**ManilaOnline 2.0 详细设计说明书**

**郭浩 姚照原 张溶倩**

目录

[1 引言 2](#_Toc536046011)

[1.1编写目的 2](#_Toc536046012)

[1.2背景 2](#_Toc536046013)

[1.3定义 2](#_Toc536046014)

[1.4参考资料 2](#_Toc536046015)

[2 程序系统的结构 3](#_Toc536046016)

[3 流程控制模块 4](#_Toc536046017)

[3.1程序描述 4](#_Toc536046018)

[3.2性能 4](#_Toc536046019)

[3.3接口 4](#_Toc536046020)

[3.4流程逻辑 5](#_Toc536046021)

[4 机器算法模块 8](#_Toc536046022)

[4.1程序描述 8](#_Toc536046023)

[4.2性能 8](#_Toc536046024)

[4.3算法 8](#_Toc536046025)

[4.4接口 9](#_Toc536046026)

[4.5其它设想 9](#_Toc536046027)

[5. 用户交互模块 9](#_Toc536046028)

[5.1 概述 9](#_Toc536046029)

[5.2 功能 10](#_Toc536046030)

[5.3 输入项 10](#_Toc536046031)

[5.4 输出项 10](#_Toc536046032)

[5.5 流程逻辑 10](#_Toc536046033)

[5.6 限制条件 12](#_Toc536046034)

[6 Manila通信协议 12](#_Toc536046035)

[6.1 Manila请求报文 12](#_Toc536046036)

[6.2 Manila响应报文 12](#_Toc536046037)

[6.3 对象封装： 13](#_Toc536046038)

[7 服务器端用户登录管理模块 13](#_Toc536046039)

[7.1程序描述 13](#_Toc536046040)

[7.2功能 13](#_Toc536046041)

[7.3性能 14](#_Toc536046042)

[7.4输入项 14](#_Toc536046043)

[7.5输出项 14](#_Toc536046044)

[7.6对象封装 14](#_Toc536046045)

[8 服务端对局管理模块 16](#_Toc536046046)

[8.1程序描述 16](#_Toc536046047)

[8.2功能 16](#_Toc536046048)

[8.3性能 16](#_Toc536046049)

[8.4输入项 16](#_Toc536046050)

[8.5输出项 17](#_Toc536046051)

[8.6对象封装 17](#_Toc536046052)

[9 服务端用户注册模块 18](#_Toc536046053)

[9.1程序描述 18](#_Toc536046054)

[9.2功能 18](#_Toc536046055)

[9.3性能 18](#_Toc536046056)

[9.4输人项 19](#_Toc536046057)

[9.5输出项 19](#_Toc536046058)

[10 服务器配置 19](#_Toc536046059)

[10.1 主机 19](#_Toc536046060)

[10.2 硬件环境 19](#_Toc536046061)

[10.3 软件环境 19](#_Toc536046062)

# **1 引言**

### 1.1编写目的

本文旨在对ManilaOnline项目的2.0版本进行详细设计，明确本阶段内系统的主要体系结构和设计细节，并作为项目评价和验收的依据之一。本文由ManilaOnline项目的开发者编写，供北京大学2018年秋季学期软件工程原理课程的教师、选课同学、系统的用户以及所有对本项目感兴趣的开发者参阅。

### 1.2背景

本软件系统名为ManilaOnline2.0，由软件工程原理课程教师黄舟给出选题方向，课程第二小组讨论和选择产生。开发者包括本课程第二小组成员（郭浩、金恬、戴琪、姚照原、张溶倩），本系统的用户包括本课程教师、选课同学及其他任何可能的游戏玩家。本系统运行于广域网上，同时支持离线的单机功能。

### 1.3定义

Manila通信协议：用于在线Manila游戏的客户端和服务器之间的交互。

Manila操作指令集：马尼拉操作的编号-具体指令的集合。

### 1.4参考资料

1. 本项目需求分析文档：

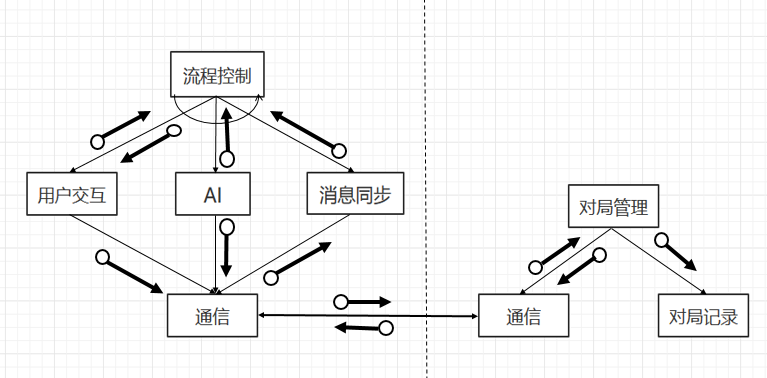
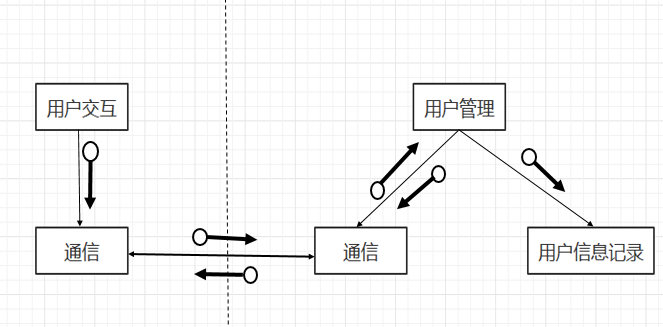
<https://gitee.com/yzyplus/ManilaOnline/blob/master/ManilaOnlineRequirementsSpecification.md>

1. 本项目概要设计文档

<https://gitee.com/yzyplus/ManilaOnline/blob/master/ManilaOnlineRequirementsSpecification.md>

1. 本项目1.0版本详细设计文档：详细设计-流程控制与机器算法-12\_11.docx

# 2 程序系统的结构

# 3 流程控制模块

### 3.1程序描述

将桌游Manila的规则实现为代码，并可以通过调用Player类的实例获取游戏决策。具体功能包括产生骰子值，维护棋盘状态、玩家资产等公共数据，推动游戏进行直至结束。程序会以日志的方式记录游戏中的所有事件，终局时输出到文件中。

### 3.2性能

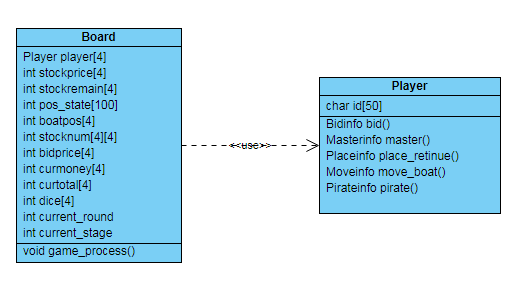
精度：处理的数据都为整型，计算方面没有精度问题。

响应时间：不涉及复杂计算，可以忽略不计。

灵活性：程序严格按照游戏流程顺序执行。下级模块会保证操作的合法性，最终版本中本模块对操作合法性不做排查，但在测试阶段，采用try-catch机制捕捉异常，防止机器算法出现错误。

### 3.3接口

本模块通过Player类与机器算法、消息同步、用户界面三个模块交互，具体方法为以无参方式调用Player实例的以下决策函数：竞价-bid(), 总督行使职权-master(), 放置随从-place\_retinue(), 海盗登船-pirate(), 领航员操纵船只-move\_boat()。函数的返回值、调用时机由下面具体说明。

本模块包括对Player类开放的公共数据区，即Board类的属性成员。

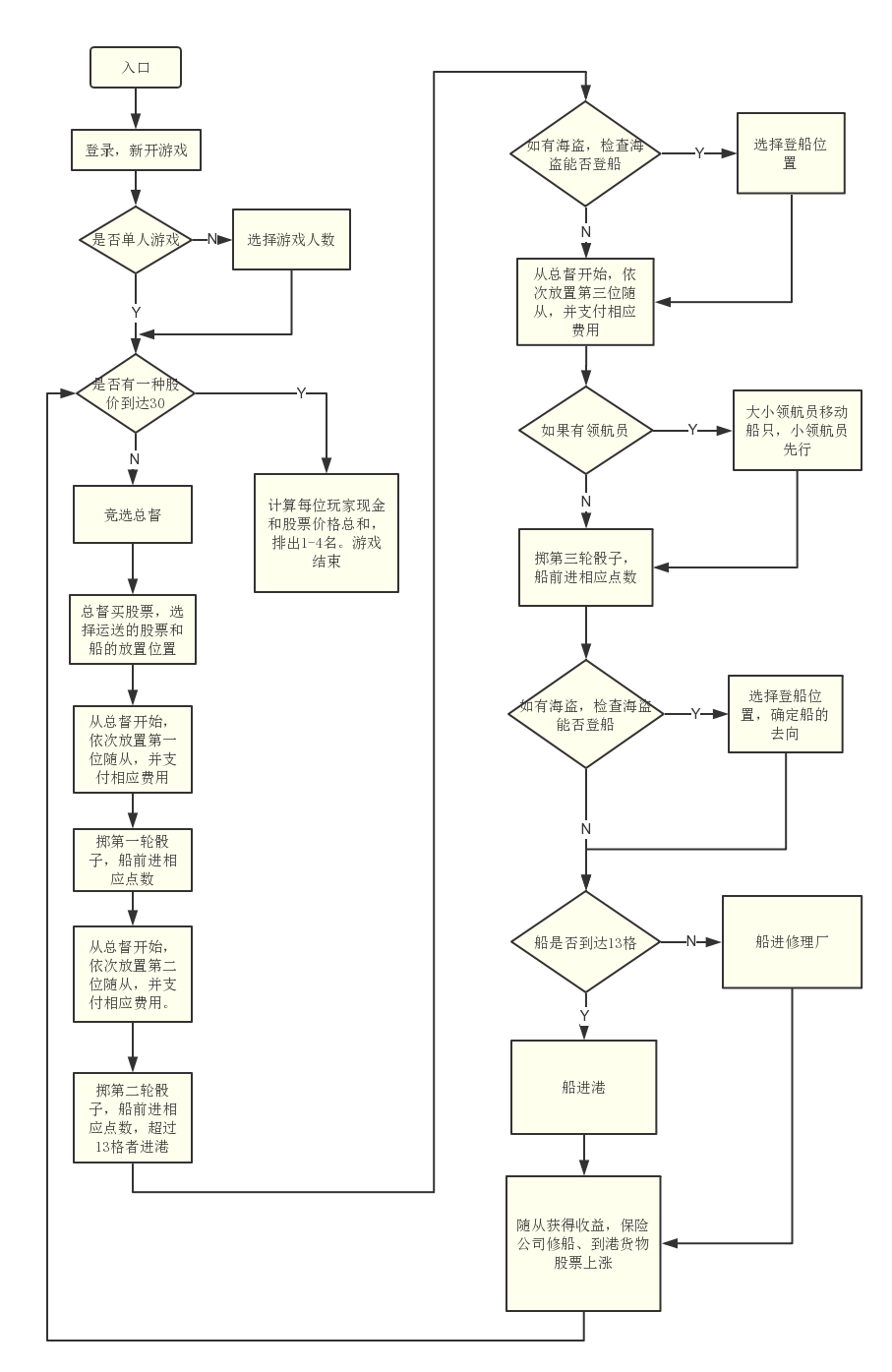
Board类与Player类成员一览

Player类五种info结构体包含当前步骤所有决策信息。具体来说，Bidinfo包括竞选报价（整型，下同；-1代表退出竞选）和抵押股票的数目；Masterinfo包括四种货物所在船的起点（0-5之间，有一种为-1，代表不装船），购买股票的类型，抵押股票的数目；Placeinfo包括防置随从的位置（可为空位置，与位置表对应），抵押股票的数目；Moveinfo包括四种货物所在船的移动格数（-2到+2）；Pirateinfo包括两个整型变量，第一个返回值表示选择登船的位置；在第二轮，不使用第二个返回值；在第三轮，第二个返回值表示船的去向。

Player的一些函数事实上需要传入参数（上图中略去）。Bid有一个参数表示上一未退出的玩家报价；move\_boat有参数1或2（分别代表小领航、大领航）；pirate有三个参数，第一个为身份（海盗船长为1，船员为2）；对于身份为船员的情形，第二轮中第二个参数为海盗船长的登船位置，第三个参数不使用；第三轮中第二个参数为海盗船长选择劫掠的船，第三个参数为海盗船长指定的船的去向。

### 3.4流程逻辑

流程控制模块流程图（摘自需求文档）如下图：



对每一回合游戏流程中的系列事件，归纳处理方法如下：

|  |  |  |
| --- | --- | --- |
| Stage值 | 事件名称 | 处理逻辑 |
| 0 | 竞价 | 从1号玩家开始，依次调用player成员的bid函数，直到三个player返回-1，保存总督成员的序号，之后放置随从从总督开始。  约束：底价为10，报价不超过手中现金 |
| 0 | 总督行使职权 | 调用总督成员的master函数  约束：四条船的起点中一个返回-1，其余在0-5之间且和为9. 所购买股票价格不超过手中现金。 |
| 1 | 放置随从 | 循环调用place\_retinue函数一次  约束：空位（pos\_state为-1）才可以放置，放置费用不超过手中现金。 |
| 1 | 掷第一轮骰子 | 为dice数组产生随机数 |
| 2 | 放置随从 | 同第一轮放置随从 |
| 2 | 掷第二轮骰子 | 为dice数组产生随机数，超过13格的船进港 |
| 2 | 海盗登船 | 若有船位置恰好为13且有海盗，调用对应player的pirate函数（先调用海盗船长，后调用船员）  约束：船员不能选择海盗船长登船的位置 |
| 3 | 放置随从 | 注意有船到达的空位也不能再放置随从，其余同第二轮放置随从 |
| 3 | 领航员操纵船只 | 若有领航员，调用对应player的move\_boat函数（先调用小领航，后调用大领航）  约束：小领航返回值移动绝对值之和为1，大领航为2 |
| 3 | 掷第三轮骰子 | 为dice数组产生随机数，超过13格的船进港 |
| 3 | 海盗登船 | 若有船位置恰好为13且有海盗，调用对应player的pirate函数（先调用海盗船长，后调用船员）；根据返回值确定船的去向 |
| 3 | 结算 | 除被海盗劫掠的船外，达到13格的船进港，否则进修理厂（建立中间变量记录船的去向）；向player分配随从收益，保险公司赔付（修改curmoney）；修改股价值（stockprice）；判断游戏是否结束。  注：若保险公司赔不起修理费，现金扣至零即可。 |

# 4 机器算法模块

### 4.1程序描述

用于参与单人模式（或多人模式玩家不足的情形）的游戏。在程序中的实体为Player的派生类，需要时将其实例化。其核心是决策函数中的算法。这一模块将会有多种实现，每一种实现可作为独立的组件加入程序，只要其符合流程控制模块定义的接口。

### 4.2性能

响应时间：运算的时间视算法的复杂性而定，朴素算法（不涉及机器学习）可在毫秒级出结果，实际游戏时可人为设定延迟，以带来真实的游戏体验。机器学习算法的响应时间为秒级。

精度：通过访问公共数据区中的棋盘当前状态，保证所作决策的合法性。

### 4.3算法

#### 4.3.1 朴素经验算法（Mr.Naive）

|  |  |
| --- | --- |
| 函数 | 决策思路 |
| Bid() | 对总督的支付意愿定义为20与（手中现金减去20）的较小值，高于该支付意愿时退出竞价。 |
| Master() | 选择自己拥有股票的货物装船，不足三种时随机选择填满。拥有一种股票时位置设为（5,2,2），两种设为（5,4,0），三种时根据股票数量分配（如三种各一张，设为（3,3,3）；有一种为2张时设为（5,2,2））  注：若有四种股票（少见），选择拥有张数较多的三种装船。 |
| Place\_retinue() | 第一轮：选择有空位的船只中最靠前的一只放置随从  第二轮：以每回合骰子点数3.5估算船能否到达，对船上、港口、修理厂的所有空位，根据船去向的估算结果计算收益，取收益与成本之差最大的一个。  第三轮：假定位置在10及以上的船可以到达，其它船不能到达，计算所有位置的收益，取收益与成本之差最大的一个。（注：领航员的收益为可以救回的有自己随从的船只的收益；海盗的收益为处于7到12（含7和12）之间船只货价之和的六分之一） |
| Move\_boat() | I．选择位置在7至10之间（含7和10）且有自己随从的船，有多只时选择到港收益较大的一只向前移动。  II.当I不能实现，选择位置在7至10之间且没有自己随从的船只向后移动。  III.当II不能实现，选择位置11至13之间且有自己随从的船，有多只时选择到港收益较大的一只向前移动。  IV.当III不能实现，选择位置11至13之间且没有自己随从的船向后移动。 |
| Pirate() | 若在第二轮，选择收益较大的船登船，替代船上（除海盗船长外）当前总资产最高玩家的随从。  若在第三轮，选择收益较大的船劫掠，并让船开到港口。 |

#### 4.3.2其他算法

* + 1. 随机决策算法（Random）

为了避免决策的恒定性，放置随从的决策改为：根据局面调整选择各位置的概率，然后由随机数产生决策。

* + 1. 精确期望算法（Expector）

枚举掷骰子的所有结果可以求出各船到港的准确概率，据此可求出船上、码头、修理厂、保险公司、海盗等位置收益的准确期望，为放置随从提供更可靠的依据。

* + 1. 进取与稳健切换算法（Weatherlight）

在精确计算期望和方差的基础上，根据自己的资产排名在稳健、进取两种模式间切换，稳健型以作为决策依据，进取型以型决策依据。

（后三种算法的细节待完善）

### 4.4接口

每一个机器算法参与游戏时被Board类的game\_process函数调用。决策时访问Board类的公共数据区。具体机制参见流程控制模块的接口描述。

### 4.5其它设想

1.把其它玩家的即时总资产加入决策，在自己发展的同时遏制领先玩家获利。

2.实现机器学习算法，采用“局面-选择”类型的历史数据进行训练。

# 5. 用户交互模块

### 5.1 概述

用于推进整个游戏进程，接收并响应用户的操作。展示当前各个玩家的公开信息和游戏进行阶段，向每个玩家展示各自的私人信息，并提示玩家在对应进程节点做出相应决策。

### 5.2 功能

主界面：获得游戏模式的选择；下一步操作（开始游戏或查看历史记录或进入商城）。

游戏界面：显示玩家财富；显示骰子点数；显示货船位置；显示股价；显示随从位置；获取竞价信息；接收总督行使权力的各种操作结果；接收随从位置；接收海盗操作；接收领航员操作；接收抵押股票信息；显示终局统计数据。

历史数据界面：表格显示当前玩家的所有游戏结果记录。

### 5.3 输入项

游戏模式选择；

玩家出价；总督选择的股票；总督选择每艘船上的货物；总督选择每艘船的起点位置；玩家放置随从的位置；海盗选择登船的位置；领航员选择船前进的步数；（异常：玩家选择抵押的股票）；

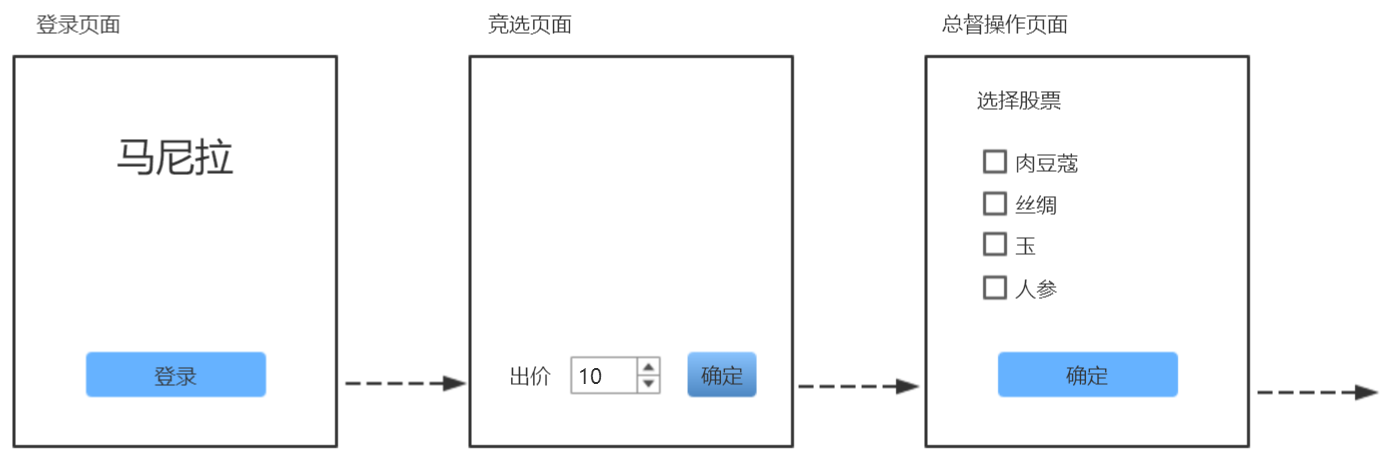
### 5.4 输出项

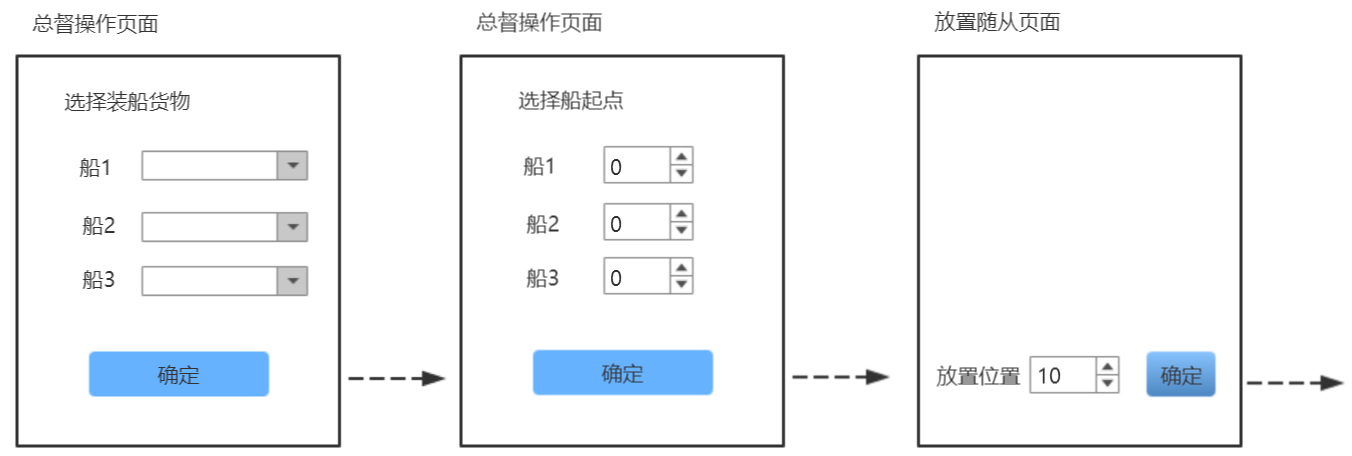
各个玩家当前财富；各支股票当前股价；掷骰子的结果；船的位置；终局的各种统计数据

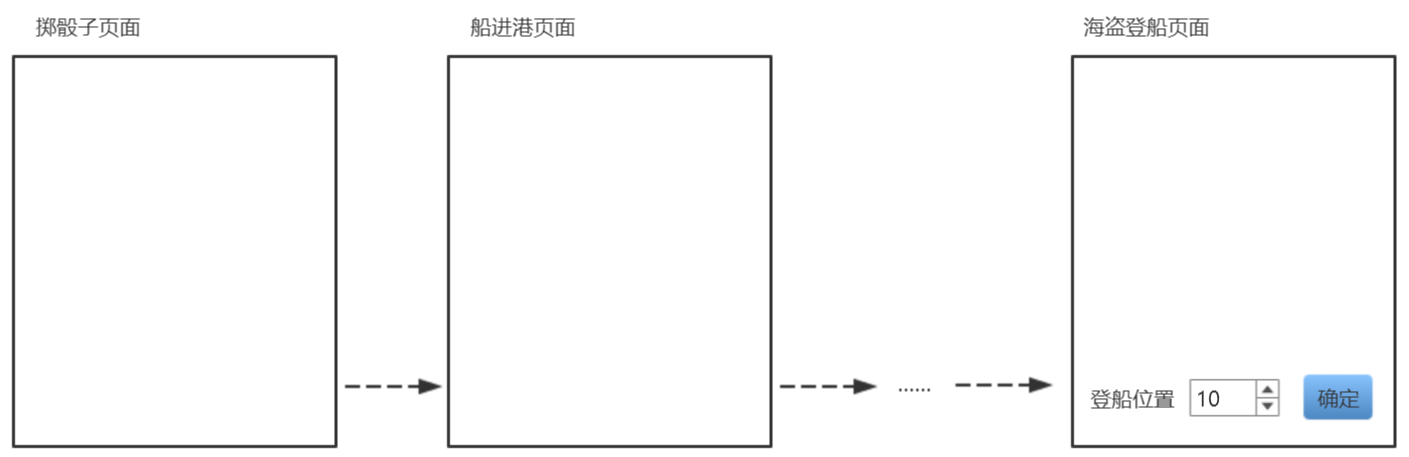
### 5.5 流程逻辑

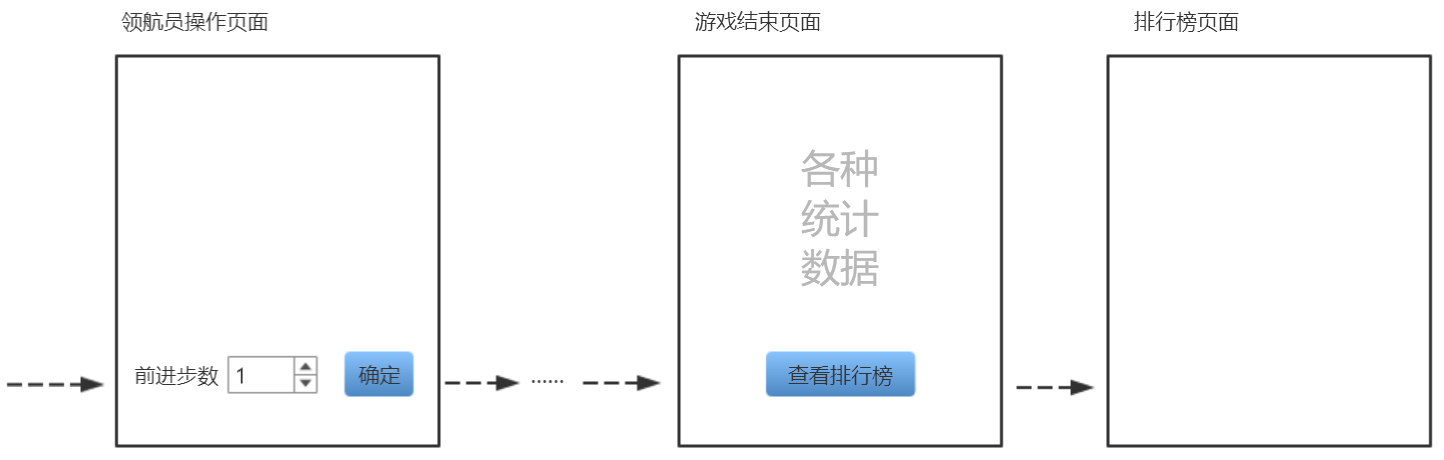
页面流：



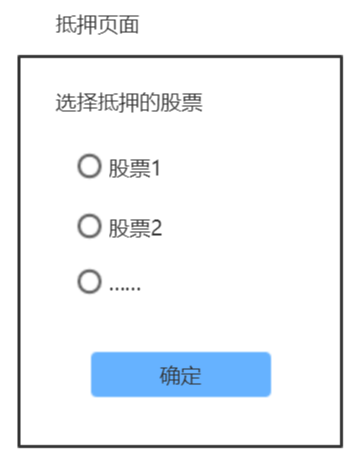








异常：



### 5.6 限制条件

无明显限制条件。网络通信状况不好时或出现操作与反馈不及时或不同步。

# 6 Manila通信协议

### 6.1 Manila请求报文

请求报文是用于客户端向服务器发出连接请求的字符串消息格式。

请求报文 = 报文头+“REQUEST=”+ 请求分类 + {“&”+补充信息}

报文头 = “$”

请求分类 = [“LOGIN”,“LOGOFF”,“ONLINE”,“TESTCONNECT”]

补充信息 = 信息项目 + “=”+项目内容

信息项目 = [“NAME”,“ID”,“PWD”,“MAIL”,“HONOR”,“DO”,“EXTRA”]

项目内容 = 1{字母}10

字母 = [“a”,“b”,“c”,“d”,“e”,“f”,“g”,“h”,“i”,“j”,“k”,“l”,“m”,“n”,“o”,“p”,“q”,“r”,“s”,“t”,“u”,“v”,“w”,“x”,“y”,“z”,“0”,“1”,“2”,“3”,“4”,“5”,“6”,“7”,“8”,“9”]

### 6.2 Manila响应报文

响应报文是用于服务器向客户端发出连接响应的字符串消息格式。

响应报文 = 报文头+“SPEAKER=”+ 响应来源 + {“&”+补充信息}

报文头 = “$”

响应来源 = [“HOST”,用户ID]

用户ID = 1..99999

补充信息 = 信息项目 + “=”+项目内容

信息项目 = [“NAME”,“ID”,“MSG”,“MAIL”,“HONOR”,“DO”,“EXTRA”]

项目内容 = 1{字母}10

字母 = [“a”,“b”,“c”,“d”,“e”,“f”,“g”,“h”,“i”,“j”,“k”,“l”,“m”,“n”,“o”,“p”,“q”,“r”,“s”,“t”,“u”,“v”,“w”,“x”,“y”,“z”,“0”,“1”,“2”,“3”,“4”,“5”,“6”,“7”,“8”,“9”]

### 6.3 对象封装：

Manila通信报文封装为ManilaMSG类，无方法，对象的成员有且仅有一个map类型的body，其键为信息项目或请求分类或响应来源，值为非空字符串。

# 7 服务器端用户登录管理模块

### 7.1程序描述

用户登录管理模块主要提供对游戏玩家在线登录的支持，本系统独立于游戏对战的进程，运行于服务器端，常驻与内存，监听特定端口(port:7000)，有一定的并发性（同时可接收的最大登录请求为10人次）。用户登录时验证数据库内是否有此用户存在，每个用户的用户名、邮箱、ID都是唯一的，用户可以通过任何一种方式+密码验证的方式登录，也可以修改密码和用户名，前提是目标用户名可用，系统会提供一个公共的只读账户来进行密码的验证。

### 7.2功能

此模块的功能主要是接收用户的登录请求，并验证登录者提供的身份信息，通过验证以后将此用户添加到活跃用户集合中去。

当用户发出在线开局的请求的时候，将用户加入到匹配队列中去，按照顺序进行匹配，如果当本次接收消息结束时还有空余的玩家请求匹配队友，那么就将剩余玩家用机器人补足，机器人运行在服务器上，然后将这些人的控制权交给服务器端游戏进程控制模块。

每次在线对局结束以后，用户都需要重新登录。

### 7.3性能

用户管理系统的功能比较简单，模式也比较固定，主要是数据库记录的添加删除修改，并不涉及大规模、较复杂的计算，也不要求强的灵活性。

本系统的重点是数据记录的准确性。数据库本身对数据修改有一整套完整的保证安全性和完整性的机制，可以满足使用的需求。对于在网络传输中可能会发生的丢包的现象，通过在程序中添加消息边界实现维护。

要求的性能主要取决于客户端和服务器之间的通信时延和用户信息管理使用的MySQL引擎的查询速度。本系统面向的对象规模较小，在千量级左右，而使用的数据库引擎处理这种量级的数据量是秒以内的响应速度，所以在时间上完全满足。而位于广域网上的服务器在华北地区，经过测试有40~50ms的时延，传输速度较快，由于数据库容量较小，每次查询在秒以内，所以对于终端来说，这个时间精度是可以满足的。

### 7.4输入项

输入包括用户昵称（必选），注册用邮箱（可选），密码（必选），以及其他可能扩展的信息；用户通过提交表单的形式进行输入，输入的内容需要符合Manila请求报文的格式。

用户昵称：字符串类型，不超过20个字符，用户可以输入任意字符；

注册邮箱：字符串类型，要求满足邮箱的基本格式；

密码：字符串类型，可以输入任意字符；要求在前端进行字符串检查，密码中不能包含特殊字符。

### 7.5输出项

服务器端的输出主要是对用户而言，发送的消息符合Manila响应报文格式。

### 7.6对象封装

用户登录管理模块封装为ManilaServer类

#### 7.6.1 ManilaServer类的成员：

|  |  |  |  |
| --- | --- | --- | --- |
| 成员名 | 成员类型 | 成员的含义 | 成员初始值 |
| PORT | 字符串正整数 | 用户管理的端口号 | ‘7000’ |
| serverSocket | 套接字对象 | 监听self.PORT | 空套接字对象 |
| USER\_MAP | 字典 | 在线用户集合 | {} |
| CONNECTION\_LIST | 数组 | 当前所有需要监听的套接字集合 | [self.serverSocket] |
| socketMap | 字典 | 用户ID:用户套接字 | {} |
| idMap | 字典 | 用户套接字：用户ID | {} |
| ONLINE\_LIST | 数组 | 等待匹配的用户ID集合 | [] |
| GAMING\_MAP | 字典 | 当前进行的在线游戏的集合 | {} |
| GamingCount | 整型 | 已经完成匹配的游戏对局数 | 0 |

#### 7.6.2 ManilaServer类的方法：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 描述 |
| \_\_init\_\_ | 无 | 无 | 初始化 |
| broadcast | 消息来源用户ID,  要广播的消息 | 无 | 广播消息 |
| p2psend | 目标用户ID,  要发送的消息 | 无 | 单独发给某人 |
| login | 登录请求套接字，  收到的请求报文 | 无 | 用户登录 |
| logoff | 退出请求套接字，  收到的请求报文 | 无 | 用户退出登录 |
| recv\_msg | 监听到消息的套接字对象 | 无 | 处理收到的消息 |
| new\_gaming | 无 | 新游戏的编号 | 开一盘在线游戏 |
| socket\_handle | 无 | 无 | 控制服务器进程，是每个对局的父进程 |

# 8 服务端对局管理模块

### 8.1程序描述

服务器端对局管理模块主要提供对游戏玩家在线对局的支持，本系统独立于客户端游戏对战的进程，由服务器端用户登录管理模块调用，运行于服务器端，常驻与内存，监听特定端口(port:71xx)，通过循环来模拟并发处理（同时可接收的最大游戏对局人数为4人，但是由于允许旁观，所以旁观者无上限，模拟的并发上限收到服务器处理能力的限制）。

### 8.2功能

此模块的功能主要是提供对游戏玩家在线对局的支持，接收用户的进入对战请求，并验证登录者是旁观者还是当局者，通过简单查证以后将此用户套接字更新到本对局用户集合中去。

创建对局时，需要包含在线请求匹配用户的id：对象集合。

当用户发出登录本局游戏请求的时候，判断用户是否为旁观者并作相应分类，对不同分类的用户的消息均进行检查，如果符合请求报文特征，则将此条报文转换成响应报文并广播。

在线对局结束以后，删除本对象，回收内存空间。

### 8.3性能

服务器端对局管理模块的功能比较简单，模式也比较固定，主要是用户消息的转播与存储。

本系统的重点是数据记录的准确性。文件的读写每次都是读写完成以后关闭文件指针，除过游戏开始时创建文档以外，每次读写都是文件尾追加方式，并且每个对局的ID均不相同，文件记录名称与此ID挂钩，所以安全性有所保证。

要求的性能主要取决于客户端和服务器之间的通信时延和对局管理系统转发和记录消息的速度。本系统面向的对象规模较小，在千量级左右。每个对局管理模块处理4个玩家，由于实际使用中旁观者并不多，所以每个端口需要监听的消息数量较少，所以在时间和性能上完全满足要求。而位于广域网上的服务器在华北地区，经过测试有40~50ms的时延，传输速度较快，由于消息格式简单，每次广播和记录都可以在很短时间内完成，所以对于服务器来说，这个时间消耗是可以接受的。

### 8.4输入项

输入包括一条包含用户ID（必选），用户请求（必选），操作时间（必选），操作（可选），以及其他可能扩展的信息；用户通过客户端封装的manilaMSG成员的send方法输入，输入的内容需要符合Manila请求报文的格式。服务器循环监听端口的消息，有消息时读取这个输入。

### 8.5输出项

服务器端的输出主要是对用户而言，发送的消息符合Manila响应报文格式。

### 8.6对象封装

用户登录管理模块封装为ManilaServerGaming类

#### 8.6.1 ManilaServerGaming类的成员：

|  |  |  |  |
| --- | --- | --- | --- |
| 成员名 | 成员类型 | 成员的含义 | 成员初始值 |
| ID | 整型 | 本场对局的编号 | 来自参数 |
| PORT | 字符串正整数 | 本场对局的端口号(房间号) | ‘7000’ |
| serverSocket | 套接字对象 | 监听self.PORT | 空套接字对象 |
| CONNECTION\_LIST | 数组 | 当前游戏中所有需要监听的套接字集合 | [self.serverSocket] |
| user\_in | 字典 | 当局者集合 id:玩家 | 来自参数 |
| watch | 字典 | 旁观者集合 id:玩家 | {} |
| user\_count | 整型 | 已经进入游戏的玩家数 | 0 |

#### 8.6.2 ManilaServer类的方法：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 描述 |
| \_\_init\_\_ | 无 | 无 | 初始化 |
| \_\_del\_\_ | 无 | 无 | 注销变量 |
| broadcast | 消息来源用户ID,  要广播的消息 | 无 | 广播消息 |
| p2psend | 目标用户ID,  要发送的消息 | 无 | 单独发给某人 |
| login | 登录请求套接字，  收到的请求报文 | 无 | 用户登录 |
| recv\_msg | 监听到消息的套接字对象 | 无 | 处理收到的消息 |
| Start | 无 | 无 | 控制一盘在线游戏，用于被用户登录模块控制产生子线程的函数 |

# 9 服务端用户注册模块

### 9.1程序描述

服务器端用户注册模块主要提供对游戏玩家注册游戏的支持，本系统独立于全部其他模块，通过数据库进行数据共享，运行于服务器端，常驻于内存，监听特定端口(port:8080)，通过Django进行开发和处理，依赖于niginx服务，拥有并发处理能力。用户注册时验证数据库内是否有此用户存在，每个用户的用户名、邮箱、ID都是唯一的，用户可以通过任何一种方式+密码验证的方式登录，也可以修改密码和用户名，前提是目标用户名可用，系统会提供一个公共的只读账户来进行密码的验证。

### 9.2功能

此模块的功能主要是提供对游戏玩家注册游戏的支持，接收用户的注册请求。当用户发出登录本局游戏请求的时候，判断注册信息是否已经存在，对已存在的用户名进行报错，如果符合注册条件，则将此注册信息写入数据库并返回成功提示。

### 9.3性能

用户管理系统的功能比较简单，模式也比较固定，主要是数据库记录的添加删除修改，并不涉及大规模、较复杂的计算，也不要求强的灵活性。

本系统的重点是数据记录的准确性。数据库本身对数据修改有一整套完整的保证安全性和完整性的机制，可以满足使用的需求。对于在网络传输中可能会发生的丢包的现象，通过在程序中添加消息边界实现维护。

要求的性能主要取决于客户端和服务器之间的通信时延和用户信息管理使用的MySQL引擎的查询速度。本系统面向的对象规模较小，在千量级左右，而使用的数据库引擎处理这种量级的数据量是秒以内的响应速度，所以在时间上完全满足。而位于广域网上的服务器在海外，经过测试有100~300ms的时延，但是传输速度较快，所以对于每15秒执行一次操作的终端来说，这个时间精度是可以满足的。

### 9.4输人项

输入包括用户昵称，注册用邮箱，密码，以及其他可能扩展的信息；用户通过提交表单的形式进行输入。

用户昵称：字符串类型，不超过20个字符，用户可以输入任意字符；

注册邮箱：字符串类型，要求满足邮箱的基本格式；

密码：字符串类型，可以输入任意字符；要求在前端进行两次输入以确认密码，并进行字符串匹配检查。

### 9.5输出项

输出项包括但不限于弹出一张注册成功的页面或者注册失败的页面，固化在程序中，显示后5秒自动转到登陆界面。并不涉及保密性。

# 10 服务器配置

### 10.1 主机

公网IP:39.106.10.117

SSH端口号: 22

### 10.2 硬件环境

硬盘：40G SSD

内存：1GB

带宽：1Mbps

### 10.3 软件环境

操作系统：CentOS 6.8 64位

数据库：MySQL5.7

Web服务器：Apache，Niginx

解释器：Python3.7