项目类型：软件设计方案

项目名称:Linux平台上“玩转地主”游戏设计

设计成员：李卓立、高奕丽、蒋天泽、吴涛、张赟

客户端设计

**显示结构：**

**·抽象类**

1. **类**
   1. **函数**

**·登录模块：class loginModular**

1. 页面交互 class interaction
   1. 登录欢迎 display\_Welcome()
   2. 操作说明 display\_Help()
2. 用户注册 class registration
   1. 提示注册 prompt\_Regist()
   2. 服务器验证合法 check\_Regist()
3. 用户登录 class login
   1. 用户发送登录信息 trans\_Account()
   2. 服务器返回信息 return\_Account()
4. 登录等待 class waitLogin
   1. 等待三名用户在线 wait\_Opponent()
   2. 等待中进行文字通讯 messaging()
   3. 三名用户需手动输入准备指令 is\_Ready()
   4. 投票踢人 kick()

**·游戏模块：class gameModular**

1. 数据传输 class Gamedata
   1. 接受牌局信息 init\_Game()
   2. 按次序叫地主 call\_Land()
   3. 接受地主信息 specify\_Land()
   4. 出牌信息验证 usedCard\_Check()
   5. 发送出牌信息 usedCard\_Trans()
2. 界面交互 class interaction
   1. 显示三人信息 display\_Triangle()
   2. 显示牌局信息 display\_AllCard()
   3. 显示聊天信息 display\_Massage()
   4. 提示用户叫地主 callLand\_Opponent()
   5. 提示用户出牌 playCard\_Opponent()
   6. 等待其他用户出牌 waiting\_Others()
3. 计时功能 class time
   1. 计时，超时自动按规则出牌 auto\_Play()
4. 聊天功能 class game\_On\_Chatting
   1. 用户输入文字 send\_Message()
   2. 接受聊天信息 receive\_Message()

**·传输模块：class transportModular**

1. 单用户向服务端发送 class dispatch\_single
   1. 发送字符串并处理 dispatch\_Str()
2. 单用户接受服务端返回 class feedback\_Single
   1. 接收字符串并处理 receive\_Str()

**·得分模块：class scoreModular**

1. 得分统计 class scoreCount
   1. 每局结束后显示得分 display\_Score()
2. 得分查询 class checkScore
   1. 游戏中进行得分查询 ask\_Score()

**·注销模块: class logoutModular**

1. 输入指定指令退出 class ask\_Logout
   1. 向服务器发送信息 trans\_Logout()
   2. 接收信息并退出 receive\_Logout()
2. 非法关闭可进行等待和恢复 class repair
   1. 将牌局保留 preserve\_Game()
   2. 超时后该用户自动出牌 auto\_Play()

服务端设计

**·验证模块：class verifyModular**

1. 验证登录 class checkLogin
   1. 接收注册信息 get\_Regist()
   2. 接收登录信息 get\_Login()
   3. 数据库操作 database\_Opera()
   4. 返回登录消息 login\_Return()

**·游戏模块: class gameModular**

1. 牌局信息 class gameData
   1. 进行牌局初始化，次序初始化及发送 class init\_Game()
   2. 接受用户叫地主信息，返回地主信息 class return\_Land()
   3. 向多用户发送牌信息 dispatch\_Card()
   4. 接受用户出牌信息 used\_Card\_Receive()
   5. 按游戏规则进行出牌信息验证※ usedcard\_Check()
   6. 更新牌局 update\_Game()
   7. 发送新牌局 return\_New()
2. 胜负判断 class judgeOutcome()
   1. 胜负判断 is\_Win()
   2. 得分记录 score\_Note()
3. 得分统计 class score
   1. 得分写入数据库 keep\_Score()

**·传输模块 class transportModular**

1. 接收数据 class receiveData
   1. 接收单用户信息字符串并处理 receive\_Str\_Single()
2. 发送数据
   1. 向单用户发送字符串 send\_Str\_Single()
   2. 向多用户发送字符串 send\_Str\_Mul()

**·统计模块 class accoutModular**

1. 查询得分 ask\_Score()
   1. 接收客户端指令查询 ask\_Score\_Receive()
   2. 返回得分信息 ask\_Score\_Return()

**·注销模块 class logoutModular**

1. 用户注销 class logout
   1. 接收用户注销请求 ask\_Logout\_receive()
   2. 判断 ask\_Logout\_judge()
   3. 返回注销信息 ask\_Logout\_returm()

**·数据库：**

Linux平台 MySQL 运行于在服务端

**·表：**

1. user：
   1. UserName varchar //用户名
   2. UserPassword varchar //密码
   3. Score int //得分
   4. Win int //胜局
   5. Lose int //败局
   6. Status varchar //用户状态
   7. Time int //用户计时
2. Card
   1. PresentCard varchar //当前牌局
   2. HistoryCard varchar //已出牌列表
   3. LandName varchar //当前地主的用户
   4. User1Card varchar //用户1当前手牌
   5. User2Card varchar //用户2当前手牌
   6. User3Card varchar //用户3当前手牌
   7. User1CardNum varchar //用户1当前手牌数
   8. User2CardNum varchar //用户2当前手牌数
   9. User3CardNum varchar //用户3当前手牌数

模块关系

客户端

服务端

指令集设计：

1. 全局通用
   1. help ：显示操作说明文字列表，按q返回
   2. exit ：退出游戏
2. 登录后使用
   1. say [Saywhat] ：向所有用户发送Saywhat对应的聊天文字
   2. register [UserName] [Password] ：以UserName和Password注册
   3. login [UserName] [Password] :以UserName和Password登录
   4. user ：显示当前已登录用户信息
   5. ready ：进行准备
   6. kick [UserName] ：对UserName用户发起投票踢人
3. 开始游戏后使用
   1. show ：显示局上已经出过的牌
   2. go [CardNum] ：打出对应序号的牌
   3. pass ：跳过出牌
4. 游戏结束后使用
   1. record ：查询个人胜负记录
   2. score [Num] ：查询第Num局的得分
   3. logout ：注销当前用户