

Text-Twist

Progetto di Fine Corso – A.A. 2016/17

1. Descrizione del problema

Il progetto consiste nello sviluppo di un gioco multiplayer online. All'inizio di una partita il server fornisce ai giocatori una lista di lettere, quindi i giocatori devono formare delle parole con quelle lettere. Alla fine il server fornisce una classifica dei giocatori in base al numero e alla lunghezza delle parole trovate. Il server mantiene anche una classifica universale degli utenti.

Per giocare gli utenti si devono registrare, quindi possono:

- effettuare il login
- iniziare una nuova partita
- partecipare alla partita (su invito)
- consultare la classifica generale
- effettuare logout

Componenti

Text-Twist deve essere implementato mediante 2 componenti principali, che interagiscono usando diversi protocolli e paradigmi di comunicazione di rete (TCP, UDP e RMI).

Le componenti sono le seguenti:

TwistClient

Gestisce l'interazione con l'utente, tramite una user interface (testuale o grafica) e comunica con il TwistServer per eseguire le azioni richieste dall'utente. La sua struttura è quella di un "thin-client", in quanto la realizzazione delle principali funzionalità è demandata al server.

In seguito indicheremo con $\text{TwistClient}(u)$ la componente TwistClient associata all'utente u .

TwistServer

Gestisce la fase di registrazione, memorizza tutti gli utenti registrati e i loro punteggi totali, riceve richieste di creazione di partite e gestisce le partite: invia le lettere agli

utenti, riceve le parole, calcola i punteggi e invia la classifica ai giocatori. Poiché il client ha una struttura “thin”, tutte le strutture dati utilizzate per queste funzionalità sono gestite dal server stesso.

2. Funzionalità principali

a. Registrazione utenti

Quando un utente u apre l'applicazione, $\text{TwistClient}(u)$ offre la possibilità di effettuare la registrazione o il login. Per la registrazione, $\text{TwistClient}(u)$ richiede il nome utente e la password, che vengono quindi spediti, tramite RMI, al TwistServer che verifica se il nome dell'utente esiste già nel sistema.

- se non esiste, lo aggiunge alla lista di utenti e restituisce un messaggio di conferma.
- se esiste, restituisce un messaggio di errore

b. Login

TwistClient chiede all'utente di inserire la password, e invia *username* e *password* al TwistServer , usando RMI. In questo passo vengono inviati tutti i dati del cliente che servono per le altre funzionalità (in base al design specifico dell'applicazione). Il TwistServer verifica la correttezza dei dati e risponde con un messaggio di conferma.

c. Richiesta nuova partita

Un utente A inserisce i nomi degli N utenti con cui intende giocare, $U_1, U_2, \dots, U_N, N \geq 1$ (almeno un opponente) e $\text{TwistClient}(A)$ invia tali nomi al TwistServer , usando una connessione TCP. TwistServer verifica che ogni U_i sia online e avvisa ogni U_i dell'invito, usando una callback RMI messa a disposizione da ogni $\text{TwistClient}(U_i)$. Per ogni utente U_i , $\text{TwistClient}(U_i)$ notifica l'invito al giocatore e lo memorizza in una lista locale di inviti. Dopo aver richiamato tutte le callback, TwistServer conferma l'inoltro dell'invito all'utente A , chiude la connessione TCP e richiama anche la callback di A . Se uno degli utenti non risulta online l'utente A riceve un messaggio di errore.

d. Setup nuova partita

Quando l'invito viene notificato al $\text{TwistClient}(u)$, l'utente u può scegliere (in autonomia) di visualizzare gli inviti, usando il menu offerto da $\text{TwistClient}(u)$. Ci possono essere vari inviti disponibili (arrivati in modo asincrono), però l'utente deve scegliere di rispondere a uno di loro. L'invito può essere accettato o rifiutato. La risposta viene inviata al server su una connessione TCP. Quando un invito viene accettato, gli altri vengono rimossi dalla lista locale del cliente.

Se tutti gli utenti accettano l'invito, la partita può essere iniziata. Altrimenti, dopo 7 minuti, la partita viene annullata dal server (gli utenti che hanno accettato ricevono un messaggio di errore).

e. Avvio nuova partita

Una volta ricevuta la risposta da tutti i giocatori, il server invia, usando le stesse connessioni TCP su cui ha ricevuto la risposta, le lettere per la partita. I TwistClient ricevono le lettere, chiudono la connessione TCP e comunicano agli utenti che il gioco è pronto. Quando gli utenti confermano, i TwistClient mostrano le lettere e accettano le parole dal giocatore per un periodo di 2 minuti. Quando i 2 minuti finiscono, TwistClient ferma l'input dall'utente e invia le parole al server, usando il protocollo UDP. Il server aspetta un massimo di 5 minuti per ricevere le parole. Se in questo tempo un utente non invia nessun messaggio, riceverà 0 punti per la partita.

Ogni TwistClient può avere solo una partita attiva ad un certo momento, mentre il server deve essere in grado di gestire più partite in contemporanea.

Tip: è messo a disposizione un file contenente un dizionario di parole in inglese. Per generare le lettere iniziali, è sufficiente scegliere una parola di 6-7 lettere e permutare le lettere prima di inviarle ai giocatori.

f. Invio-risultati partita

Il punteggio di un giocatore viene calcolato come la somma dei punteggi per ogni parola trovata. Ogni parola riceve un punteggio uguale al numero di lettere utilizzate. Solo le parole valide ricevono un punteggio ed una parola risulta valida se usa solo le lettere a disposizione ed è inclusa nel dizionario.

Tutti i giocatori di una partita sono registrati ad un canale multicast. I risultati della partita (il punteggio e la classifica di questa partita) vengono inviati in multicast a tutti i giocatori della partita.

I punteggi vengono usati per la classifica della partita in corso, ma vengono anche sommati al punteggio generale dei giocatori. In questo modo, partita dopo partita, gli utenti accumulano dei punti, che contribuiscono alla loro classifica generale.

g. Visualizzazione classifica generale

Il sistema offre la possibilità di visualizzare la classifica generale degli utenti. La richiesta viene inviata al TwistServer che risponde con la lista ordinata di utenti e il loro punteggio totale. La comunicazione avviene mediante TCP, e la connessione viene chiusa dopo che il server invia la classifica.

Il server deve avere un meccanismo che gli consenta di memorizzare in modo persistente il proprio stato, che comprende gli utenti registrati e il loro punteggio. In questo modo lo stato può essere ripristinato in caso di crash inaspettato del server.

h. Logout

TwistClient(u) invia un messaggio RMI al server che mette l'utente offline (non può ricevere inviti). L'utente dovrà effettuare il login, nel caso in cui usi successivamente l'applicazione.

3. Interfaccia utente

Offre opzioni all'utente per eseguire le varie funzionalità:

- Fornire il nome utente e la password per la registrazione.
- Fornire il nome utente e la password per il login.
- inserire le parole composte con le lettere a disposizione
- Visualizzare la classifica generale.
- Logout

La realizzazione dell'interfaccia può essere:

- testuale mediante riga di comando, nella soluzione base,
- (opzionale) grafica: quest'ultima soluzione risulta più avanzata e consente di gestire meglio le operazioni asincrone.

4. Modalità di svolgimento del Progetto

Il progetto deve essere svolto singolarmente. Il materiale consegnato deve comprendere:

- il codice dell'applicazione e di eventuali programmi utilizzati per il test delle sue funzionalità
- Le istruzioni necessarie per compilare il codice sorgente ed eseguirlo
- la relazione in formato pdf che deve contenere:
 - una descrizione generale dell'architettura del sistema, motivando le scelte che non siano già indicate nel testo del progetto.
 - uno schema generale dei thread attivati, con particolare riferimento al controllo della concorrenza, e delle strutture dati utilizzate.
 - per ogni componente e struttura dati utilizzate, una descrizione delle classi definite (preferibilmente usando anche UML).

- **Solo per il corso B:** il codice eseguibile (**2 archivi eseguibili .jar**, uno per il client, l'altro per il server) con indicazioni precise sulle modalità di esecuzione.

L'organizzazione e la chiarezza della relazione, nonché l'organizzazione del codice e i commenti, influiranno sul voto finale.

5. Modalità di sottomissione del progetto

- **Corso A:** relazione e codice sorgente devono essere consegnati *sia in formato elettronico*, sul sito del corso, (<https://elearning.di.unipi.it/course/view.php?id=79>), *sia in formato cartaceo*, presso la portineria del Dipartimento,
- **Corso B:** relazione, codice e eseguibili devono essere consegnati *solo in formato elettronico* sul sito del corso (<https://elearning.di.unipi.it/course/view.php?id=78>), seguendo il link all'assignment che sarà disponibile per ogni appello.
- **Non si accettano sottomissioni in ritardo.**