

Progetto Programmazione di reti: TriviaTavern

Alessandro Palladino

A.A 2020-2021

Indice

1	Introduzione	2
2	Specifiche progetto	3
2.1	TriviaTavernServer	3
2.2	TriviaTavernClient	5
3	Note di progetto	6
4	Guida all'utente	7

Capitolo 1

Introduzione

Traccia 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 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.

Capitolo 2

Specifiche progetto

Il progetto ha come scopo quello di sviluppare un gioco stile chatgame dove una o più persone si sfidano in real time sfruttando una connessione in lan. Esso si basa su due file: TriviaTavernServer e TriviaTavernClient. TriviaTavernWithoutNetwork invece costituisce una prima versione semplificata senza l'uso di connessioni internet.

2.1 TriviaTavernServer

Rappresenta il lato server e costituisce il fulcro del gioco. Esso può essere diviso in 3 blocchi:

- Il primo rappresenta le principali funzioni invocate dal main:
 - `accept_connections()` Accetta le connessioni in arrivo
 - `client_manager()` Gestisce la connessione con il singolo client
 - `broadcast()` Invia messaggi a tutti gli altri client connessi
 - `send_msg()` Invia un messaggio al singolo client, semplifica il codice
- Il secondo rappresenta i metodi utilizzati dal blocco `client_manager()`:
 - `add_player()` Inizializza informazioni aggiuntive: ruolo e score.
 - `user_login()` Avvisa agli altri utenti che una persona è entrata.
 - `insert_num_player()` Chiede quanti giocatori parteciperanno al gioco, viene richiesto solo una volta.

- `preliminary_question()` Assegna ad ogni giocatore una scelta da effettuare, l'utente attraverso l'inserimento di un numero selezionerà la risposta. Se dietro la propria scelta si cela un tranello la persona ha perso il gioco e verrà disconnessa dal client. Viceversa il giocatore può andare allo step successivo e quindi all'inizio del vero gioco.
 - `start_question()` Invia domande diverse ad ogni client fino a quando qualcuno non finisce, avvisa inoltre se le risposte sono giuste o errate e il punteggio del giocatore.
 - `choose_question()` Viene invocata per poter generare domande randomiche.
 - `find_winner()` Trova il punteggio più alto, se ne è presente più di uno avviserà che è presente un pareggio, altrimenti stampa il nome del vincitore con il relativo punteggio.
 - `client_close()` Chiude la connessione con il client e avvisa che l'utente ha abbandonato la chat.
- Infine il terzo rappresenta i metodi usati per effettuare controlli di vario genere:
 - `check_number()` Controlla che l'input sia un numero e che sia compreso fra determinati parametri a seconda della funzione chiamante.
 - `check_player_ready()` Controlla che il numero di giocatori sia quello inserito inizialmente dal primo utente, avvisa se mancano o servono altre persone per poter iniziare a giocare
 - `check_answer()` Controlla che la risposta sia corretta
 - `check_score()` Aggiunge / sottrae punteggio a seconda che la risposta sia giusta / sbagliata
 - `check_quit()` Controlla che non sia stata inserita la stringa {quit}. Se è stata inserita allora avvierà la chiusura della connessione.

2.2 TriviaTavernClient

Riguarda il lato client e viene utilizzato da uno o più utenti per poter interagire con il server.

Possiede i seguenti metodi:

- `receive()` Gestisce l'arrivo dei messaggi
- `send()` Gestisce l'invio dei messaggi
- `on_closing()` Gestisce la chiusura della finestra

Capitolo 3

Note di progetto

In fase di progettazione si è pensato di rendere il gioco il più estensibile possibile di conseguenza basterà modificare le principali variabili per poter cambiare alcune parti del gioco. Le variabili modificabili sono:

- `role` Basterà aggiungere altri valori al dizionario per poter aggiungere ulteriori ruoli.
- `max_player` Per poter aggiungere o ridurre il numero massimo di giocatori.
- `min_player` Per poter impostare un minimo di giocatori.
- `max_option` Per poter selezionare il numero massimo di scelte disponibili per la domanda preliminare a tranello.
- `question_max` Per modificare il numero di domande che verranno eseguite, di conseguenza la durata del gioco verrà modificata.
- `question` ed `answer` Come per `role` basterà aggiungere valori per poter avere altre domande a disposizione.

Capitolo 4

Guida all'utente

Per poter eseguire il gioco basterà avviare il file `TriviaTavernServer` e successivamente in un altro terminale `TriviaTavernClient`.

All'avvio del server esso si metterà in attesa di connessioni mentre all'avvio del client verrà chiesto di inserire il server host, se non si conosce può essere lasciato vuoto, e successivamente la porta del server host, qui digiteremo 53000.

Dopodiché si aprirà una finestra che ci spiega il funzionamento del gioco e ciò che dovremo digitare. Per poter giocare in più giocatori sarà sufficiente avviare più client che si connetteranno tutti allo stesso server seguendo la stessa procedura.

Una volta che il gioco verrà avviato ad ogni giocatore viene chiesto di digitare un nome o un suo nickname e gli sarà attribuito un ruolo casuale che imposta un diverso valore di partenza in termini di punteggio. Verrà chiesto inoltre solo al primo utente di inserire il numero dei giocatori, dopodiché a tutti verrà chiesto di inserire un numero a scelta fra determinati valori.

Se il giocatore casca nel tranello ha perso e verrà disconnesso dal client.

Al termine di tutto ciò i superstiti aspetteranno che tutti i giocatori siano connessi, successivamente inizieranno ad apparire domande diverse a ciascun giocatore.

Il primo che risponde a più domande possibile fa terminare il gioco, vince però chi ha il punteggio più alto. In caso di parità essa verrà notificata ma non si sapranno i nomi dei giocatori. Poiché il gioco finisce i client vengono disconnessi automaticamente.