

使用说明

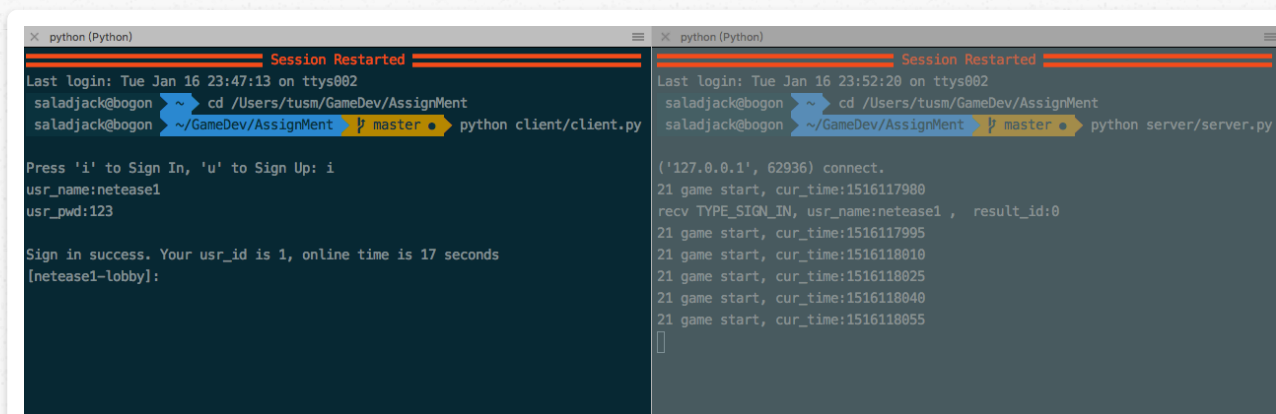
1.必须先运行服务端程序 `server.py`

```
$ python server.py
```

2.之后运行 `client.py`

```
$ python client.py
```

3.客户端输入 `i` 登录, 输入 `u` 注册



```
python (Python) Session Restarted
Last login: Tue Jan 16 23:47:13 on ttys002
saladjack@bogon ~$ cd /Users/tusm/GameDev/AssignMent
saladjack@bogon ~$ python client/client.py

Press 'i' to Sign In, 'u' to Sign Up: i
usr_name:netease1
usr_pwd:123

Sign in success. Your usr_id is 1, online time is 17 seconds
[netease1-lobby]:

python (Python) Session Restarted
Last login: Tue Jan 16 23:52:20 on ttys002
saladjack@bogon ~$ cd /Users/tusm/GameDev/AssignMent
saladjack@bogon ~$ python server/server.py

('127.0.0.1', 62936) connect.
21 game start, cur_time:1516117980
recv TYPE_SIGN_IN, usr_name:netease1, result_id:0
21 game start, cur_time:1516117995
21 game start, cur_time:1516118010
21 game start, cur_time:1516118025
21 game start, cur_time:1516118040
21 game start, cur_time:1516118055
[]
```

4.登录成功后, 用户会默认进入大厅(lobby), 用户可直接在大厅里聊天, 也可以输入特定的**命令**来进行其它的操作, 具体操作方式如下:

| | |
|---------------------------------------|----------------------------|
| \$ [netease1-lobby]:/create room | # 创建并进入房间 |
| \$ [netease1-lobby]:/enter room 1 | # 进入1号房间 |
| \$ [netease1-room-1]:/quite room | # 退出房间 |
| \$ [netease1-room-1]:/21game 4+5+6+6 | # 参加21点游戏, 提交的答案为: 4+5+6+6 |
| \$ [netease1-lobby]:/chat to netease2 | # 与用户netease2私聊 |
| \$ [netease1-private]:/chat quit | # 退出私聊 |
| \$ [netease1-lobby]:/sign out | # 注销 |

架构说明

- 1.C/S端各有一份协议 `p4c/p4s`，协议在传输时会序列化成 `JSON` 格式
2. `server/db` 存储了所有已有用户的信息，文件每一行的数据格式为

用户名 密码 用户ID 最近一次登录的时间戳(秒) 总登录时长(秒)

- 3.考虑到在Python中，Windows的 `select()` 方法只能接收 `socket` 的输入流，而不像Linux还能接收 `sys.stdin` 的输入流。所以为了使 `sys.stdin` 不会阻塞接收信息，在Client端收、发逻辑要处于不同的线程

4.架构示意图

