**Relazione Progetto** Traccia 3 – CHATGAME . Matteo Vanni

935584

**INDICE**

1. – Introduzione
2. – Descrizione
3. – Dettagli implementativi 4 – Librerie utilizzate

# – INTRODUZIONE

Il progetto implementa un’architettura client-server per il supporto di un Multiplayer Playing game testuale in rete locale con interfaccia grafica.

# 2– DESCRIZIONE

Ho sfruttato il principio della chat per realizzare un gioco a domande e risposte.

I giocatori sono accolti dal master (server\_gui.py) che richiederà

l’inserimento del nome e dell’ip per collegarsi, assegnando un ruolo casuale ad ogni giocatore.

Quando viene premuto il bottone “Connetti” la grafica verrà mostrata e il gioco attenderà 3 secondi prima di partire con il timer

A questo punto si dovrà scegliere una delle tre porte. Due di esse conducono a una domanda mentre una l’altra ad un trabocchetto che farà uscire dal gioco. Se si risponde correttamente alla domanda verrà aggiunto un punto, in caso contrario verrà tolto.

Durante il gioco il punteggio sarà visibile poco sopra dove verranno generate le domande

Le domande sono una serie di domande di cultura generale.

Il gioco può terminare o allo scadere del tempo, la “botola” o premerà il bottone di disconnessione, che in ogni casoi mostreranno il punteggio finale e un messaggio

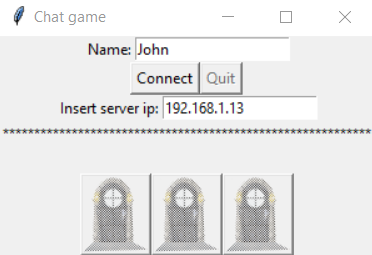
Il timer di gioco è un thread che lavora in background in modo da consentire l’esecuzione del gioco

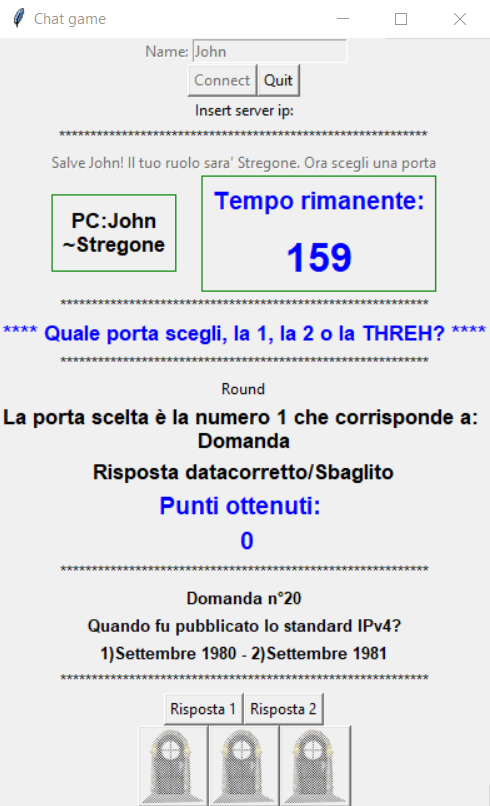
*Scelte effettuate:*

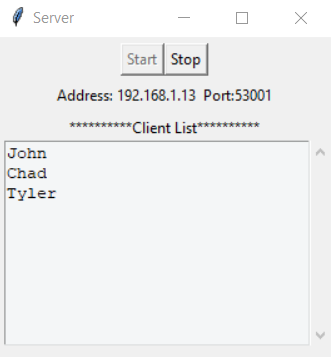
* Utilizzo dei dizionari per controllare in maniera più efficiente i dati, essendo i dizionari pensati per gestire coppie di elementi chiave- valore.
* Uso di label per la comparsa delle domande e delle risposte per una più immediata visualizzazione
* Gestione dell’interazione delle domande tramite bottoni per semplificare l’inserimento dei dati da parte del giocatore.
* È stata scelto una durata per il gioco di 3 minuti

La grafica utilizzata è stata implementata usando come base il laboratorio 12 – RockPaperScissor game

Qui di seguito la grafica di gioco:

(GUI di log)

(GUI del gioco avviato)

(GUI del server con una lista di utenti

# – DETTAGLI IMPLEMENTATIVI

Ho implementato una connessione Client-Server utilizzando come porta per la socket la 53001, mentre per la gestione dell'indirizzo IP viene richiesto un inserimento da parte dell'utente dell'IP desiderato, che verrà mostrato nella GUI del server.

Sono state utilizzate le funzioni rcv() e send() della libreria socket per

gestire l’invio dei messaggi con un buffersize di lunghezza 4096 byte.

Ho gestito tutte le eccezioni che si verificano frequentemente con il sistema di try-except quali, ad esempio, l’eccezione che viene generata se si tenta di eseguire il client prima del server o se si prova a far partire il gioco prima di aver inserito nome e ip.

*Strutture utilizzate:*

* + ampio uso delle liste

*Thread utilizzati:*

* + un thread per il server
  + un thread per ogni client
  + un thread per la gestione del timer pre partita
  + un thread in background per le gestione del timer di gioco

# – LIBRERIE UTILIZZATE

* + messagebox, tkinter
  + AF\_INET, socket, SOCK\_STREAM
  + time, sleep
  + threading, Thread
  + random