

1 Program Structure

To execute my program, type “python3 run.py”.

1.1 Environment

- OS: Arch Linux (4.18.7)
- IRC server: Irssi (1.1.1)
- Programming language: Python (3.7)
- Non-standard Python packages: BeautifulSoup4 (4.6.3)

1.2 File Description

The whole program is in “run.py”. Thus, the followings describes the content of “run.py”.

1.2.1 Configuration

- user_name = 'Bot'
- host_name = 'NTU'
- server_name = 'ntu.edu.tw'
- real_name = 'Fan-Keng Sun'
- nick_name = 'bot_b03901056'
- channel = '#CN_DEMO'
- lower_bound, upper_bound = 1, 10 (lower and upper bound for number guessing)

1.2.2 IRC Client Class

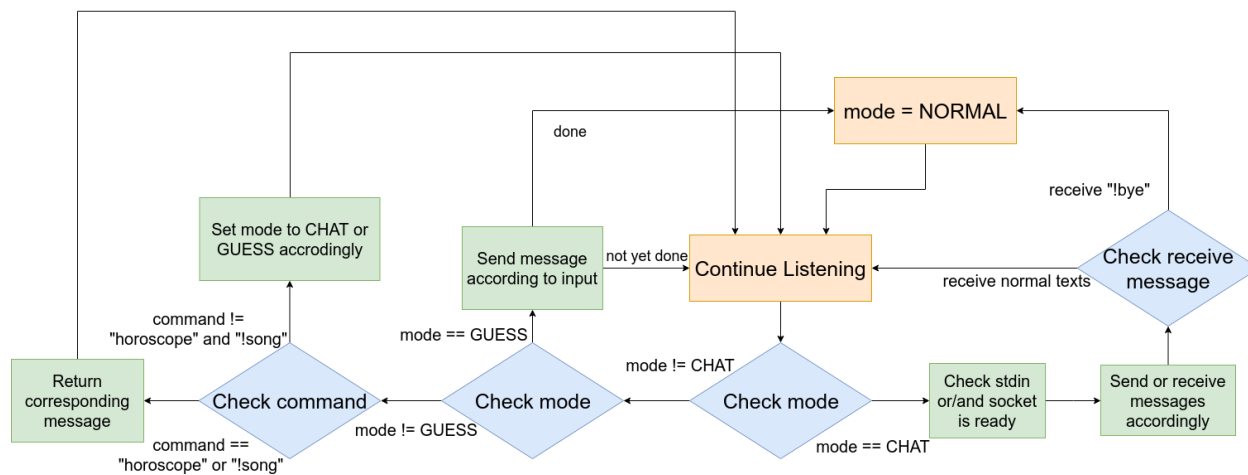
A class that represents the IRC client. Functions in this class includes:

- **__init__()**: Initialize socket (`socket.socket(socket.AF_INET, socket.SOCK_STREAM)`).
- **connect(host, port)**: Connect to a the port of the host (`socket.connect((host, port))`).
- **send(message)**: Send the message via the socket (`socket.send(msg.encode())`).
- **join(channel)**: Join the channel (`send('JOIN'.format(channel_name))`).
- **get()**: Receive message from socket (`socket.recv(1024).decode('utf-8')`).
- **user(user_name, host_name, server_name, real_name, nick_name)**: Set the names and nick name.
- **is_ping(text)**: Return *True* if *text* is *PING*, else *False*.
- **priv_msg(channel, message)**: Send a PRIVMSG message to channel. (`send('PRIVMSG :'.format(channel, msg))`).

1.2.3 Global Functions

- **get_soup(url)**: Get the BeautifulSoup of *url*.
- **get_daily_horoscope(horoscope)**: Get the daily horoscope of *horoscope* by web crawling.
- **get_song_url(song_name)**: Get the url of *song_name* by web crawling.

1.3 Program Flow



2 Challenges & Solutions

2.1 Unable to connect to Freenode by personal hotspot

The first I tried to connect to Freenode by Irssi, I used my personal hotspot to access the Internet, since the public wifi (NTU, ntu-peap) is unstable at my location. However, I found that Irssi can't successfully authenticate my connection. After several tries, I decided to change a location and use ntu-peap. Then, everything worked as expected.

2.2 No response

Other client has no response to my message no matter what I sent. After browsing on-line, I found that I forgot to add an EOL in my message.

2.3 Cannot determine the type of commands

I want to use "str.endswith" to detect the type of commands. Thus, I need to remove the trailing newline characters at the end of the received messages.

2.4 Simultaneously receiving and typing text

To realize the "!chat" command, I use the linux "select" command via python, so that the program can simultaneously receiving and typing text. Also, I have to print the backspace character "\b" to erase the ">" when text was received.

3 Reflections

This is a very interesting assignments because it realizes the procedure of chatting online, which is heavily used nowadays after the appearance of social networks, such as Facebook, Line. Additionally, I think it would be more interesting if students can work in group to implement a more sophisticated chatting system.