

Progetto di  
”Programmazione di Reti”  
Chat Game

Daniele Martignani

2020/2021

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Istruzioni di gioco . . . . .	2
1.2	Ruoli dei giocatori . . . . .	3
<b>2</b>	<b>Strutture dati</b>	<b>4</b>
2.1	Liste . . . . .	4
<b>3</b>	<b>Thread attivi</b>	<b>5</b>
3.1	Server . . . . .	5
3.2	Client . . . . .	5
<b>4</b>	<b>Istruzioni</b>	<b>6</b>
4.1	Server . . . . .	6
4.2	Client . . . . .	6

# Capitolo 1

## Introduzione

Il progetto consiste nel realizzare un programma che rispetti istruzioni della traccia n.3:

”Sfruttando il principio della CHAT vista a lezione implementate 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 si viene eliminati dal gioco e quindi si esce dalla chat. Se si seleziona invece una delle domande e si risponde correttamente al quesito si acquisisce un punto, in caso contrario si perde un punto. Il gioco ha una durata temporale finita; il giocatore che al termine del tempo ha acquisito più punti è il vincitore.”

### 1.1 Istruzioni di gioco

Per prima cosa, bisogna eseguire il server, in modo da permettere di connettersi. Nel momento in cui client viene eseguito esso si connette al server e quest’ultimo chiederà al client di autenticarsi inserendo il suo nome. Il server assegnerà un ruolo casuale al giocatore. Quando un giocatore è pronto, dovrà cliccare sul pulsante ”Ready”; quando tutti i giocatori presenti saranno pronti il gioco comincia, e non si potrà più entrare in partita.

Il giocatore per ricevere una domanda dovrà cliccare uno dei tre pulsanti, ma se seleziona quello trabocchetto perderà la partita. Per rispondere alla domanda bisogna scrivere la risposta in una apposita area e cliccare sul pulsante ”Answer”: se la risposta sarà giusta riceverà un punto altrimenti gli verrà sottratto. All’inizio del gioco, inoltre, viene avviato un counter di round e alla fine del counter il giocatore con più punti vince.

## 1.2 Ruoli dei giocatori

Quando un giocatore risponde ad una domanda, gli viene aggiunto un punto se la risposta è corretta, o gliene viene sottratto uno se la risposta è sbagliata.

# Capitolo 2

## Strutture dati

### 2.1 Liste

Ho utilizzato le liste per memorizzare le domande, i giocatori, i ruoli. La lista delle domande viene riempita con le domande dal file "questions.json" con un metodo della libreria json, la lista dei giocatori serve per tenere traccia delle informazioni di ogni giocatore (nome, punteggio, ruolo), mentre la lista dei ruoli serve per assegnarne uno ad ogni giocatore in modo casuale.

# Capitolo 3

## Thread attivi

### 3.1 Server

Nel server vengono utilizzati due thread: un thread che si occupa di accettare le connessioni in entrata, mettendo in attesa tutti i client; ed uno che invece ha il compito di gestire tutta la logica di gioco. Come argomento ha il client e ha il compito di gestire l'invio dei messaggi. A seconda del messaggio che l'utente invierà verrà eseguita una diversa azione.

### 3.2 Client

Nel client vengono utilizzati due thread: uno che gestisce la ricezione dei messaggi e li mostra tramite la libreria tkinter; ed uno che inizializza un counter, che ad ogni domanda aumenta e termina quando raggiunge il numero massimo di round.

# Capitolo 4

## Istruzioni

### 4.1 Server

Eseguire `src/game_server.py`. il server rimarrà in attesa di eventuali connessioni sull'indirizzo locale `"127.0.0.1"` e sulla porta `"8080"`.

### 4.2 Client

Eseguire `src/game_client.py`. Il client si conatterà all'indirizzo IP `"127.0.0.1"` e alla porta `"8080"`.