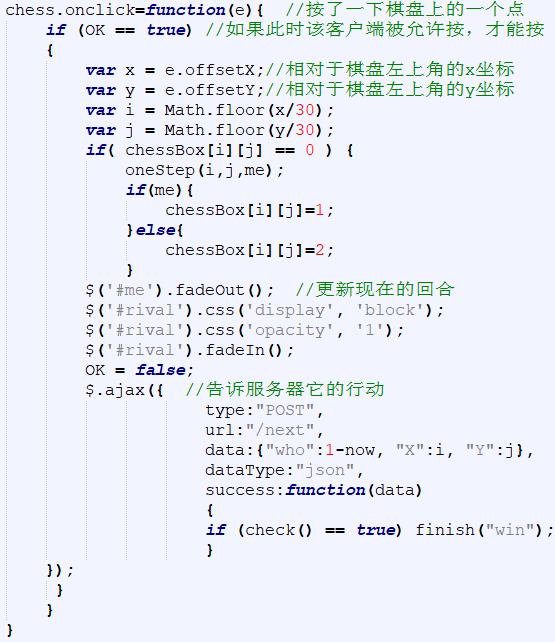
否则会弹出：



之后双方进入对战界面：

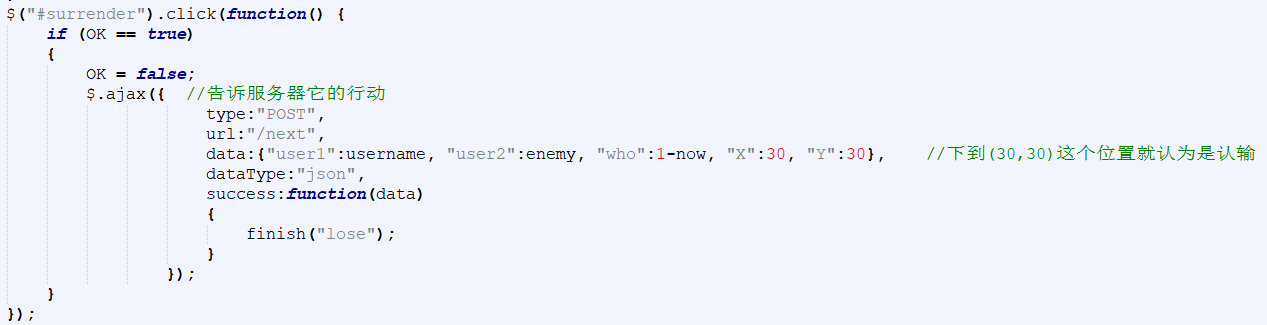


通过以下代码控制一方无法点击屏幕以及与服务器交互：



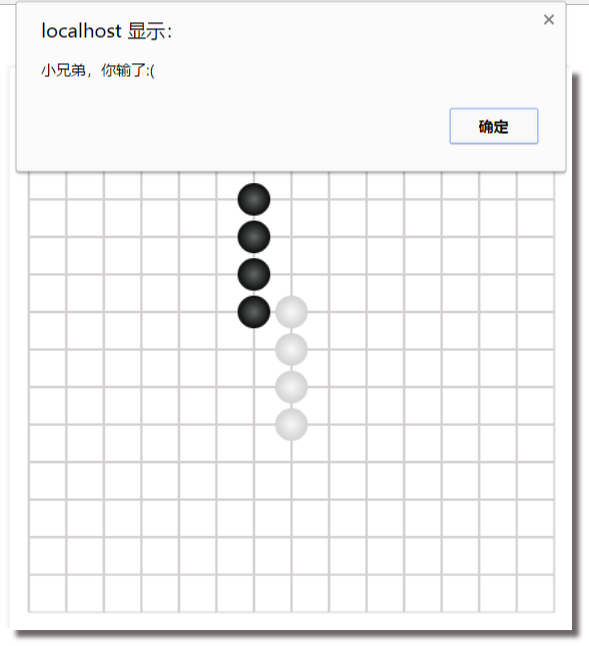
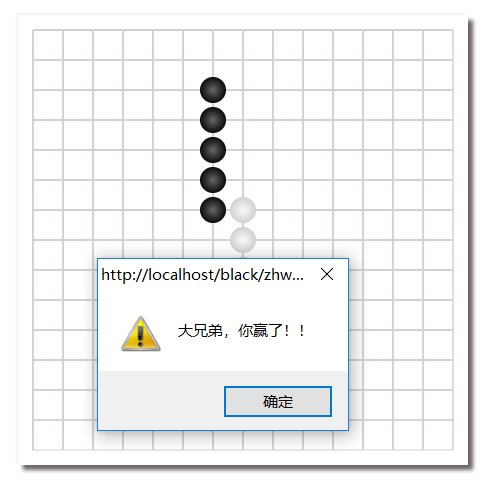
显然当OK为false时，点击屏幕并没有什么用。通过计算得到那个格子，如果格子是空的才采取行动，并告知服务器这个行动。若check为true也就是说游戏已经结束了，则表示胜利。

投降的按钮设置如下：



也就是说，一旦点击这个按钮，就会向服务器发送落在棋盘外的信息。

双方会不断的地下棋直到一方获得胜利或者一方投降为止，以正常胜利为例，会有如下提示：



点击确定后再次回到初始界面。选手可以继续登录下棋。

为了让老师方便的体验这个工程得到的效果，我制作了三个不同的demo：注册登录，双人对战和多人对战。

这三个demo几乎囊括了所有能用到的功能。

在多人对战中采取的是4个客户端，在两幅棋盘上进行对战并且互不干扰，这是通过在html以对战双方名写入链接来实现的。

# 总结与展望

在程序设计思维这门课上，我们简略学习了js，flask，css，html等，但都是非常粗略的学习。在这个工程上，是将js,flask,css和html融为一体，实现出了一个美观且能正常工作的网络对战版五子棋。这件事之前在我看来是一个遥不可及的事情。没有这次大作业，或许我将永远不会掌握这些。在做这个大作业的过程中，都是不断地去思考，探索，询问，更多的是去查询并调用一个从没见过的函数。在了解js如何工作以及前后端如何交互时，甚至一度兴奋到不想吃饭睡觉，只想赶紧写代码赶紧调bug，实现一个可视化工程给我带来的快乐是毋庸置疑的。

事实上现在的版本只是粗略的实现了五子棋的正常运行，但还不足以推广并使用。由于服务器向客户端发送信息是客户端不断请求的结果，因此当客户端一多，服务器的压力会变得非常大。在demo展示中，4个客户端服务器就已经显得有点卡顿了。因此要想克服这个问题，只能利用html5让服务器告诉客户端的方式来实现。

另外，网上对战版的五子棋具有很多功能，例如设置时间，设置是否有禁手，求和，悔棋，判断是否已经无解等。这些功能都是可以逐一添加进去的，由于学业压力较重，在期末季就不能一一实现了，希望能得到老师的谅解！

最后，希望能看到这款网络对战版五子棋真正投入使用的那一天！

最后的最后，为了方便大家学习与交流，我已将我的代码开源，开源网址：

https://github.com/alpq654321/chess