

IS301 计算机通信网络大作业 聊天程序

段淇瀚

电子信息与电气工程学院

2019 级信息安全专业

duanqihan@sjtu.edu.cn

1. 介绍

本项目主要基于 python 的 socket 和 pyqt 进行编程，其主体功能是实现线上用户的即时聊天（不支持离线留言），交互逻辑为所有登陆用户均为好友，均在同一群聊中，在群聊中发送的消息所有人可见；默认界面为群聊界面，如需私聊，只需双击用户列表对应的用户，可以打开私聊窗口进行信息发送。

2. 环境

2.1. 运行环境

- 开发环境 windows11
- 开发语言 python & socket
- 显示环境 16: 10 显示器
- 建议运行配置 R7 5800H 处理器，16G 运行内存

2.2. 源文件列表

- server.py 服务器程序
- client.py 客户端程序
- UI.py Pyqt 图形界面程序
- audio.py 音频支持程序
- image 文件夹 logo、背景、图标文件
- 用户名文件夹 缓存文件

3. 算法

本项目设计基于 tcp 连接，客户机和服务器建立连接，当客户将信息发送给服务器时，服务器进行转发（包括再转发给客户），客户端设置了一个接收消息的序列来进行处理。

UI 设计主要有两个页面，私聊窗口和群聊窗口，分别启动线程进行处理，当消息队列中的目的地址是 all，显示再群聊窗口，当消息队列中目的地址是具体用户，显示在私聊窗口

传输的数据结构为消息长度 {"type": "msglen", "destname": destname, "name": self.username, "len": constlen} 和消息内容 {"type": "msglen", "destname": destname, "name": self.username, "len": constlen} 客户端发送消息，区别不同的消息类型，文本对话信息为 msg，表情信息为 emoji，文件、图片、语音本质都以文件形式传输，相关类型和文件类型相同。发送消息时，先发送消息长度，然后发送消息内容。

客户端接受消息，区别不同的消息类型，文本对话直接显示，表情信息以表情的形式展现，文本、图片、语音先保存在本地再以各自的形式展现出来

服务器首先侦听连接，然后存储用户信息，当收到用户发来的长度信息，接着接受用户发来的数据信息，然后使用消息队列进行转发

详细的功能和参数均以 python 注释的格式标注在 src 的源码中

4. 测试截图

下以用户 666 和 888 和 999 为例进行测试

4.1. 登陆注册

首先进行注册 666

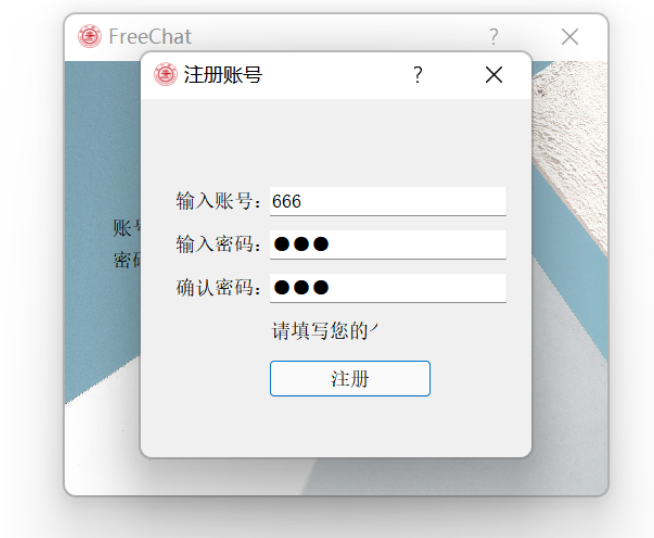


图 1. 注册

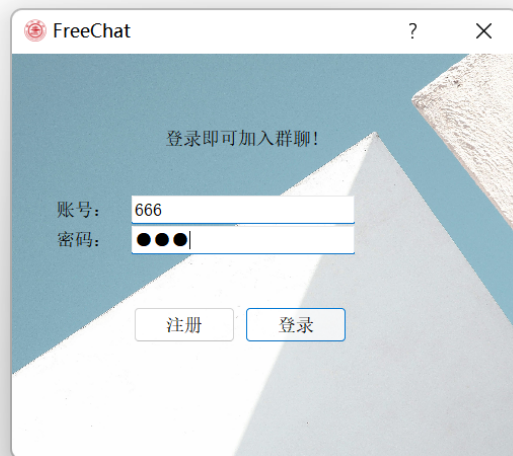


图 2. 登陆

4.2. 发语音

下面以用户 888 和 999 为例测试聊天功能

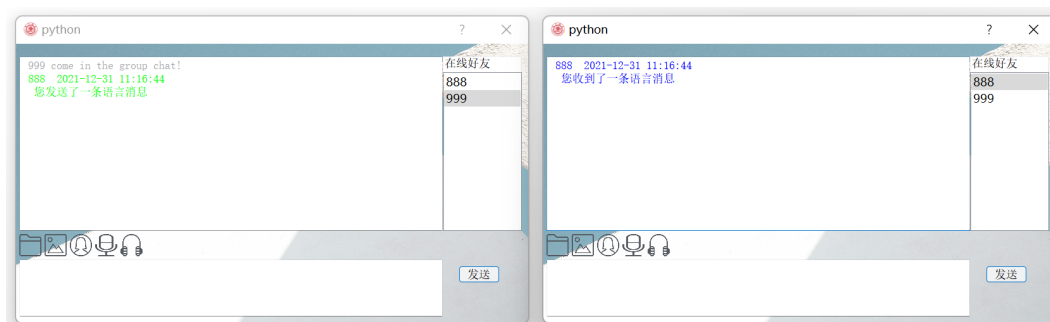


图 3. 发语音

4.3. 发表情

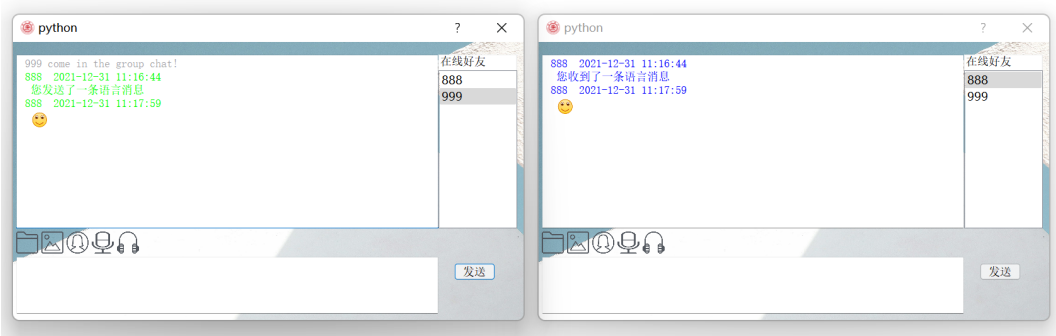


图 4. 发表情

4.4. 发文件

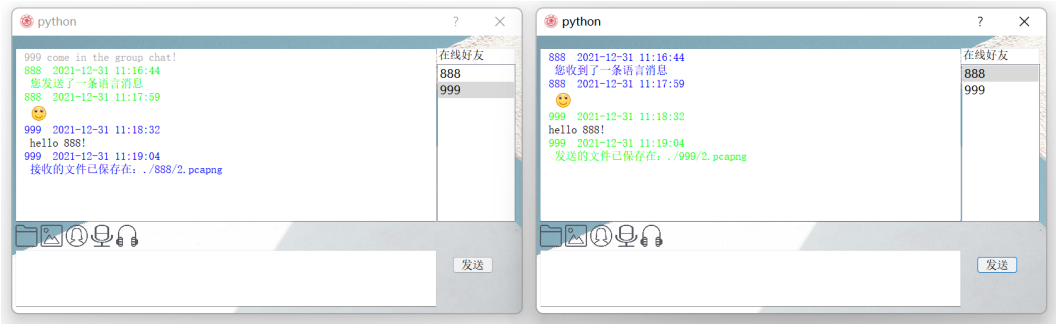


图 5. 发文件

4.5. 发图片

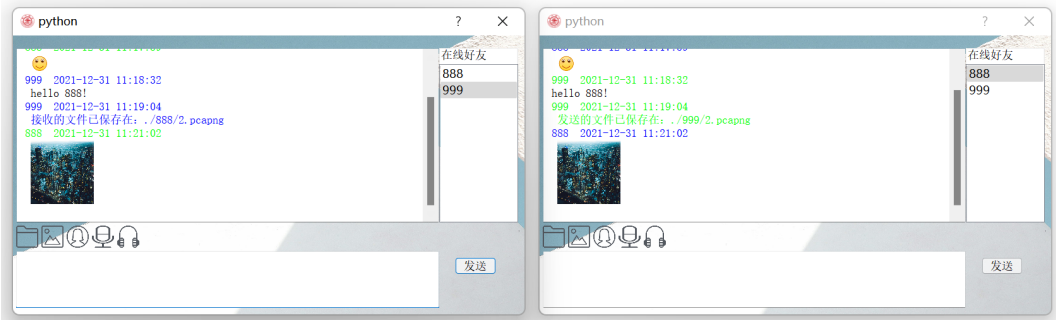


图 6. 发图片

4.6. 私聊发表情

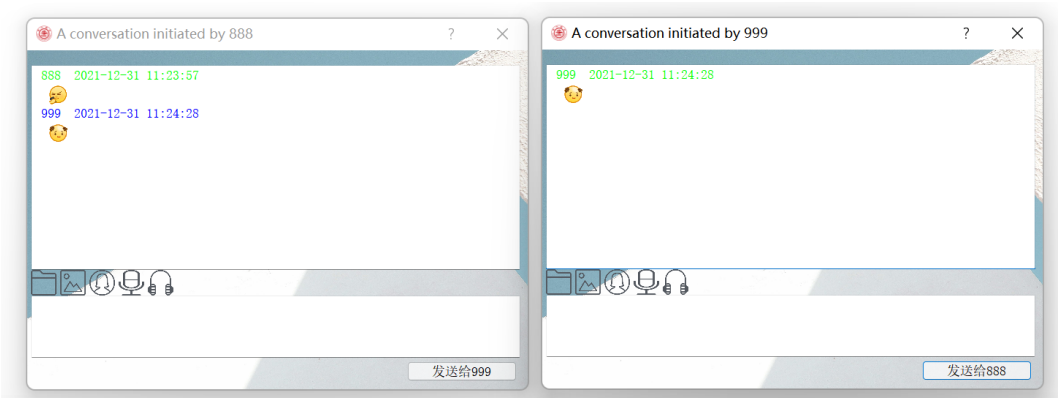


图 7. 私聊发表情

4.7. 私聊发文字和图片

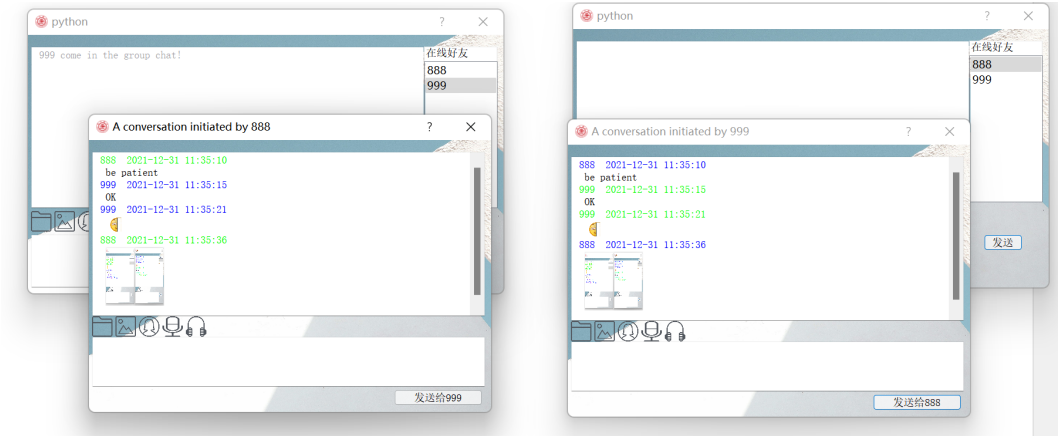


图 8. 私聊发文字和图

4.8. 私聊发语音和文件

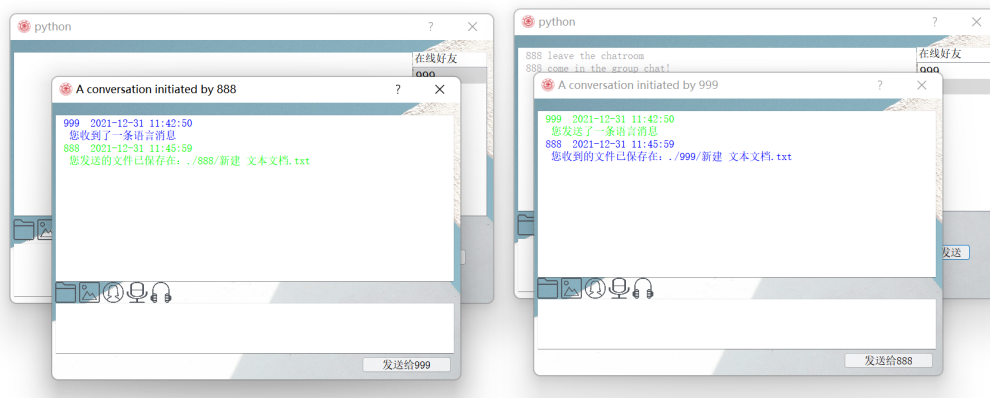


图 9. 私聊发语音和文件

5. 暂未解决的问题

- 服务器和客户端之间小概率丢失信息（服务器发向客户端信息的丢失）
- 交互界面中信息发送者和时间信息概率性显示异常，需选中或点击才能完全显示，问题在于子线程不能调用主线程的 Gui，此问题受限于时间问题暂时未解决
- 优化不足，没有设置 sleeptime，性能占用较大，服务器和客户端同时运行在 R7 5800H 处理器上偶尔会有卡顿
- 文件传输较慢，为了实现较好的传输成功率和效果，文件传输速度不高，导致语音或大文件发送较慢

6. 总结

因为个人对网络编程和软件开发比较感兴趣因此本次大作业我选择开发聊天程序，该项目的主要任务是利用 Python 提供的 socket 调用接口实现网络数据通信，实现了多用户私聊、群聊，发送文本、表情、文件等功能。该项目的难点主要在于协调服务器与客户端之间的通信、服务端多线程处理、客户端 GUI 设计等。通过该项目的实践，我对计算机通信网络中的网络层、传输层通信原理加深了理解。尽管最终开发的程序运行消耗很大、也不太稳定、有未解决的 bug，但是我仍然有很大的收获：我学习了 Python 的 socket 库和 threading 库，引入了多线程和线程管理、消息队列、分片和格式化等思想到项目实现中，这些概念之前从未接触过，通过本次项目的实践进行充分学习和掌握，UI 部分的调整也让我尝试了 PyQt 库的使用，因为这一步需要反复调节各组件的参数以及调试各种错误，所以这部分是我耗时最长的一步。但是，在发现问题、寻求解决方案的过程中，我查阅大量资料，与同学沟通，在过程中不断搭建让我收获颇丰。此次实验提高了我的编程实战能力，并且第一次接触了 GUI 编程。另外，我对操作系统的知识也有了更多的了解。总之，通过此次大作业，我获益良多。非常感谢老师和同学提供的帮助和指导！