

# 软件架构课程设计



## 旋转五子棋

### 第二组

姓名： 陈明哲 学号： 41512154 班级： 2015 级软工 1 班

姓名： 张海浩 学号： 41512151 班级： 2015 级软工 1 班

姓名： 谢 敏 学号： 41512157 班级： 2015 级软工 1 班

姓名： 刘奇灿 学号： 41512153 班级： 2015 级软工 1 班

姓名： 杨 美 学号： 41512178 班级： 2015 级软工 1 班

# 目 录

一、选题 .....	3
二、需求分析 .....	4
三、系统功能架构图.....	5
四、团队界面 .....	7
五、对程序的改进 .....	10
六、代码质量审查机制与质量属性分析 .....	14
七、设计模式 .....	18
八、软件评估 .....	18
九、技术说明 .....	21
十、难点攻关 .....	22
十一、软件使用补充说明 .....	23

## 一、选题

旋转五子棋是一款以五子棋为基础的益智游戏，不同于传统五子棋，旋转五子棋将一个 6\*6 的棋盘分为了 4 个 3\*3 的棋盘，对战双方每次落子后，都可以而且必须在四个区域中挑选一个，将其进行顺时针或逆时针的 90 度旋转。这就为棋局的变化带来了更多的可能，在落子时需要进行的计算也更加复杂。

本游戏分为三个游戏模式：人机对战，双人模式和线上对战。

1. 人机对战：人机对战分为三种 AI 模式（简易、中等、较难），玩家可以自由选择，而且可以选择自己执黑子还是白子（均为黑子先下）；
2. 双人模式：用一个 Android 手机双方依次下棋，适合于两人近距离休闲娱乐进行游戏；
3. 线上对战：两人用两个 Android 手机通过房间号进行配对进行游戏，随时随地只要有网络就可以进行线上游戏，灵活方便。

游戏的基本形式借鉴与 ios 系统上的同类小游戏，但 ios 上的小游戏只有线下双人游戏模式和简单人机对战模式，而我们的这个 Android 版本不仅拓展的人机对战的难度（采用自己设计的 AI 算法），也进行了网络线上对战模式的开发，采用云服务器进行线上支持，利用我们所学知识和自学的网络传输，利用可靠的 TCP 点对点会话，完成了线上对战的设计。

本次开发周期受限于比赛的时长限制，只有短短的二十余天，所以还有一些功能没有达到我们预期设计的效果，比如加上线上登录模块记录用户比赛信息，加入线上对战聊天模块，统计用户有效的赢棋策略，优化人机对战的 AI 算法，采用深度学习来训练更好的对战逻辑，优化服务器的吞吐量，提高服务器并发处理能力等。

## 二、需求分析

### 涉众

用户、客户：面向大众

架构师：陈明哲

开发、集成、测试：陈明哲、张海浩、谢敏、刘奇灿、杨美

维护：陈明哲、张海浩

### 目标：

实现一个旋转五子棋对战程序，供搜索算法研究

软件开发时间,初期版本 1 个月,实现基于网络的双人对战、初级人机对战

### 实现双人对战

黑白方

6×6 棋盘

规则

黑先行

某方向上五连则胜

### 实现人机对战

在人机对战的基础上，可实现 Android 客户端执黑、执白对战

### 多样性的对战方式

远程对战

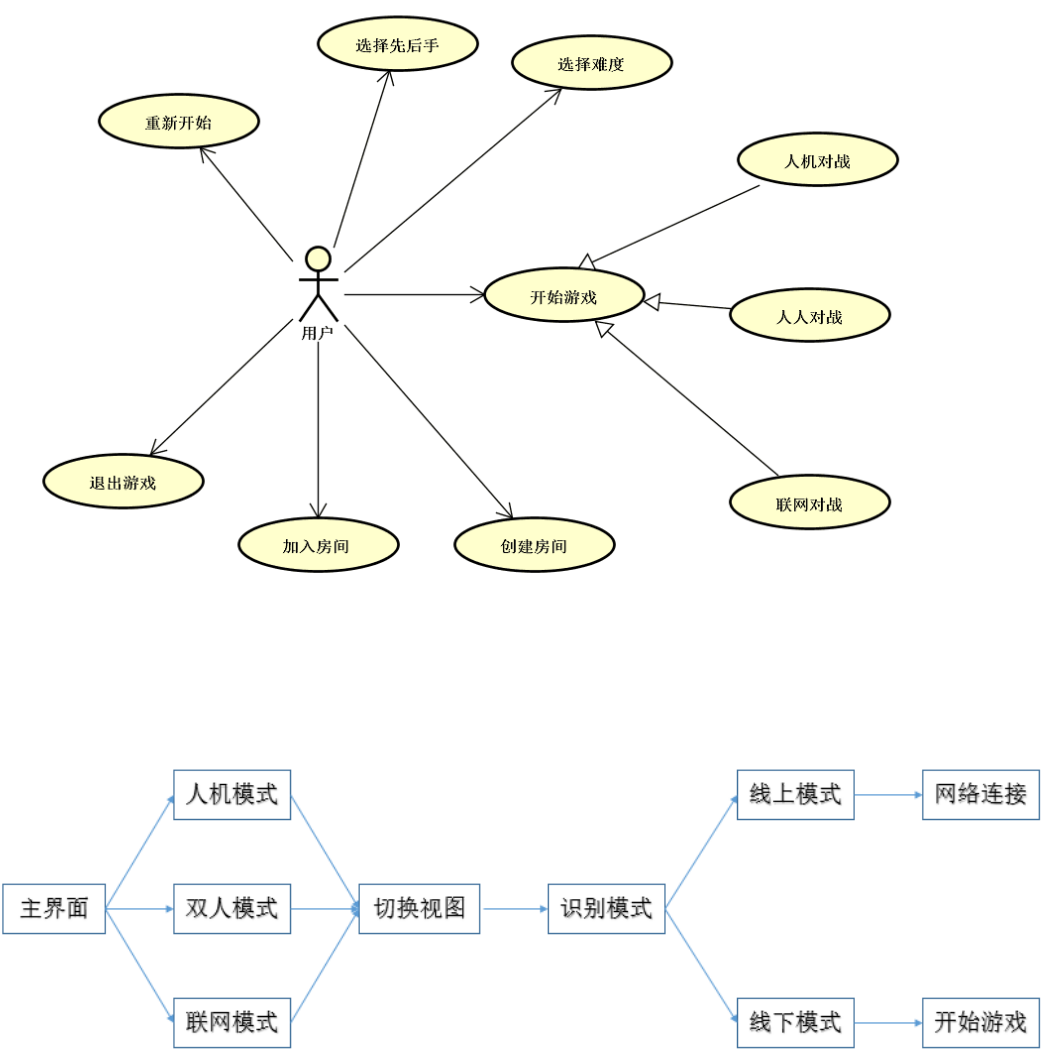
远程登陆、身份认证

对战聊天功能

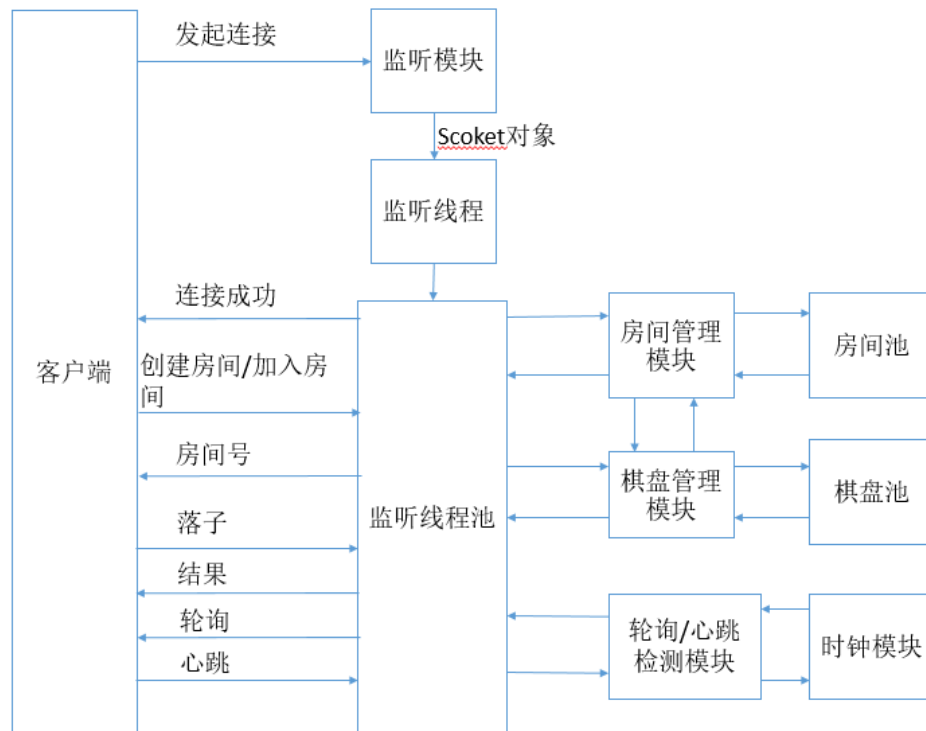
对战计时功能

程序可能要移植到新的软、硬件环境（IOS 系统）

三、系统功能架构图

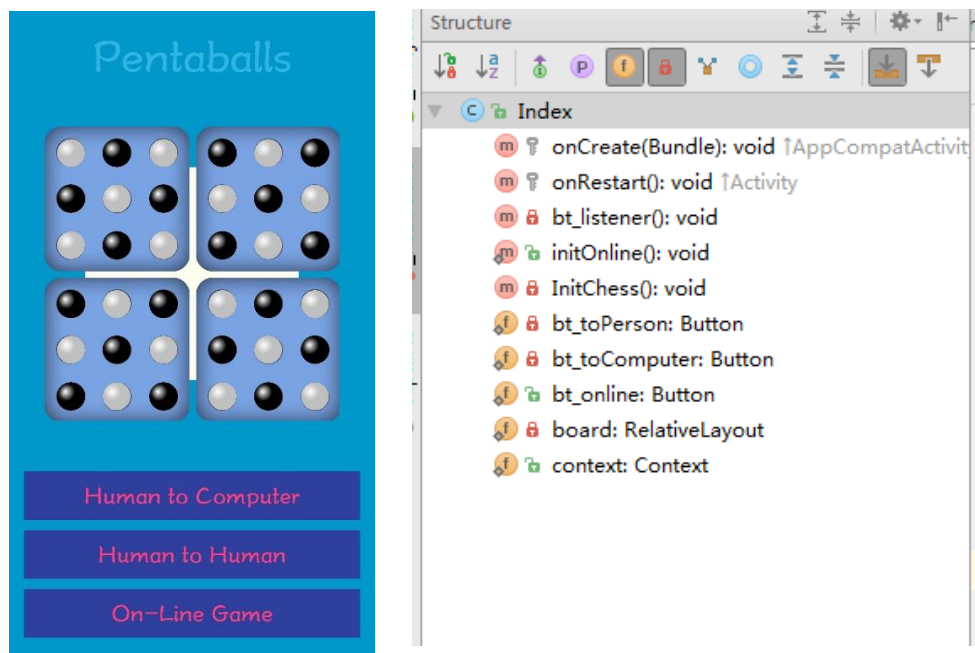




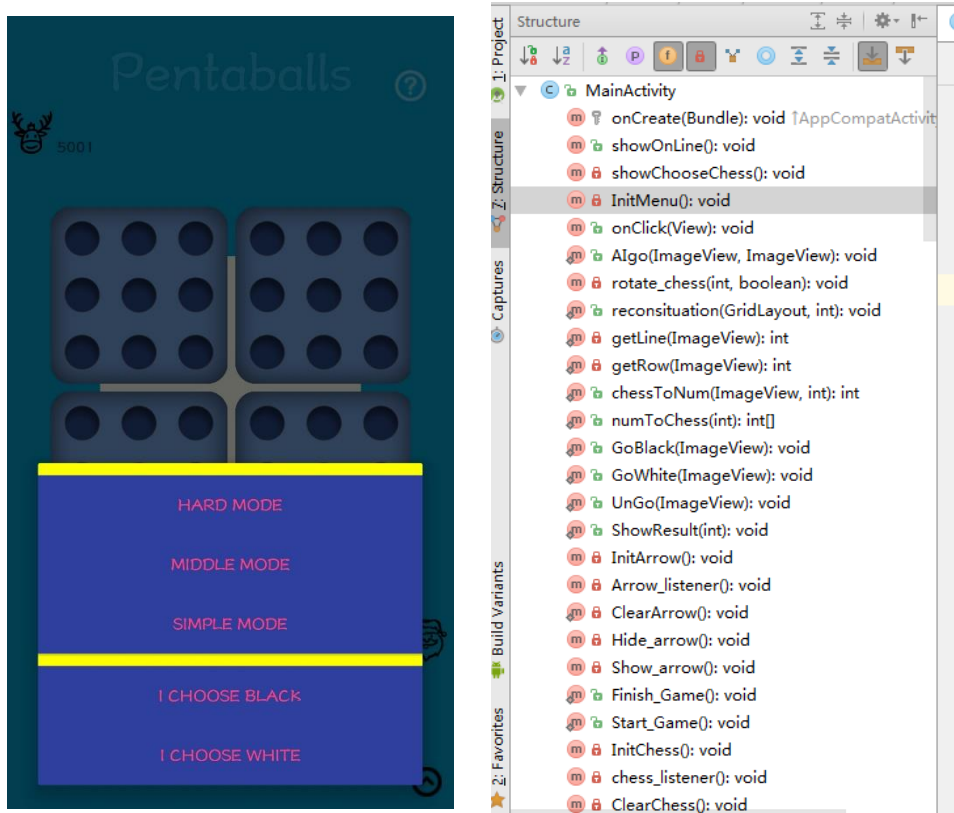


#### 四、团队界面

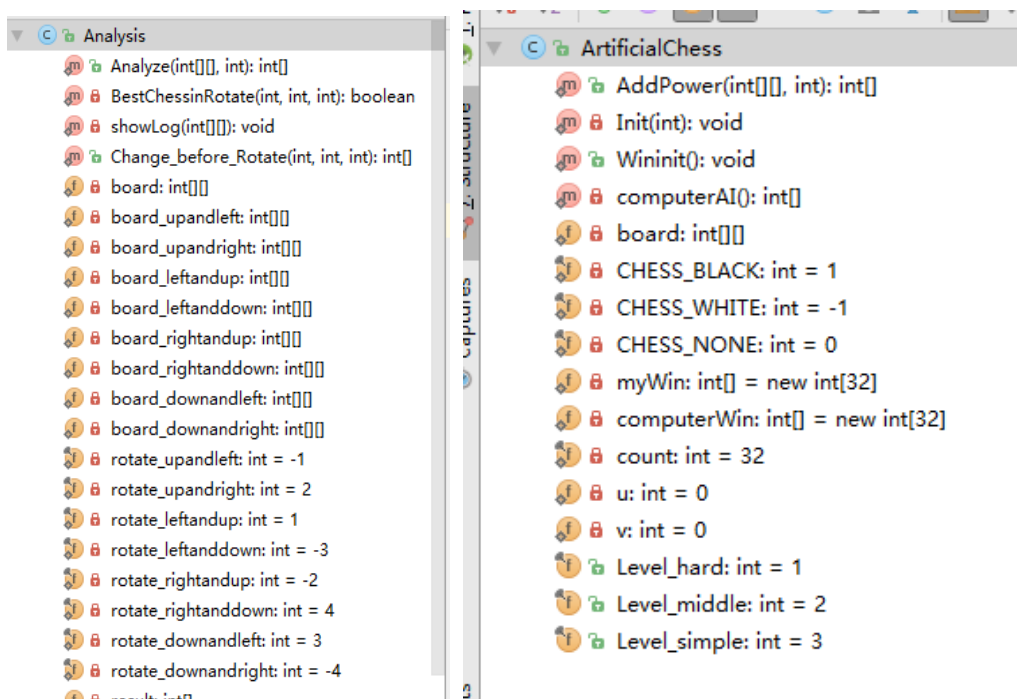
团队分工明确，包括客户端的设计编码、服务端的设计构建和软件测试等等。



首页三个按钮分别代表三种模式：人机对战，线下双人模式，线上对战。



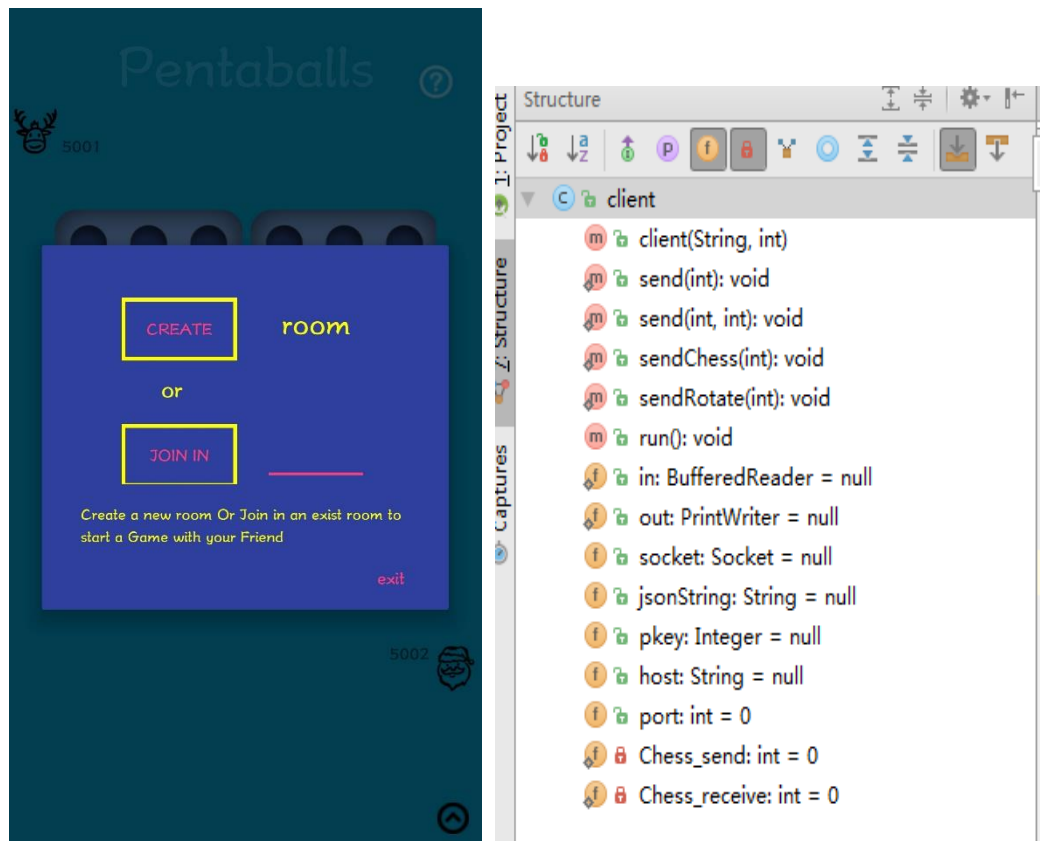
模块高度内聚，函数功能设计明确，函数间调用逻辑清楚。



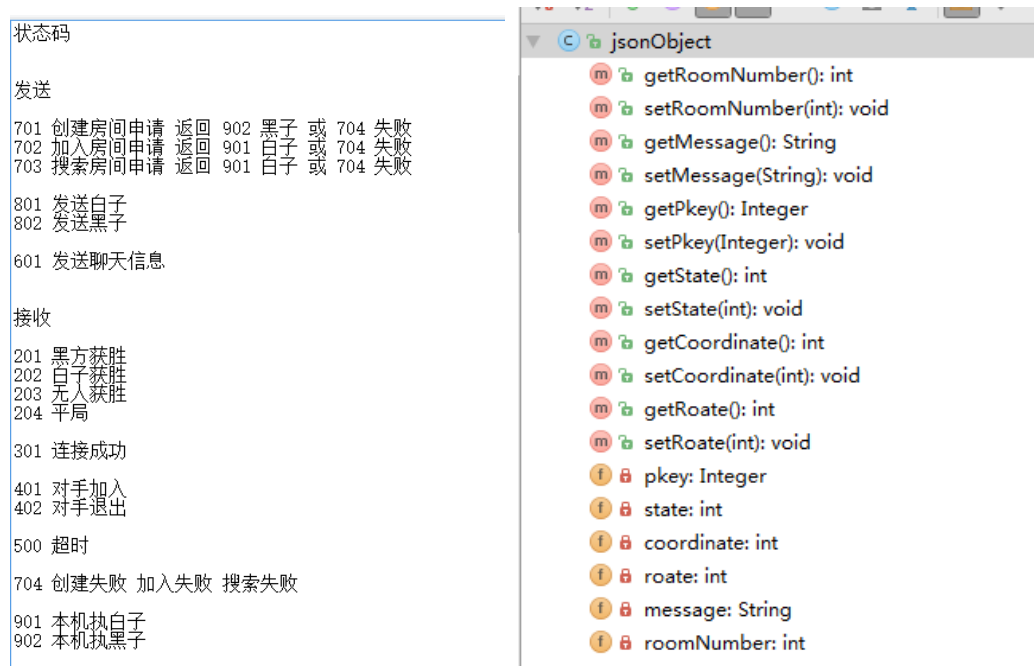
人机对战 AI 算法设计，高内聚，低耦合，对棋盘状态判断效率较高，根据不同的难度



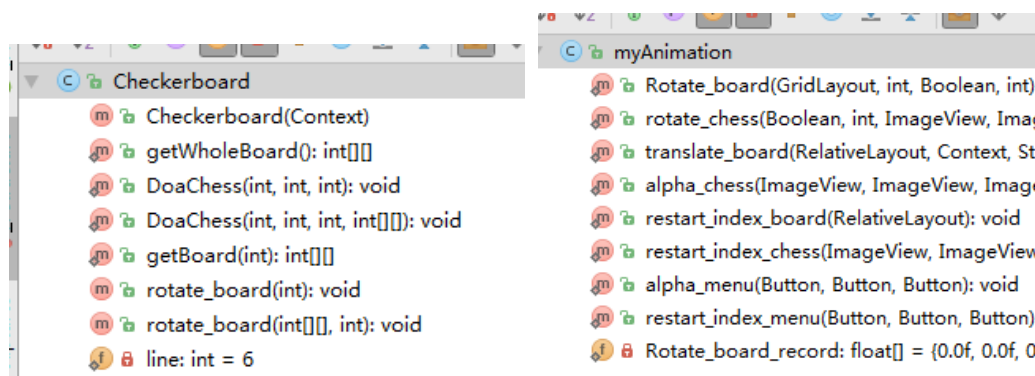
分配不同的权值，采取有效的防守和进攻策略，高难度模式算法还需要继续优化，高难度现行算法在运算过程中可能出现偏差导致防守策略失败，后续我们还会继续将这个应用继续设计，争取达到预期效果。



在线对战模式设计思路清晰，配上部分功能分布式处理的云服务器作为平台，尤其是对房间的申请和释放处理做了很多工作防止线程资源浪费，，为后续功能预留多种接口，可重用设计也为服务器的优化提供了有效的保证。

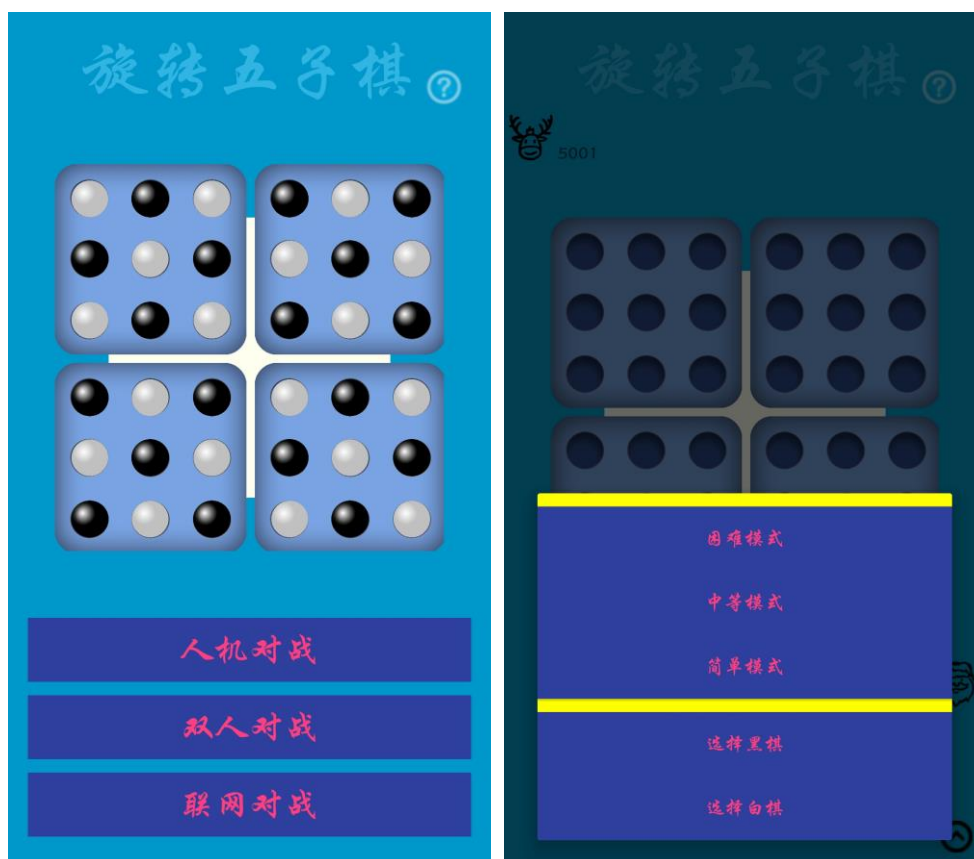


采用 json 作为通信介质，设计了一系列的通信状态码，以请求服务器对客户端操作做出相应功能的反馈。

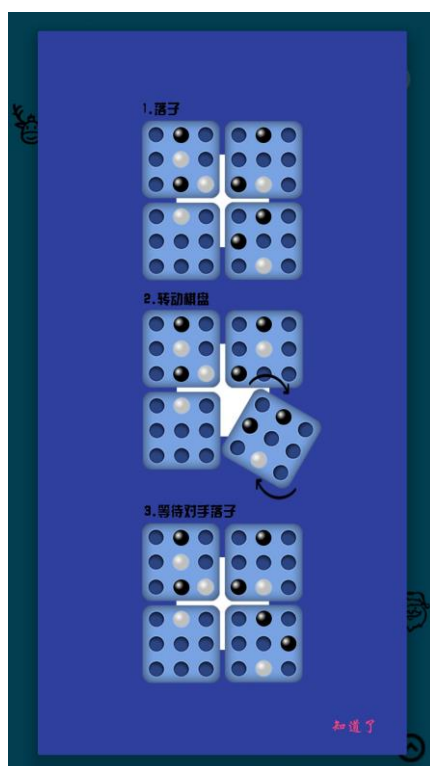


## 五、对程序的改进

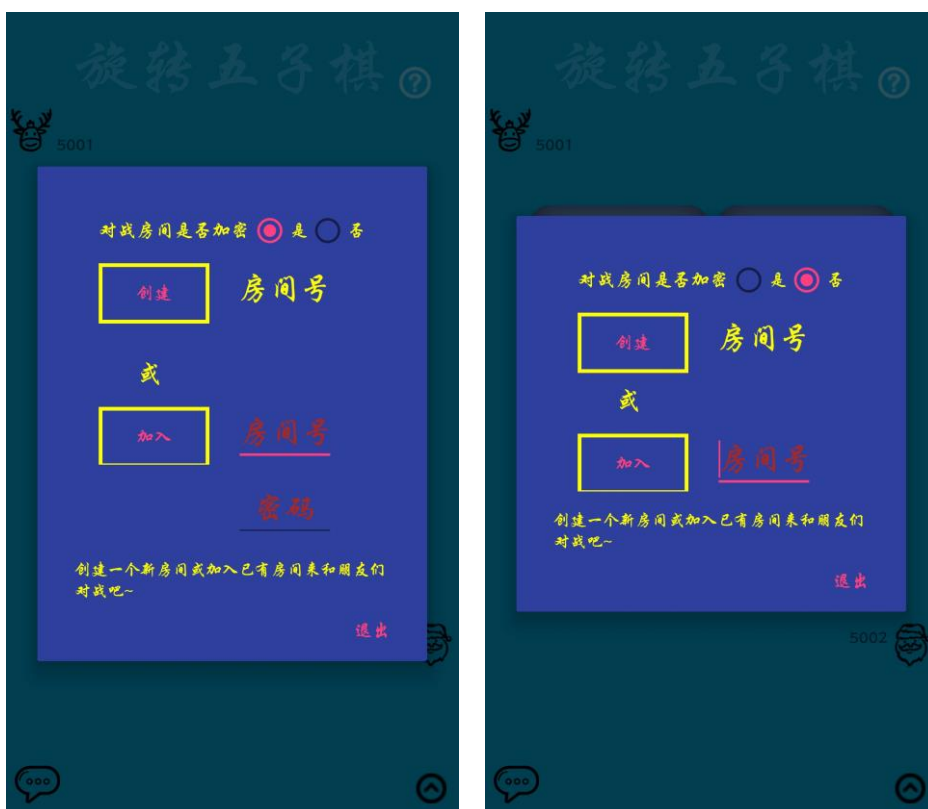
- 1、首先是对中文版的拓展：在收集一些用户体验之后，有些反映刚开始对规则的不清楚，再加上英文版的限制，导致无法快速上手，所以我们扩展了中英两种语言支持；



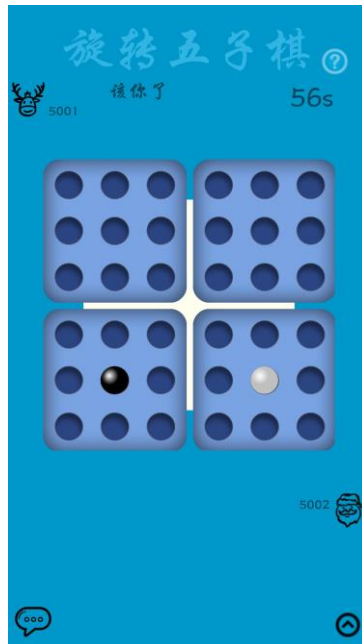
2、对教程的绘制：为了让游戏新手尽快的了解规则一遍更加便捷的操纵，我们整理并且制作了规则教程以方便用户学习查看；



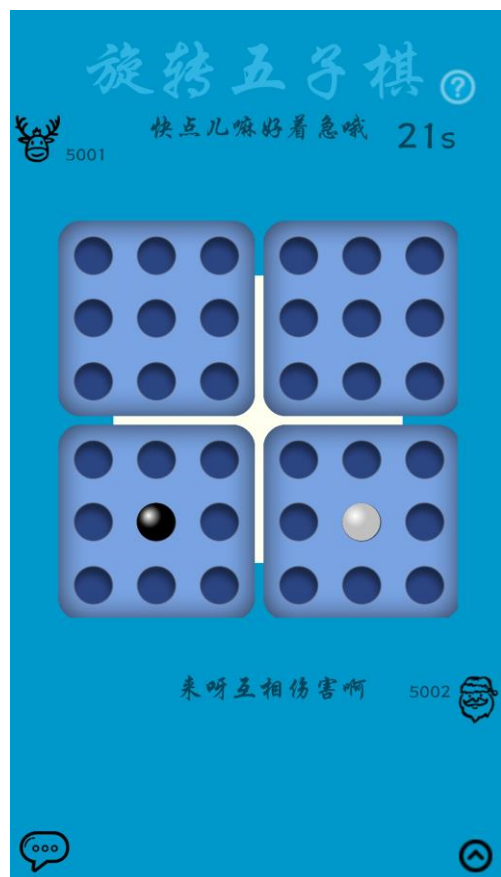
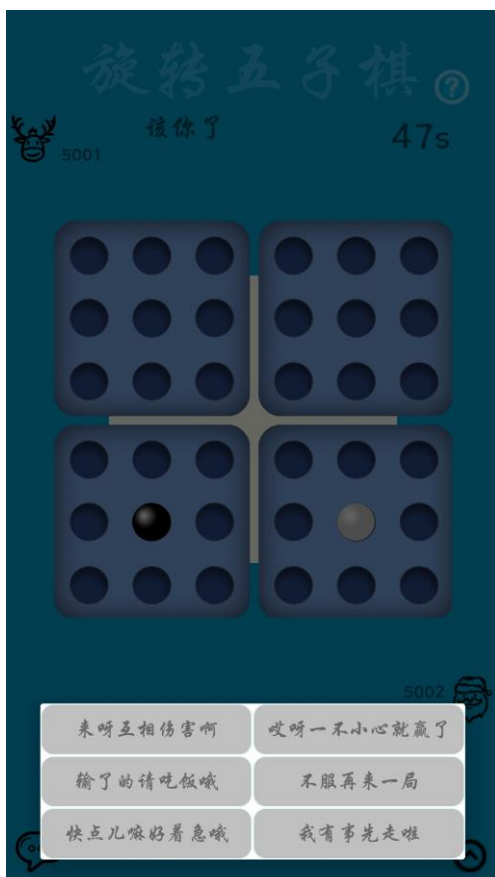
- 3、对线上连接质量做了极大的改善，之前经常超时和掉线，现在与服务器搭设了双向心跳检测，很大程度上减少断线的情况；
- 4、增加了加密对战的房间模式，两个人对战必须输入正确的房间号码和密码，才能连接对战，增加了对战的多样性，避免了房间加错的情况；



- 5、增加了对战的时间限制，增强了竞争性，增加了游戏的紧密感，以一分钟为限制，让玩家合理的掌握游戏节奏；



6、引入在线对战聊天功能，发送一些常见的会话，增强对战的趣味性。



## 六、代码质量审查机制与质量属性分析

**代码审查：**小组成员交流密切，共同讨论功能模块的搭建构架及接口设计，对其中的逻辑测试也十分重视，用大量注释标记处每一个逻辑点的作用以及范围，对每一个类都做了函数功能逻辑设计，每天备份开发过程，总结开发中的创新点，对一些 BUG 进行讨论，利用互联网查找解决方法。

**系统缺陷：**游戏在线模块对 Android 7 以上系统出现不兼容现象，无法通过套接字连接，无法使用在线对战功能，经上网查阅，Android 7.0 对市场上很多应用都出现不兼容情况，其中在应用市场 Top1000 有 166 个应用都对 Android 7.0 不兼容，其中游戏类应用不兼容情况最多。

### 本游戏具有如下特点：

#### 1) 基于 C/S 体系

Client/Server 结构(C/S 结构)是客户机和服务器结构。它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将任务合理分配到 Client 端和 Server 端来实现，降低了系统的通讯开销。

#### 2) 完善的通信机制

对线上连接质量做了极大的改善，之前经常超时和掉线，现在与服务器搭设了双向心跳检测，很大程度上减少断线的情况；增加了加密对战的房间模式，两个人对战必须输入正确的房间号码和密码，才能连接对战，增加了对战的多样性，避免了房间加错的情况；

#### 3) 游戏多方位的模式

本游戏分为三个游戏模式：人机对战，双人模式和线上对战。

人机对战：人机对战分为三种 AI 模式（简易、中等、较难），玩家可以自由选择，而且可以选择自己执黑子还是白子；

双人模式：用一个 Android 手机双方依次下棋，适合于两人近距离休闲娱乐进行游戏；

线上对战：两人用两个 Android 手机通过房间号进行配对进行游戏，随时随地只要有网络就可以进行线上游戏，灵活方便。

#### 4) 个性化操作界面

一个好的游戏不仅仅要体现在强大的功能上，还要在使用上具有方便、快捷、高效的特点。系统简单上手：游戏系统界面简单易懂，容易操作。对于游戏规则介绍详细。一目了然。也引入在线对战聊天功能，发送一些常见的会话，增强对战的趣味性。各项功能安排井井有条。

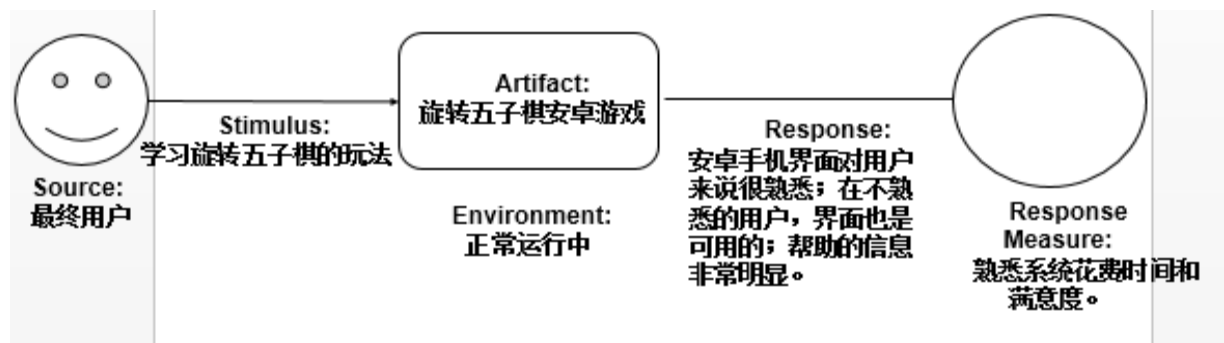
### 质量属性需求：

#### 1 可用性

##### 1.1 原因

可用性关注于如何让用户简单容易的实现他想要完成的动作。这样可以使用户快速地熟悉游戏的每个环节，愉悦地玩旋转五子棋，从而将错误最小化并且让用户对该旋转五子棋满意。

Portion of Scenario	Possible Values
Source	最终用户
Stimulus	学习旋转五子棋的玩法
Artifact	旋转五子棋安卓游戏
Environment	正常运行中
Response	安卓手机界面对用户来说很熟悉；在不熟悉的用户，界面也是可用的；帮助的信息非常明显。
Response Measure	能正常清楚玩游戏所花费的时间，用户的满意度，错误操作的数量

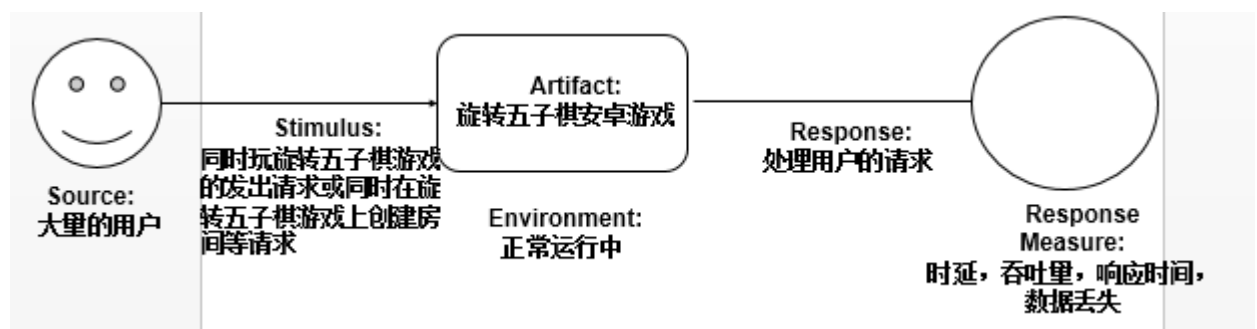


## 2 性能

### 2.1 原因

旋转五子棋游戏必须保证其性能，才能为用户提供正常可靠及时的服务，用户无法或者需要等待很久才能得到系统的服务，这必然会降低用户对该系统的评价，严重影响系统的质量。性能主要关注于响应时间。所以本次游戏开发在在线对战模式设计思路清晰，配上部分功能分布式处理的云服务器作为平台，尤其是对房间的申请和释放处理做了很多工作防止线程资源浪费，服务器端采用 REDIS 数据库对战数据做可靠存储，为后续功能预留多钟接口，可重用设计也为服务器的优化提供了有效的保证。

Portion of Scenario	Possible Values
Source	大量的用户
Stimulus	同时玩旋转五子棋游戏的发出请求或同时在旋转五子棋游戏上创建房间等请求
Artifact	旋转五子棋安卓游戏
Environment	正常运行中
Response	处理用户的请求
Response Measure	时延，吞吐量，响应时间，数据丢失。



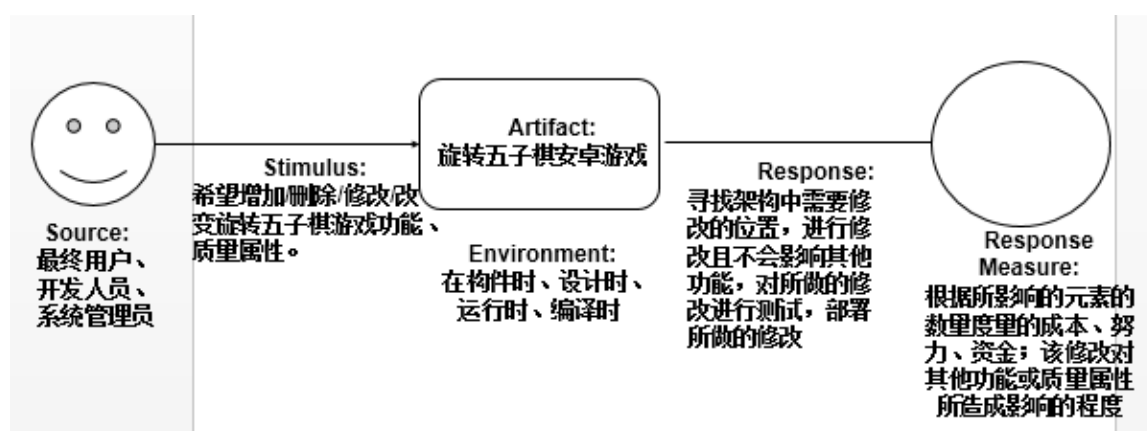


### 3 可修改性

#### 3.1 原因

在开发过程中或在最后交付用户使用 ,可能会根据开发进度和最后用户使用习惯进行修改。而且在旋转五子棋游戏的使用中我们可以增加/删除/修改/改变系统功能使之随着市场的改变更具有可用性。

Portion of Scenario	Possible Values
Source	最终用户、开发人员、系统管理员
Stimulus	希望增加/删除/修改/改变旋转五子棋游戏功能、质量属性。
Artifact	旋转五子棋安卓游戏
Environment	在构件时、设计时、运行时、编译时
Response	寻找架构中需要修改的位置，进行修改且不会影响其他功能，对所做的修改进行测试，部署所做的修改
Response Measure	根据所影响的元素的数量度量的成本、努力、资金；该修改对其他功能或质量属性所造成影响的程度



经过这一个月的发展，作为大二软件工程的我们可是好好实践了一次，在这之前我们对我们所做的也非常不熟悉，这一个月的努力让我们的编码能力得到了很大的提升，对那些核心代码的掌握也比较熟练，虽然说不能做到每个知识点都特别清楚，但解决问题的过程却是让我们学到了很多，也算是一次时间紧任务重的历练吧。在开发过程中，我们也在不断推翻着之前许多个解决问题方法，由只是简单的处理 BUG 到比较全面的

特意设置一些接口来统一的处理一类问题，我们体会到了自顶向下的开发方法，与逐步求精的开发准则，在接下来的学习生活中，我相信我们会更快的进步，理论结合实践，更用心的在 IT 这条康庄大道上走下去。

## 七、设计模式

本软件采用标准 C/S 架构，客户端和服务端紧密连接，其中所使用到的设计模式如下：

- 1、工厂模式：创建棋盘类、创建房间匹配。
- 2、适配器模式：Android 开发中经常用到的适配器，来解决对象多样化问题。
- 3、责任链模式：在 AI 人机算法中，方法之间相互迭代，各个模块结果反馈密切，最终输出最优解。
- 4、线程池模式：在客户端和服务端中均建立线程池，多线程处理调度结果和计时器等消息。
- 5、并发型模式：用来处理多线程规范型调度，线程之间正常通信机制。

## 八、软件评估

（一）系统总体架构描述：

- 1、该游戏分为在线游戏对战平台子系统和提供网络连接功能的服务器子系统。

在线游戏对战平台满足交互性能需求，存在有内部自定义的人机交互处理功能。

服务器偏向于局部网内部信息共享与实时响应。

- 2、要将服务器系统构建具有较强的灵活性。
- 3、对战平台的兼容性和可移植性。



非正式表示法中的分层视图

## （二）搜集场景：

经过小组相关不同角色开发人员讨论，经整理后，得到部分与系统交互的场景。

部分与系统交互的场景描述	
序号	场景描述
1	利用对战平台进行双人同机对战检测
2	加入交互 AI 算法的进攻策略
3	将等待对方时间扩展成 1min
4	房间号从 3 位数字到 5 位的拓展
5	房间号的加密支持
6	数据访问间隔从 100ms 到 20ms 不等的测试
7	访问设备由 2 到 30 同时在线对战
8	强制下线后的系统处理测试
9	添加不同 Android 系统进行对战

## （三）生成质量效用树

对战平台根据不同用户定义了三种不同的游戏模式，同时定义了多个不同的对战技术框架，它们一起描述该系统的结构，以保证游戏的多样性。

质量效用树				
0 层	1 层求精	2 层求精(优先级, 个别场景略)	场景概念 (3 层)	场景求精 (4 层)
系统效用(根)	可用性	1、进入人机对战(L,L) 2、进入联网对战(M,L)	O(1) O(3)	<10ms(2) <10ms(2)
	兼容性	1、添加不同 android	对 Android7 不兼容	
	可修改性	1、添加新的 AI 算法难度控制 2、改善聊天界面	1、需运用深度学习训练 2、2 个小时前端优化	
	性能	1、每一次判断胜负(L,L)	O(n <sup>2</sup> )	<10ms(2)
		2、每一次旋转棋盘(M,L)	O(n <sup>2</sup> )	<10ms(3)
		3、一次等级切换(L,L)	O(1)	<10ms(2)
		4、每一次 AI 反应(H,L)	O(n <sup>3</sup> )	<10ms(6)
		5、网络连接(L,L)	O(1)	<10ms(7)
		6、网络获取对方状态(L,L)	O(1)	<10ms(5)

#### (四) 分析质量属性

##### 4.1、可用性

可用性关注于如何让用户简单容易的实现他想要完成的动作。在不断对添加不同 Android 系统进行对战的测试下。其用户体验均没有太大差异，对当前情景的可用性和兼容性都得到极大的保证。同时对场景中的数据访问的测试结果是我们当前系统确保游戏通畅的先决。而 20ms 对于视觉上是实时的，并不会耽误用户体验，提高可用性。

##### 4.2、兼容性

兼容性是游戏存在的必须条件，对于对战系统而言，由于构建时采用的是 C/S 结构，服务器端对局域网的支持对于不同手机支持游戏并不得到影响，而不管是对战还是下载，均对不同设备进行兼容，考虑了多种并发情况对服务器的承载情况，均做出了不同情况的处理，兼容性问题基本解决。

##### 4.3、性能

性能是对当前的游戏体验的评估，而在场景分析中，我们考虑了对用户等待时长和房间号码的选择，通过不同的测试结果选定了较好的体验度的 1min 的等待时间和 5 位长的房间

号，同时对 AI 算法进行不同程度的优化，设定了三种不同游戏难度供用户体验，其中中等模式专注防守，困难模式专注于进攻，带来的性能的提升。而且尽量减少服务器对数据的验证和计算，尽量让其在本机上完成检测计算，这样极大的减少了响应时长，带来极好的性能提升。

#### （五）评估结果：

对系统多方面的测试，通过场景抽取和场景分析法，采用 ATAM 对设计好的系统分析，可以得到一个完整的分析报告，由此也可看出不管对可用性，兼容性、性能的评估都得到较好的结果，表明在设计上对不同情景都有考虑。但是仍存在一定弊端，就是服务器的承载量较小，难以承载太多人同时在线进行游戏对战，此仍待改善。

## 九、技术说明

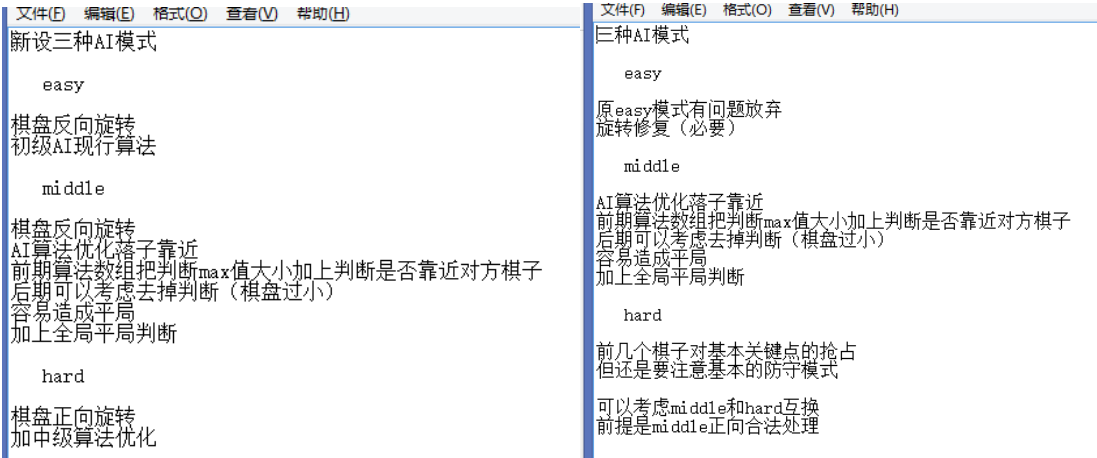
之前没有做过人机对战，突然一下要做 AI 感觉没有什么思路，在网上查设计 AI 的思路，感觉网上的棋类 AI 的特点是棋盘大，棋子固定，但是我们这个旋转五子棋有棋盘旋转的特点，每下一个子就会重绘一下棋盘，这对数组的处理要求很高，而且网上五子棋的 AI 算法完全不适用，于是就考虑算法思路，自己写旋转后的多重数组处理。在基础算法基本实现以后，又考虑改变算法中核心的权值分配，让 AI 由只顾防守到攻防兼备，达到一种和初玩此游戏的人思路差不多的落子方式。

这只是部分的 AI 代码，AI 的级别还不算特别高，一些固定的套路还不能很好的处理，接下来希望优化成向前分析两步，但对系统的计算能力一定有影响，现在的 AI 算法为落一子进行 8 次运算，如果分析两步就是  $8*8*8=512$  次运算，在运算中的启用多

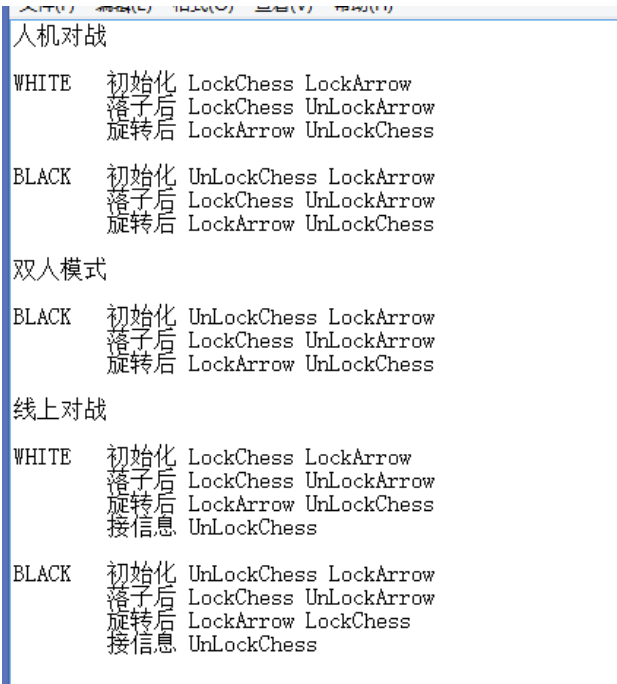
线程希望能节省些时间，另外还要优化算法，如果做适当的分析，能够降低到  
 $(8*1+1)*8=72$  次运算，大大降低资源调用，也能增强 AI 难度，为玩家提供更有挑战  
性的人机对战，我们一定会继续研究算法进行开发的。

十、难点攻关

在设计 AI 的过程中，进行了多次的推翻重构，每一次的想法都十分有意义。



中间还会把自己的一些思维整理下来，比如对棋盘的锁定方式，运用状态机方式：



而期间还会把逻辑抽象出来帮助思维：

		3	1	4	3	1	4
	0	1	2	3	4	5	6
5	1						
2	2						
6	3						
5	4						
2	5						
6	6						

还有很多很多思维过程都记录下来了，我觉得随着难点一个一个被攻破，这次的开发对自己思维的训练有很大帮助，希望以后会进行越来越多这样的项目，给自己提升的空间，为以后积攒经验。

## 十一、软件使用补充说明

Android 设备操作系统：

Android 版本	API
Android 4.4 KitKat	19
Android 4.4W	20
Android 5.0 Lollipop	21
Android 5.1 Lollipop	22
Android 6.0 Marshmallow	23