

《狼人杀》 答辩展示

Group 5

王晓智 2016011257

吴一凡 2016011269

王世因 2016011246

刘明明 2016011263

李瀚祎 2016012084

视频演示&功能介绍

特色功能

- 美观易用的UI
- 游戏中交谈分为房间频道和狼人频道，二者间不共通消息
- 投票使用从列表中选择一个玩家并确认的方式，非常直观
- 不同类型的信息以不同颜色来标注，一目了然
- 倒计时栏来提示玩家进行当前操作的剩余时间

特色功能

- 加入房间设定
- 输入房间号进入房间，若是第一个进入房间则为房主，开始设定这局游戏的配置（村民数、狼数、各种神都有没有）
- 在房间中存在准备/取消准备设定，只有当所有玩家均准备后房主才可以点击“开始游戏”按钮进入游戏

特色功能

- 允许同时投票，提升游戏体验
- 如果要是玩家一个一个投票，就会有漫长的等待时间...
- 而我们可以做到设定一段投票时间所有玩家一起投票
- 事实上，要做到这一点并不非常容易

特色功能

- 服务器完全异步实现无阻塞
- 这意味着可以在远程服务器的多个房间上同时运行多局游戏
- 统一投票——典型的阻塞操作（只有当集齐选票后，游戏逻辑才能进行下去）

特色功能

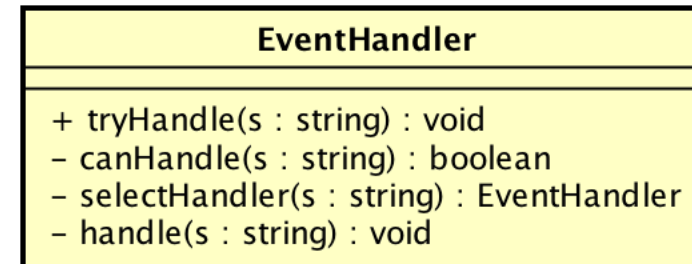
- 服务器搭建在远程服务器上，无需考虑ip

通信架构

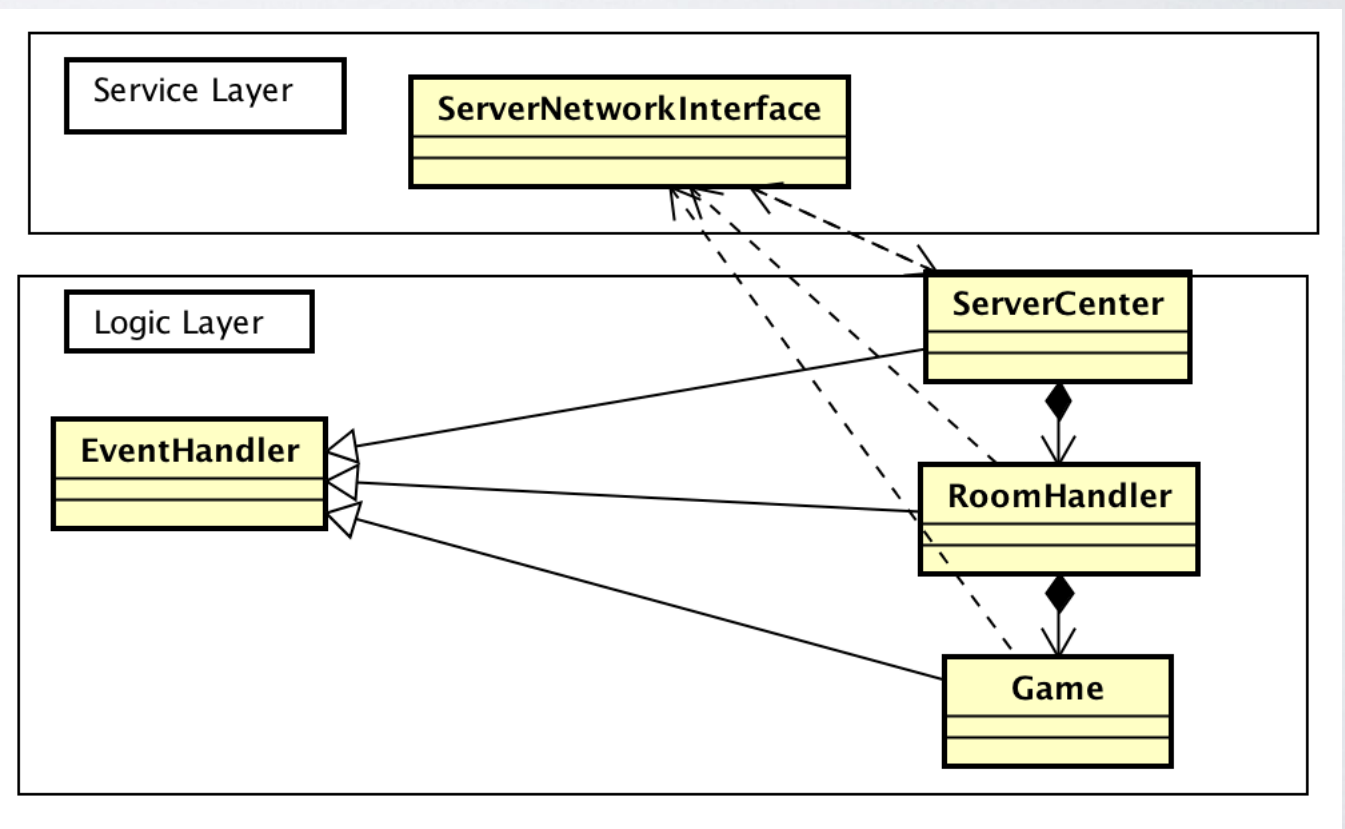
- 上面的设计中，同时存在房间、游戏，其逻辑不同，且都需要管理。
- 但是可以发现游戏是基于房间来进行的，事实上我们经常会遇到这种类似的情况，也就是一种逻辑处理的用户是基于另外一种逻辑产生的。

通信架构

- 使用抽象类EventHandler的子类来构建一颗“责任树”。
- 右下为服务端架构。（客户端架构类似）



```
tryHandle(s)
{
    if (canHandle(s))
        handle(s);
    else
        selectHandler(s)->tryhandle(s);
}
```



通信架构

- 每次收到命令首先交给ServerCenter，如果不能处理就将命令不断下传。
- 存在一个寻找下一个Handler的方法，故为在责任链设计模式基础上的拓展，故称“责任树”。
- 优势：方便进行管理以及内存回收；遵循单一责任原则，代码易于复用、拓展。
- 过程中使用的设计模式：责任链模式、代理模式。

游戏逻辑架构（设想）

- 无阻塞的进行游戏逻辑，也就是每收到一个包的时候要知道现在逻辑处理到哪里了。
- 对于游戏中的两种核心操作（投票、发言）进行抽象成为接口类，并维护其子类的事件双端队列，当遇到阻塞事件时放到队头，否则放在队尾。遇到一个包的时候就取出队头的处理方法进行处理。
- 若时间允许会考虑尝试这种架构。

编码技巧

- 文档先行：
- 经过讨论后将各种约定总结为清晰而详尽的文档，方便写代码时查阅

游戏正式开始后(Game阶段)

此时client界面需要显示的包括Status包传递的信息（多少人活着）等，此外关键需要控制开闭的就是两个地方。
一个是一个投票按钮，功能为指定一人（向server发送一个vote包）（此按钮为一次性，每次收到Set包指示打开后只能选择一次），这个选择按钮的含义会有很多变化，具体在游戏流程中解释。
第二个是聊天框，开闭严格受server传来的Set包控制，对应为向server发送chat包（room/wolf）。

server向client传递包格式规定

- Status

username role 1/0 1/0

username 1/0 1/0

...

报告当前游戏状态（收到此包刷新一次游戏状态显示），第二行为自己的username，role为自己的游戏角色，两个1/0依次为是否活着，是否是警长
游戏角色代号为villager,werewolf,prophet,witch,defender,hunter
第三行到最后各行为一个用户的状态报告（也会包括自己的，顺序固定），1/0含义相同

- Set

0/1 0/1 channel //能否投票，能否说话，什么频道（Room/Wolf）

设置client允许的行为，第二行两个1/0依次为投票按钮是否打开，聊天框是否能发送信息
channel为一个字符串（Room/Wolf），表示当前发言在什么频道（Room频道全房间皆可见，Wolf只有狼人伙伴可见）

- Day xx

表示进入了第xx个白天

内容目录

逻辑主文档

所有包的格式以及相关约定

游戏正式开始前（房间

client向server传递

server向client传递

游戏正式开始后(Game

server向client传递

client向server传递

游戏流程简述

游戏开始前

正式游戏阶段

夜晚

白天

编码技巧

- 利用约定的唯一的身份标识来实现解耦
- 所有类均以唯一的字符串来代表一个用户，解除了维护用户信息时的部分耦合关系

编码技巧

- 关于用户操作合法性判定
- 客户端采取“包驱动”的方法运行，比如客户端的信息发送按钮只有当接收到服务器“你可以开始说话了”的包的时候才能激活，这样使得客户端的表现符合服务端的预期，过滤掉用户大量不合法操作
- 对于其余的不合法操作则由客户端本地逻辑中的Validator判掉，不会提交给服务端给服务端带来额外负担

开源框架使用

- 使用成熟的QT框架来实现GUI以及网络通信
- 组内使用Git来维护代码并开源发布

组员分工

- 吴一凡（组长）：网络通信接口、远程服务器搭建、客户端逻辑、整体架构设计
- 王晓智：服务端逻辑（70%Commit）、服务端架构设计
- 李瀚祎：客户端GUI制作、PPT制作
- 王世因：客户端GUI制作、客户端逻辑
- 刘明明：界面设计、文档维护、调试

Thank you for listening.

–Questions are welcomed.