

Relazione Progetto

Traccia 3 – Chat game

Zacconi Andrea

0000942346

Incarnato Luigi

0000934392

June 2021

0.1 Introduzione

Il progetto consiste nel realizzare un programma con il linguaggio Python che permetta una connessione client-server sfruttando il protocollo TCP-IP.

0.2 Descrizione

Abbiamo realizzato due differenti parti nel progetto:

- **game_gerver.py**

Il server a cui si connettono i client per giocare e inviare gli eventuali messaggi.

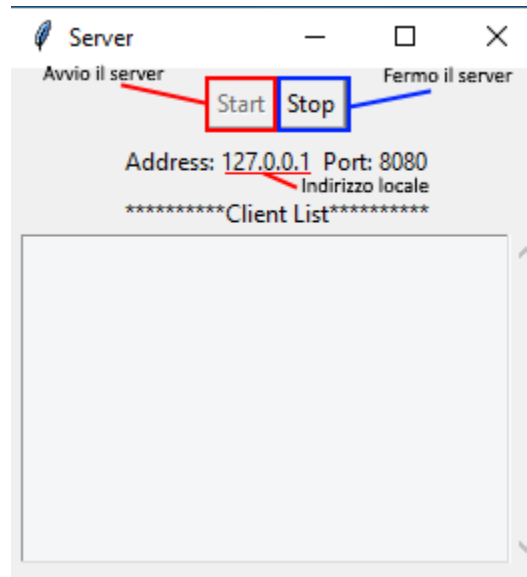
- **project_clinet.py**

Il/I client che si connettono al server per inviare messaggi ad esso.

Abbiamo anche un file chiamato **Domande.py** dove vengono racchiuse tutte le domande e soluzioni del gioco in piu' Dictionary (la prima risposta per ogni Dictionary e' quella corretta).

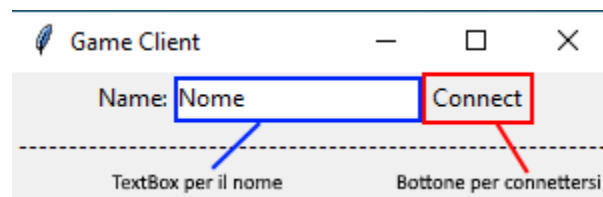
E' necessario aver installato sul proprio dispositivo l'IDE Spider o un qualsiasi compilatore di codice Python 3.

Una volta eseguito il server, questo rimarra in attesa di eventuali connessioni sull'indirizzo locale 127.0.0.1, precisamente attende il collegamento di 2 client per l'avvio del gioco.

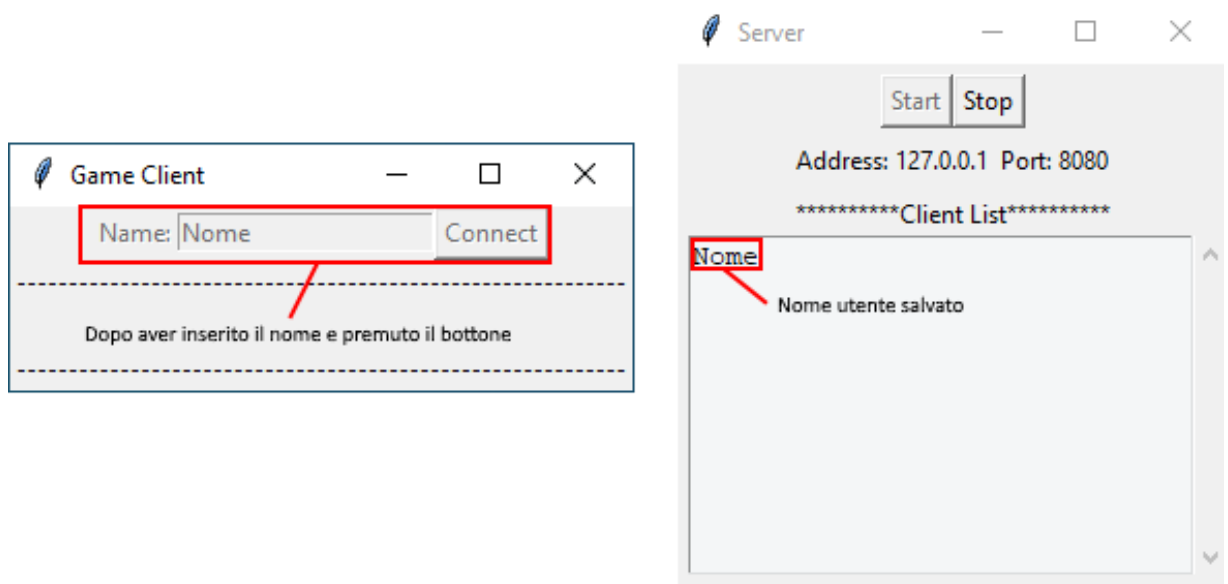


Nel momento che avviamo il client esso presentera' 2 elementi:

- Una **TextBox** per l'inserimento del nome.
- Il bottone **Connect** per connettersi al server.



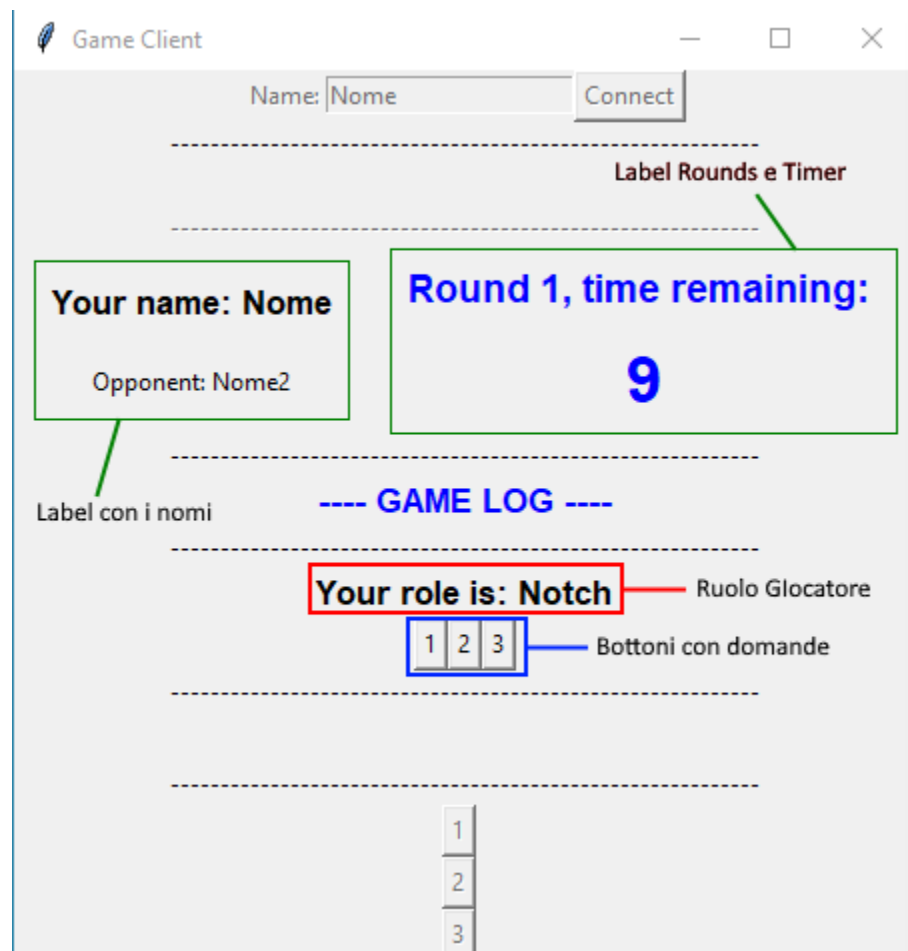
Dopo l'inserimento di uno dei 2 nomi da parte dei clients, il server lo salvera' al suo interno e restera' in attesa del secondo client.



Una volta ricevuti entrambi i nomi, il server salvera' anche quest'ultimo e avviera' la partita.

Una volta iniziata la partita, si genereranno nuovi elementi grafici:

- Una **Label** contenente i nomi dei rispettivi giocatori.
- Una seconda **Label** contenente le informazioni del Round attuale e un **Timer** per indicare il tempo a disposizione.
- Una **Label** con il ruolo assegnato al giocatore.
- Una linea orizzontale di **Button**, questi rappresentano le 3 domande dipendenti dal ruolo assegnato, una di queste e' la domanda a trabocchetto che fara' perdere la partita chiudendo la connessione col server.



Se il player riesce a non prendere la domanda a trabocchetto, la GUI di esso si aggiornerà nuovamente facendo comparire:

- Una **Label** contenente la domanda effettiva a cui il player dovrà rispondere.
- I **Buttons** in linea verticale si sbloccheranno e a fianco ad essi le rispettive risposte compariranno.

Game Client

Name: sdf Connect

Your name: sdf
Opponent: sdf3

Round 1, time remaining:
6

---- GAME LOG ----

Your role is: Notch

1 2 3

Quale tra questi è definito come gioco MMORPG?

Domanda

1 World of WarCraft
2 League of Legends
3 Paladins

Risposte

Scelta una risposta tra quelle generate, il player dovrà aspettare la fine del timer per poi passare al Round successivo.

Il gioco termina al Round 5, concluso l'ultimo round il player ha a disposizione altri 15 secondi per vedere il resoconto della partita, ovvero se ha vinto o perso. Alla fine di questi ultimi 15 secondi il gioco ripartirà da capo e si potrà giocare ancora.

0.4 Librerie utilizzate

- Tkinter
- Random
- Socket
- Time
- Threading

In caso di problemi con la cartella compressa, è possibile visionare il progetto al seguente link:

https://github.com/LuKe2Ink/Progetto_Programmazione_Reti