

# RELAZIONE PROGETTO DI PROGRAMMAZIONE DI RETI

## Traccia 3 - CHATGAME

Lorenzo Zanetti,  
Matricola: 0000933486

### INTRODUZIONE

Il progetto consiste nel realizzare un programma con il linguaggio Python che permetta una connessione client-server sfruttando il protocollo TCP-IP di possibilmente infiniti client che accedono ad una chat comune e possono partecipare ad un gioco.

### DESCRIZIONE

Sono stati realizzati due applicativi python: `game_chat_server.py` e `game_chat_client.py`.

Il primo applicativo permette di avviare il server, all'avvio del file comparirà infatti una GUI con 2 pulsanti "start" e "stop", dove il primo tasto sarà attivo e premibile, mentre il secondo no. Infatti, il nostro server non è ancora stato avviato, per farlo si deve premere il tasto start che diverrà inattivo, rendendo attivo il tasto "stop" in modo che ora il server può solo essere fermato. Nella GUI saranno presenti anche l'indirizzo ip e la porta utilizzati (di default 127.0.0.1 e 8080) e l'elenco dei giocatori collegati tramite client.

Il secondo applicativo permette invece ai giocatori di connettersi al server. Avviandolo verranno richiesti a linea di comando ip e porta del server e una volta inseriti il client si connetterà mostrando la sua GUI che consiste in un pannello in cui verrà mostrata la chat, una casella di testo in cui poter scrivere i propri messaggi e un tasto invio per inviarli (in alternativa è possibile premere il tasto invio della tastiera).

Nella chat verrà subito richiesto che il client inserisca un nome, una volta che il nome è inviato, il server gli assegnerà anche un ruolo e salverà questo nuovo giocatore nell'elenco, oltre ad annunciare a tutti gli altri giocatori il suo arrivo.

Una volta registrato il nuovo giocatore ha accesso alla chat dove può parlare con gli altri giocatori o inviare alcuni "comandi speciali". Può infatti scrivere {help} (come gli viene subito consigliato dal server) per vedere tutti i comandi utilizzabili: {play}, {quit} e {record}.

Scrivendo "{quit}" si può infatti lasciare la chat, disconnettendosi dal server, che aggiornerà l'elenco dei client e comunicherà a tutti che il giocatore li ha abbandonati.

Scrivendo "{record}" il server comunicherà al giocatore chi detiene il punteggio più alto e quant'è questo punteggio.

Infine scrivendo "{play}" il giocatore inizierà effettivamente il gioco, che verrà svolto in una nuova finestra.

(tutti i giocatori vengono informati quando qualcuno gioca, sul punteggio raggiunto e sull'eventuale nuovo record raggiunto)

Nella nuova GUI che si aprirà scrivendo {play} si svolgerà il gioco vero e proprio. Vi è un countdown di 3 secondi, poi il gioco inizia, si attivano i tasti in basso a, b, c e inizia il timer del gioco di 40 secondi. Premendo uno dei 3 tasti si può cadere in un “trabocchetto” perdendo la partita immediatamente, mentre con gli altri 2 si prosegue, i tasti si disattivano, appare una domanda ed è possibile rispondere in una casella di testo subito sopra ai 3 tasti. Con una risposta esatta si guadagna un punto, con una sbagliata se ne perde uno e una volta data la risposta verrà di nuovo disattivata la casella di testo e riattivati i tasti a, b, c in modo che si possa provare di nuovo. Al termine del tempo o beccato un “trabocchetto” il gioco finisce, viene inviato il punteggio al server che controlla se è un nuovo record e lo comunica agli altri giocatori. A questo punto la finestra di gioco si chiude ma è comunque disponibile quella della chat da cui si può continuare a scrivere, si ricevono i messaggi ed eventualmente si può giocare di nuovo o ci si può disconnettere.

## DETTAGLI IMPLEMENTATIVI

Il server permette la connessione di teoricamente infiniti client, è infatti “continuamente” in attesa di nuove connessioni. Esso tiene aggiornati tutti i client di ciò che stanno facendo gli altri e gli permette di comunicare fra loro, oltre a questo assegna i ruoli casualmente ai giocatori scelti da una lista, memorizza nomi e ruoli, e mantiene in memoria il punteggio più alto raggiunto e il giocatore che l’ha totalizzato.

Il codice del gioco vero e proprio è nel file utilizzato dal client, che oltre a poter mandare messaggi al server gestisce la logica del gioco, che si ferma allo scadere del tempo o quando si incontra un “trabocchetto”, che gestisce le domande estratte in modo casuale e il controllo delle risposte, il punteggio e invia il risultato finale solo se la connessione al server è ancora presente.

Chiudendo le finestre del server o dei client senza prima aver eseguito la disconnessione lascia i socket aperti, per cui per il server è necessario premere il tasto stop e per i client è necessario usare il comando {quit}, in questo modo non si rischia di lasciare occupata la porta utilizzata anche dopo aver chiuso il server.

## GESTIONE THREAD

All’avvio il server lancia un thread dedicato all’ascolto di nuove connessioni da parte dei client che si chiuderà solo con la chiusura del server. Poi per ogni client che si connette crea un nuovo thread dedicato all’ascolto di quel determinato client.

Mentre per il client vi è il thread principale che si attiva premendo “send” per mandare dei messaggi al server, un secondo thread in ascolto dei messaggi del server che vengono scritti sulla chat e due nuovi thread lanciati ogni volta che si inizia una nuova partita per gestire la GUI del gioco e il countdown. E’ infatti possibile leggere in chat messaggi degli altri giocatori e scriverne anche mentre si sta giocando!

(nel caso in cui ci si disconnetta del server durante una partita questa verrà terminata senza errori)