## 文件说明：

客户端代码：（py文件）

main.py 客户端主类

client\_gui.py 客户端界面类（使用Tkinter开发的简单聊天界面）

Connect\_Server.py 连接服务器类（包含创建套接字，发起连接，关闭连接）

配置文件

server\_config.ini 包含服务器IP地址和端口号两个配置项

客户端启动后从该文件读取连接服务器所需要的信息

## 代码类关系说明：

### main.py:

**class client**:  
 **def** \_\_init\_\_(self):  
 self.conn\_sock = Connect\_Server.Connect\_S(clientp = self) # 连接服务器的类  
 self.gui = Login\_application(clientp = self)  
 self.text = None  
 self.stop\_flag = None # 停止标志  
 self.main\_app = None # 客户端大厅主界面引用  
 self.room\_app = {} # 客户端打开的房间引用字典 key 为房间名称 value 为房间引用  
 self.person\_app = {} # 客户端打开的私人聊天界面引用字典 key 为对方名称 value 为窗口引用  
 self.start\_time = None  
  
 **def show**(self):   
 self.gui.mainloop()  
  
 **def close**(self):  
 self.gui.quit()

#### client主类

包含：

一个连接类conn\_sock

客户端大厅主界面的指针（引用）main\_app

客户端打开的房间界面的指针 room\_app

客户端打开的私聊界面指针 person\_app

stop\_flag标识用来在登陆时根据服务端反馈信息看是否应该打开主界面

当服务端返回标识为0时客户端生成大厅主界面，否则给出对应错误提示

### Connect\_Server.py

#### Connect\_S类：

包含主类Client的一个引用

**class Connect\_S**:  
 **def** \_\_init\_\_(self,clientp = None):  
 self.server\_ip,self.port = get\_config('server\_config.ini')  
 self.flag = True  
 self.client = clientp # 客户端主类引用  
 self.s = None  
  
 **def connection**(self): # 发起服务端连接  
  
 **def send\_mess**(self,mess\_type,action\_type,str\_mess):

# 客户端发送消息 mess\_type表示消息类型 action\_type表示动作类型 str\_mess 表示消息内容  
  
 **def mess\_process**(self,mess\_recv,head\_str): # recv\_mess方法调用 **def recv\_mess**(self): # 发起连接后生成一个线程，专门接收服务端消息  
   
 **def close\_connection**(self): # 关闭服务端连接

### client\_gui.py

该文件有较多界面类，作业只注重服务端，这里只举例说明

#### Main\_Room\_application类：

该类是负责绘制客户端大厅的界面类

**class Main\_Room\_application**(Tkinter.Frame):  
  
 **def** \_\_init\_\_(self, master=None,clientp=None,username = None):  
 Tkinter.Frame.\_\_init\_\_(self, master)

**def createWidgets**(self): # 绘制界面

**def entry\_room**(self): # 点击进入房间按钮调用  
  
 **def create\_room**(self): # 创建房间  
 **def send\_mess**(self): # 发送大厅消息

**def quick\_send\_mess**(self,event): # 快捷键发送消息

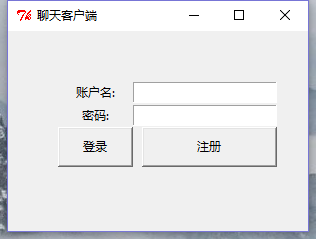
**def create\_person\_chat**(self,event): # 双击在线用户弹出私聊界面  
   
 **def back\_login**(self): # 退出登录

## 3．使用说明：

在python环境下运行main.py文件即可（首先打开服务端, 配置好服务端信息）

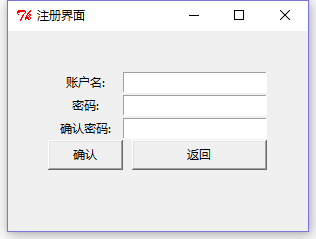
双击main.exe(我已经通过pyinstaller 打包好,以便在没有python环境时运行)

### 开始：

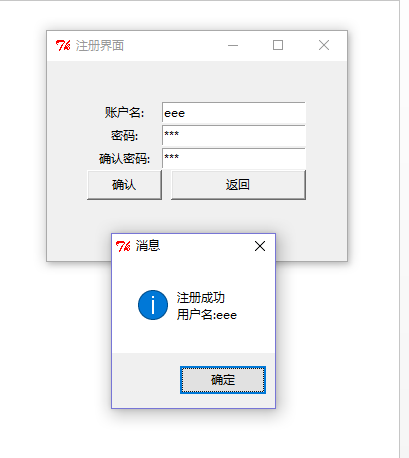


### 注册：

点击注册按钮，后注册信息



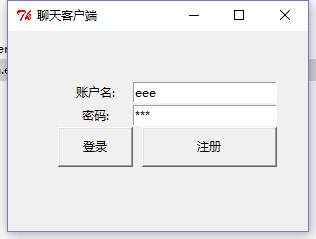
注册成功：



点击返回登录即可

### 登录使用：

输入注册信息，点击登录按钮（若出现无法连接服务器提示，说明服务端有问题）



登录后：

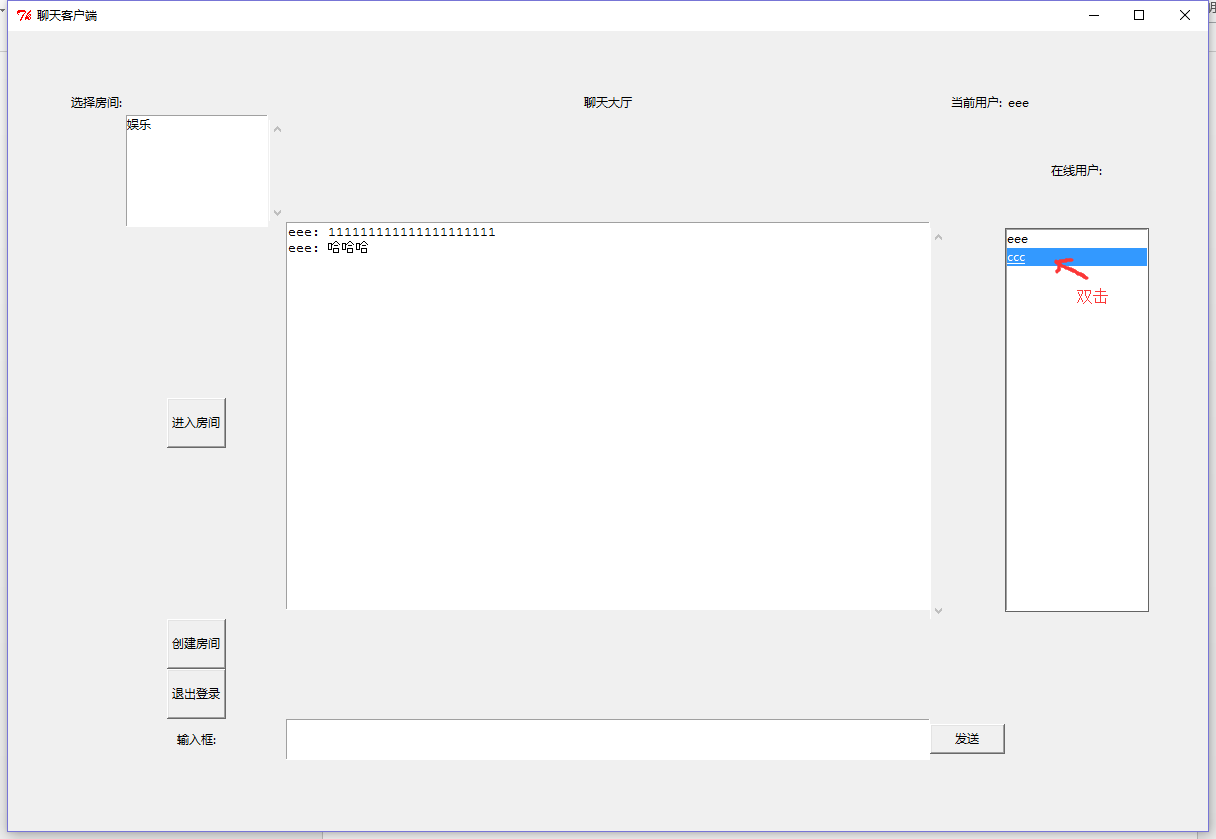


#### 大厅聊天：

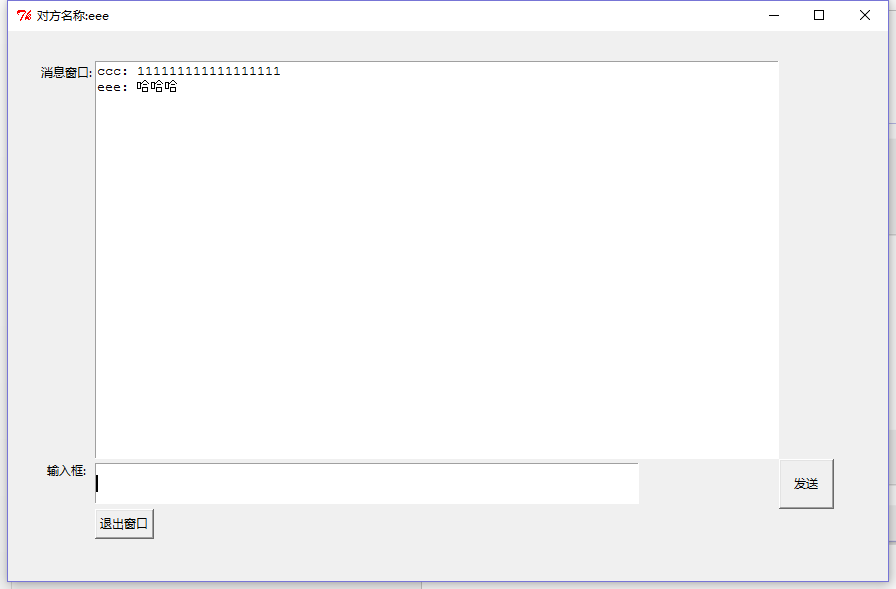
登录后是大厅主界面，在输入框输入后点发送（ENTER）后大厅所有玩家都将看到

#### 私人聊天：

右边的列表栏是当前服务器所有玩家信息，双击玩家名字，可以打开私聊界面



私聊界面如下：



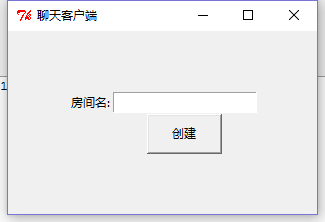
输入信息点击发送后如果对方用户没有打开与你的聊天窗口，将会自动弹出与你的私聊界面

然后显示您发送的信息

### 房间会话：

#### 创建房间：

点击创建房间按钮（注意不能创建已经有的房间名称）



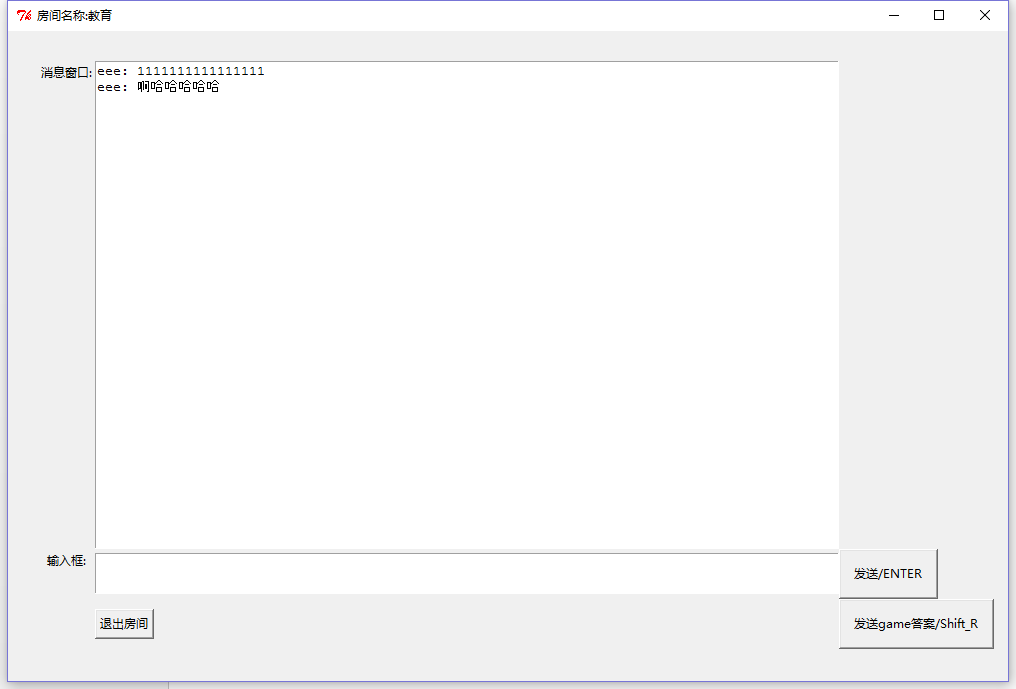
输入名称后点击创建即可；

#### 进入房间：



#### 房间聊天：

下面就是进入房间的界面，有两个按钮，一个是发送消息，一个是发送游戏答案，都有快捷键，一个是ENTER回车, 一个SHIFT\_R（右边的）



#### 21点游戏：

进入房间后，房间每隔30秒出现一个21点游戏题目，（作业要求半点，但是不好测试，等待时间长，改为半分测试）

对于不合法的游戏答案（没有使用给定的数字或者没有全部使用，超过21，或者有

非法字符，都会认为是无效回答）

玩家有20秒时间回答（15秒感觉太短），答案在聊天框输入，后选中相应按钮发送答案

20秒到之后选出胜利玩家，服务器给出消息。



#### 退出房间：

如果想退出，点击退出房间按钮或者关闭窗口即可