Relazione elaborato Programmazione di Reti

Frattarola Marco, Siroli Alex

4 giugno 2021

Indice

Capitolo 1

Introduzione

La traccia scelta è la numero tre, Chat Game, in cui bisogna realizzare un multiplayer playing game testuale. Il gioco offre la possibilità di giocare ad un numero variabile di giocatori che se rimangono connessi al termine del turno potranno partecipare ad una nuova partita.

Capitolo 2

Dinamiche di gioco

Il gioco permette l'accesso ad un numero variabile di giocatori che si sfidano accumulando punti; chi raggiunge per primo un determinato punteggio vince. Il numero di giocatori e il punteggio da raggiungere per la vittoria sono a discrezione del server. Un giocatore, per partecipare, deve inserire il suo nome e connettersi al server. Una volta connesso saranno visibili nella classifica i nomi di tutti i giocatori connessi alla partita. Una volta raggiunto il numero di partecipanti prescelto, la partita comincia:

1. Il giocatore deve scegliere un emoji tra le tre visualizzate. Una di queste tre porta il giocatore ad essere eliminato e disconnesso dalla partita mentre le altre due emoji permettono al giocatore di continuare la partita.

2. Se il partecipante è ancora in gioco, vede visualizzata una moltiplicazione da risolvere e il giocatore giocatore deve inviare nel più breve tempo possibile la risposta corretta.

3. Se la risposta inviata é giusta il punteggio del giocatore aumenta di un punto, viceversa, se é sbagliata, il punteggio diminuisce di un punto. Tutti i giocatori connessi vedranno a schermo il suo punteggio aggiornato.

4. Il gioco continua in questa maniera finché un giocatore non raggiunge il punteggio di vittoria prescelto.

Conclusa una partita, automaticamente i giocatori rimasti vengono connessi ad una nuova, con lo stesso nome inserito in quella precedente. Si aspetta che si raggiunga il numero di giocatori prescelto e una nuova partita comincia, con tutti i punteggi resettati, nella stessa modalità descritta sopra.

Capitolo 3

Implementazione

Per la realizzazione del gioco sono stati utilizzati un client ed un server. Per entrambi sono state utilizzati degli array di dizionari in cui vengono salvate le informazioni sui partecipanti alla partita. (nome, punteggio, id, ecc...)

3.1 Server

Il server si é occupato di gestire una partita con piú partecipanti, in particolare:

- Permette all'utente di scegliere il numero di giocatori e il numero di punti necessari per vincere.
- Accettazione di un nuovo client e inserimento nella partita.
- Avvio della partita
- Gestione dell'invio delle domande e della ricezione delle risposte.
- Comunicazione ai partecipanti della classifica in tempo reale.
- Fine della partita.
- Gestione di eventuali giocatori che si disconnettono.

3.2 Client

Il client si é occupato di gestire la partita per un solo giocatore, in particolare:

- Connessione al server
- Permette all'utente di inserire un nome
- Permette all'utente di visualizzare la classifica in tempo reale
- Scelta dell'emoji e disconnessione in caso di scelta sbagliata
- Permette all'utente di visualizzare la domanda e di inserire una risposta
- Gestione dell'invio delle risposte
- Disconnessione al server

3.3 Gestione dei thread

Lato server sono stati utilizzati i seguenti thread:

- thread principale per la gestione dell'interfaccia utente
- thread che si occupa delle connessioni in entrata
- più thread, ognuno dei quali si occupa di gestire la comunicazione con un singolo client

Lato client, invece, sono stati utilizzati due thread:

- thread principale per la gestione dell'interfaccia utente
- thread che si occupa della comunicazione con il server

3.4 Comunicazione client - server

Per la comunicazione tra client e server viene utilizzata una connessione TCP. Per facilitare la scrittura del codice sono stati identificati i vari tipi di pacchetti attraverso una enum, assegnando ad ognugno di essi un id numerico. In questo modo sia il client che il server sono in grado di riconoscere il tipo di pacchetto ricevuto e di assumere diversi comportamenti in relazione ad esso. Per semplificare la costruzione di pacchetti con diverse informazioni al suo interno (ad esempio id e nome di un giocatore), sono state create due funzioni encode() e decode() che permettono di stabilire delle regole di comunicazione, cioè in che modo estrarre o inserire le diverse informazioni in un singolo pacchetto.

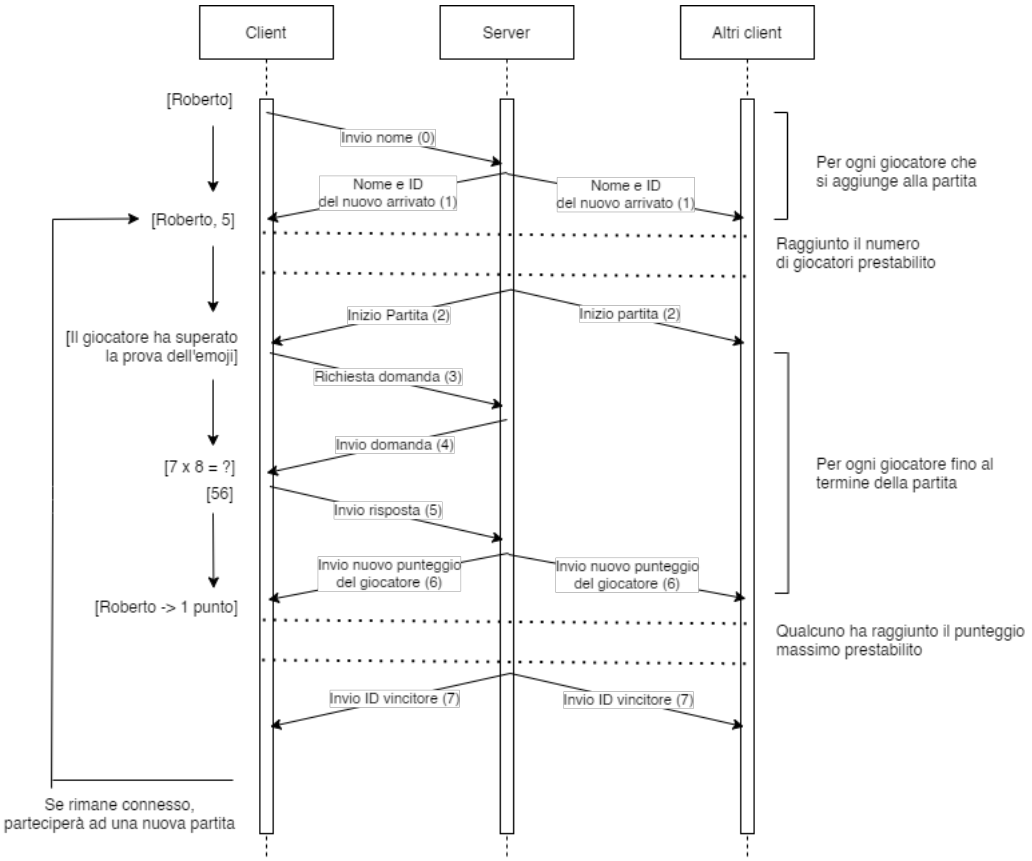

Figura 3.1: Diagramma di sequenza delle comunicazioni tra client e server nel caso in cui il client riesce a superare la prova delle emoji

Nel caso in cui un giocatore si disconnette, il server provvede a notificare agli altri giocatori la sua disconnessione, dando loro modo di aggiornare la classifica.

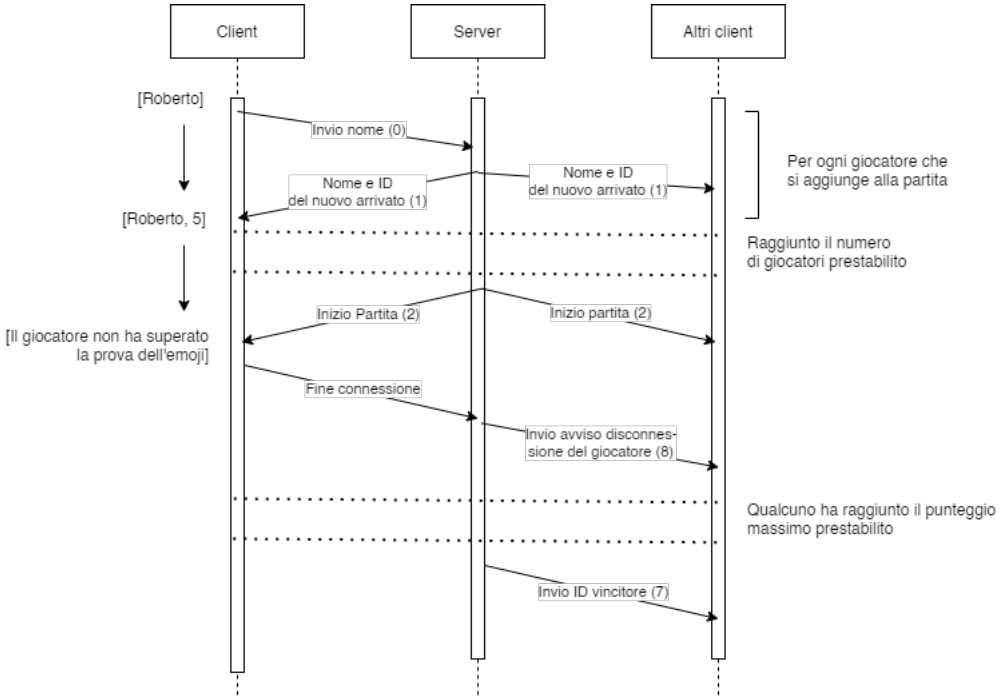

Figura 3.2: Diagramma di sequenza delle comunicazioni tra client e server nel caso in cui il client non riesce a superare la prova delle emoji