

# Progetto di Programmazione di Reti

## A.A. 2020/2021

Relazione a cura di

**Fabio Pedrini**

[fabio.pedrini3@studio.unibo.it](mailto:fabio.pedrini3@studio.unibo.it)

0000916427

## Traccia numero 3:

Sfruttando il principio della chat vista a lezione, implementare un'architettura client-server per il supporto di un Multiplayer Playing Game testuale. I giocatori che accedono alla stanza sono accolti dal Master (server) che assegna loro un ruolo e propone loro un menu con tre opzioni, due delle quali celano una domanda mentre la terza è l'opzione trabocchetto. Se sceglie l'opzione trabocchetto viene eliminato dal gioco e quindi esce dalla chat. Se seleziona invece una delle domande e risponde correttamente al quesito acquisisce un punto, in caso contrario perde un punto. Il gioco ha una durata temporale finita; il giocatore che al termine del tempo ha acquisito più punti è il vincitore.

## Introduzione:

Il progetto consiste nel realizzare un Multiplayer Playing Game testuale con il linguaggio *Python* che permetta una connessione client-server sfruttando il protocollo TCP-IP.

## Descrizione:

Per iniziare il gioco, si esegua il file "*chatgame\_server.py*" sia tramite linea di comando sia tramite *Spyder* (ad esempio). Comparirà una finestra con due pulsanti (start e stop): cliccando sul pulsante start il server rimarrà in attesa di client all'indirizzo locale **127.0.0.1** e sulla porta **8080**.

A questo punto si esegue il file "*chatgame\_client.py*" per connettere (almeno) un client al server. Comparirà una finestra in cui andrà inserito il nome e inviato al client, nella parte inferiore sono presenti tre bottoni (A, B e C) ma sono ancora disattivati.

Dopo che il server ha ricevuto il nome del client, la finestra di dialogo si aggiorna con la comparsa di nuovi widget (disattivati, per il momento); successivamente parte un countdown di 5 secondi per avvisare il client che il gioco sta per iniziare. Terminati i 5 secondi, si attivano i pulsanti A, B e C e parte un altro countdown di 30 secondi che determina la durata della partita. Cliccando su uno dei tre pulsanti inferiori, verrà scelta una domanda o un trabocchetto: nel caso si scegliesse il trabocchetto, la partita termina e il risultato viene inviato al server.

Nel caso venisse scelta una domanda, si attiverà in basso il box per scrivere e il pulsante per inviare la risposta. Subito dopo aver risposto, verranno riattivati i

pulsanti A, B e C e si potrà fare un'altra scelta. Quindi, la partita finisce se finisce il tempo (30 secondi) o se si trova un trabocchetto.

## Dettagli implementativi:

- La partita finisce anche se vengono esaurite le domande (in questo caso con 12 domande e 30 secondi è abbastanza difficile che succeda, ma può succedere);
- Le domande sono associate ad un indice che viene generato casualmente: creo un array con numeri da 1 a *numQuestions* (12, in questo caso) e poi permuto gli elementi con la funzione *numpy.random.shuffle()*, per poi leggere gli elementi del vettore in modo sequenziale;
- Nella finestra del server, compare un elenco dei client connessi e vengono cancellati quando la connessione tra client e server viene chiusa (appena finisce una partita)
- Sempre nella finestra del server, cliccando sul pulsante stop vengono stampati sulla console i nomi e i risultati di tutti i client che hanno partecipato al gioco;
- Essendo un gioco multiplayer, si possono connettere più client e giocare contemporaneamente: il server è sempre in ascolto grazie ad un *while True* nella funzione *accept\_clients()*.

## Librerie utilizzate:

- *socket*,
- *threading*,
- *tkinter* per creare le GUI,
- *numpy* per generare indici pseudo-casuali, in modo da non fare sempre le stesse domande nello stesso ordine,
- *time* per permettere i countdown,
- *random* per verificare se la scelta tra i tre pulsanti A, B e C è una domanda o un trabocchetto.

## Link utili:

Si può copiare e incollare questo link per scaricare il progetto e questa relazione:

[https://github.com/Pedro-1309/Progetto\\_Programmazione\\_di\\_Reti](https://github.com/Pedro-1309/Progetto_Programmazione_di_Reti)