一． 概述

学号：20171003994

班级：111172

姓名：吴龙永

创新点

a) 按照 OOP 的思想，将 GUI 与窗口间的主逻辑控制分开，做到了一部分的程序解耦。
b) 通过对 GUI 的逻辑设置，在本地实现了对用户一部分非法操作的规避。减少了对服务器的请求次数。
c) 在使用较为流行的 wxPython 可视化库，继承了 wx.Frame wx.Dialog 等父类，使代码更具可读性。
d) 使用了 wxFormBuilder 进行了辅助开发，效率更高
e) 使用了 Python 的多线程，没有调用系统级方法，可以跨平台运行。
f) 完善了好友列表需求，从单个好友的“按钮”查询升级为，以 Excel 为载体的通过文本框输入姓名即可创建聊条的一体式交互方式。
g) 在 f 的基础上，做到了可以一次性查询所有同学发给本机的未读消息，由于协议问题，只能做到对服务器进行暴力询问。
h) 通过端口随机生成可以实现多客户端在同一台主机上运行。

二． 挑战与困难

需求太多不知从何下手

刚拿到作业发现需求多到读完一遍不知道应该从哪一个需求开始蚕食。于是我用笔开始画用户的交互流程，对照着 QQ。参照协议记下一些不可能发生的越权事件。终于颅内搭建出了一个 FakeQQ 的大致框架。
并且粗略的用代码对协议与自己的构思进行了测试，无误后才开始开发。

GUI 零基础不知如何入门

于是上知乎查找关于 Python 的 GUI 开发相关的经验，发现 Python 的 GUI 开发大致有三个

方向，分别是 Tkinter，wxPython 和 PyQt。最后我选择了用 wxPython 进行开发。
中文的参考文档严重不足又成了一个问题，后来我找到了可以用于可视化布局的 wxFormBuilder。

mainLoop 事件机制不熟悉

第一次登录窗口做完后发现不知如何在登录窗口销毁后打开第二窗口。于是我 Google 了 mainLoop 的文档。

> So, why don't you need to call this interactively? That's just a convenience, because otherwise it would be impossible to enter any commands once you call `mainloop` since `mainloop` runs until the main window is destroyed.

多线程阻塞了图形界面初始化

第一次写多线程的时候，线程一开启直接导致图形界面崩溃。各种写法都尝试过了，花了四个多小时也没知道应该怎么做，直到我发现自己多打了一双括号。显然，多加的两个括号会导致指向函数的线程变为函数的调用，多线程退化成单线程。以后用 IDE 的代码补全一定要注意。

关于用户交互逻辑的迷思

按钮的创建是静态的，测试起来很麻烦，发布出去也不具有普适性，那这个设计就是很失败的。但是服务器并没有好友协议来支持我动态的扩充好友列表，该怎么解决呢。经过缜密思考，既然全年级都是可以互相通信的，那我何不直接用本地的学号查询来建立聊天窗口呢。于是，我建立了一张具有全年级姓名与学号信息的 xlsx。

我如何知道是否有我不熟悉的同学发消息呢

于是我就在好友列表对服务器进行了暴力请求，如果有未读就自动的新建窗口。用了一个列表来托管开启的窗口，避免对一个用户开启多个聊天窗口。

三．　得到的帮助与资料推荐

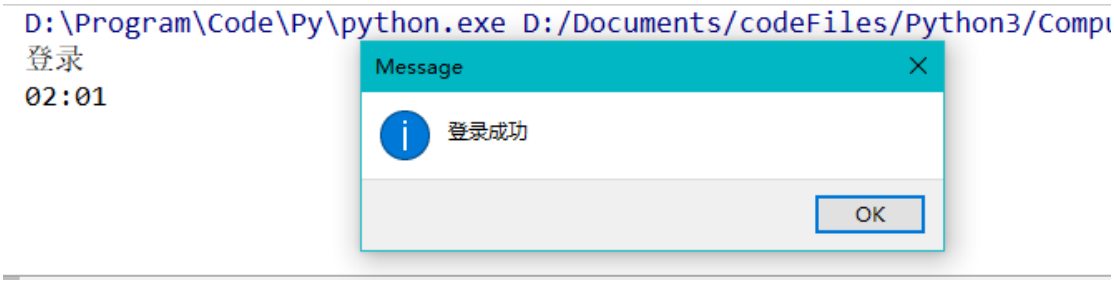
推荐资料

菜鸟教程
廖雪峰
Bilibili

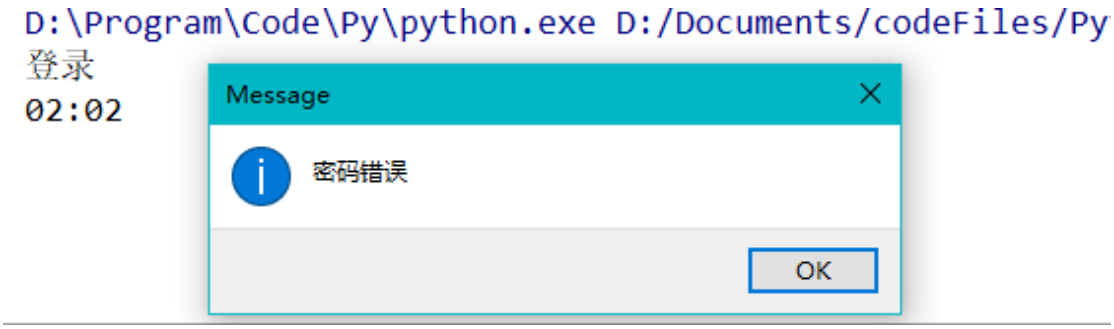
四 . 测试文档与成果展示

02-用户登录

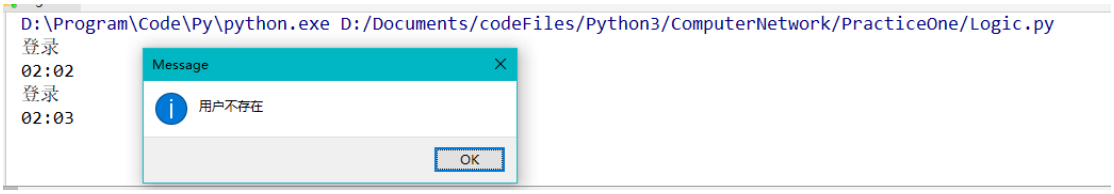
登录成功



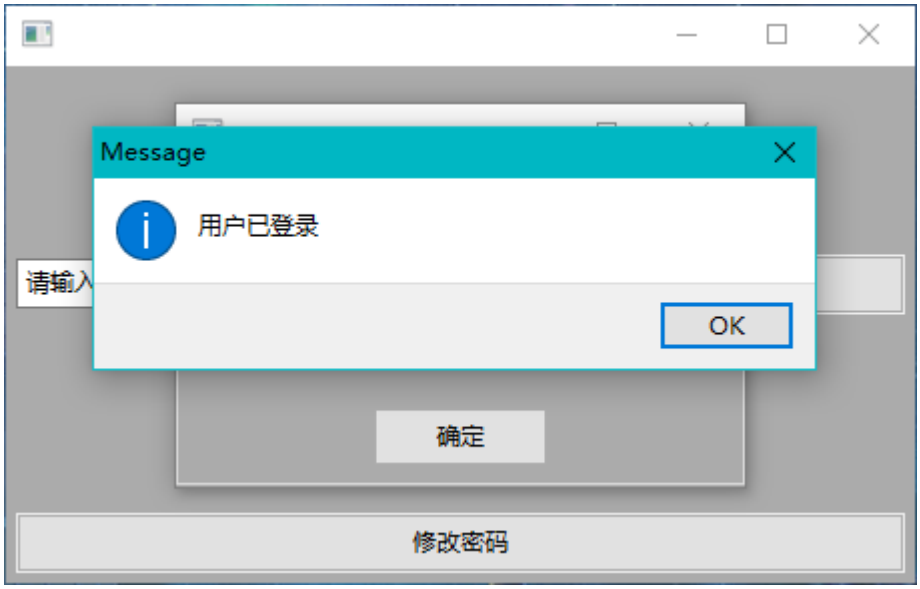
密码错误



用户不存在

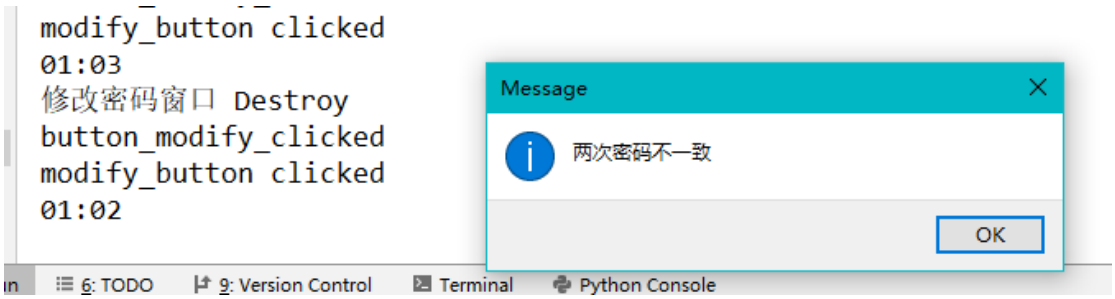


用户已登陆

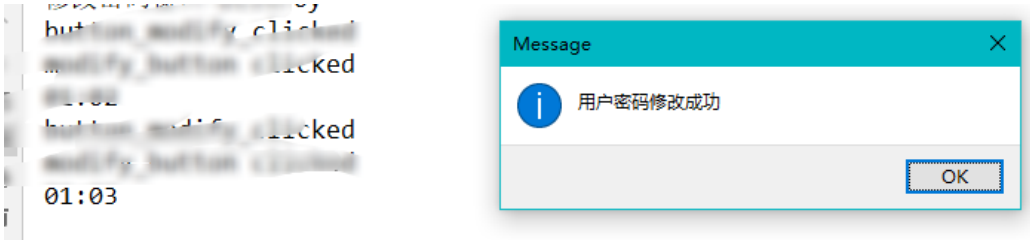


01-修改密码

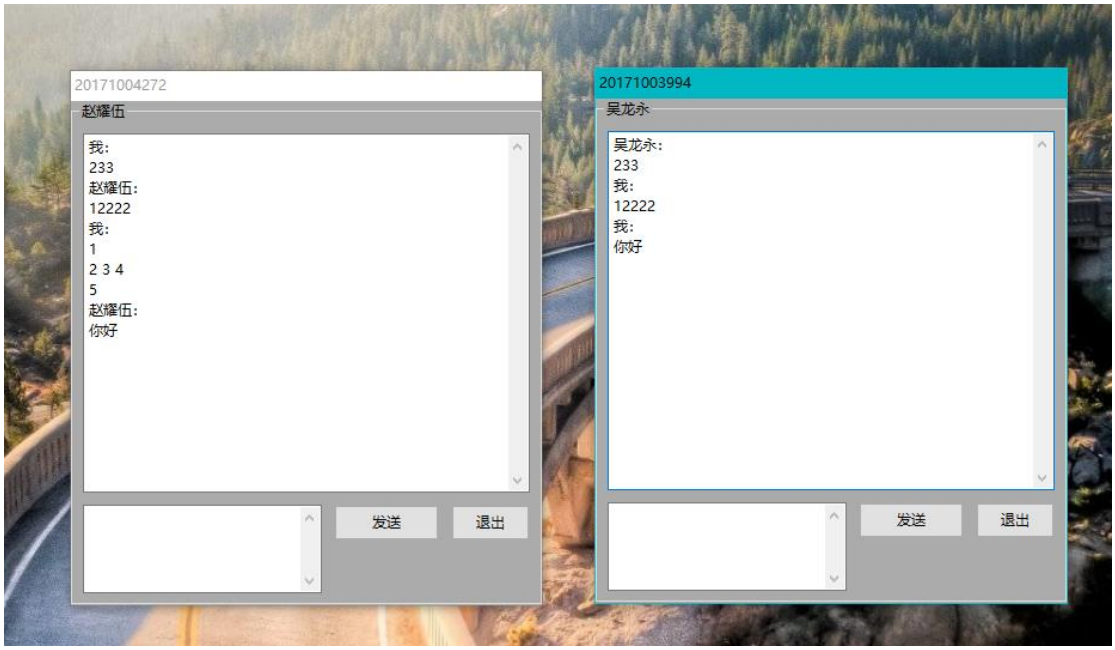
两次密码不一致



密码修改成功



0X-发送消息



注:

其他的因为 GUI 的逻辑设置，不可能出现违法操作，还有对输入文本的检查，也列在了代码之中，不一一细说。