Documentazione del sistema client-server di chatroom

Questo sistema client-server di chat room è implementato utilizzando la programmazione socket Python 3.9.0 e tkinter per la GUI.

Il server gestisce più client contemporaneamente, consentendo agli utenti di inviare e ricevere messaggi in una chat condivisa.

L'applicazione client consente agli utenti di connettersi al server, inviare messaggi alla chat e ricevere messaggi da altri utenti, tutto ciò avendo sempre una lista di tutti i nickname collegati alla chat in tempo reale.

Requisiti

Si consiglia di scaricare la cartella del progetto dal link di github così da poter eseguire i due file direttamente con un doppio click su quest'ultimi.
In alternativa bisogna aprire i file e eseguire manualmente i file da un propio terminale python oppure attraverso Visual-Studio-Code.
Per creare questo sistema client-server di chat abbiamo usato Python ed alcune sue librerie:

- `socket`
- `threading`
- `tkinter`
- `simpledialog(tkinter)`

Code Explanation

Server

Il codice del server crea un socket e lo associa a un IP e una porta specifici, mantenendo anche un elenco di client connessi e il loro nickname.

Resta in ascolto delle connessioni in entrata e genera un nuovo thread per gestire ciascuna connessione client.

Funzioni principali del Server:

- `handle_client()`: Riceve e trasmette messaggi da un client.
- `broadcast()`: Invia messaggi a tutti i client connessi tranne il mittente.

- `send_nicknames()`: Invia l'elenco dei nickname collegati a tutti i client.
- `remove_client()`: Rimuove un client dall'elenco e chiude la connessione.

Client

Il codice client chiede l'inserimento di un nickname con il quale collegarsi alla chat, successivamente crea una GUI utilizzando tkinter e si connette al server.

Gestisce l'invio e la ricezione di messaggi e aggiorna di conseguenza gli elenchi di chat e nickname.

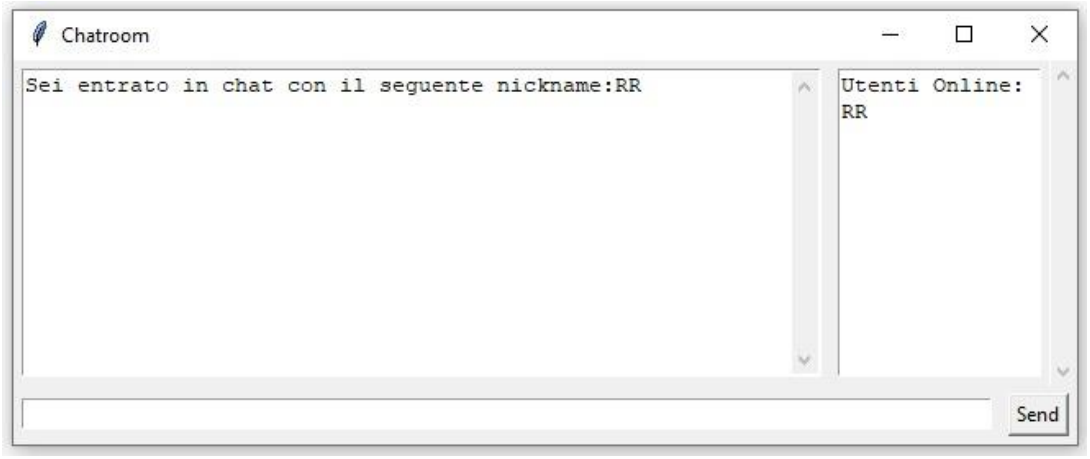
Funzioni principali del Client:

- `receive_messages()`: Ascolta continuamente i messaggi dal server.
- `send_message()`: Invia un messaggio al server.
- `update_nickname_list()`: Aggiorna l'elenco dei nickname collegati.
- `on_closing()`: Gestisce la disconnessione del client.

Descrizione GUI

L'applicazione client presenta un'interfaccia utente grafica (GUI) creata utilizzando tkinter e così composta:

- Chat Frame: dove verrà visualizzato tutto ciò che riguarda la chat.
  Contiene un widget di testo (`chat_box`) per mostrare i messaggi e una barra di scorrimento (`chat_scrollbar`) per navigare nella chat.
- Nickname Frame: visualizza l'elenco degli utenti connessi.
  Contiene un widget di testo (`nickname_box`) per mostrare i nickname e una barra di scorrimento (`nickname_scrollbar`) per navigare nell'elenco.
- Input Frame: consente all'utente di digitare e inviare messaggi.
  Contiene un widget di voce (`input_field`) per l'input del messaggio e un pulsante (`send_button`) per inviare il messaggio.



Passaggi per chattare con altri utenti

1. Avvia il Server: Esegui con doppio click il file ‘Server.py’.
2. Avvia il Client: Esegui con doppio click il file ‘Client.py’. Al momento del lancio, il client ti chiederà di inserire un nickname. Questo passaggio è da ripetere per quanti client si vuole avere collegati.
3. Inserisci Nickname: Inserisci il nickname desiderato. Questo nickname sarà visualizzato dagli altri utenti nella chat.
4. Invia Messaggi: Digita il tuo messaggio nel campo di input e premi "Invio" o clicca sul pulsante "Invia" per inviare il messaggio.
5. Visualizza Messaggi: Tutti i messaggi inviati da te e dagli altri utenti saranno visualizzati nella casella di chat.
6. Visualizza Utenti Connessi: L'elenco degli utenti connessi sarà visualizzato nella casella dei nickname.

Considerazioni Finali

- Gestione delle eccezioni di Connessione: Entrambi i codici, sia del server che del client, ora gestiscono le eccezioni comuni relative alla connessione in modo da evitare crash imprevisti e migliorare la robustezza del sistema.
- Chiusura Sicura: Quando il client si disconnette, invia un messaggio al server per informarlo e chiude la connessione in modo sicuro.
- Logging degli Errori: Gli errori vengono stampati sulla console per facilitare il debug, ma è possibile estendere questa funzionalità per registrare gli errori in un file di log.
- Mentre attualmente non sono presenti controlli nei nickname e non ci sono controlli su cosà possa essere scritto nella chat.