

PROGETTO RETI INFORMATICHE 2023/2024 - Francesco Panattoni 604230

Il progetto consiste in un'applicazione distribuita che implementa il gioco dell'Escape Room. L'applicazione è composta da un file `server.c` e un file `client.c`, che implementano per l'appunto il processo Server e il processo Client dell'applicazione. Server e Client comunicano attraverso il protocollo TCP, poiché per l'applicazione è necessaria una comunicazione affidabile. Il Server può essere di due tipi: Main Server o Game Server.

Client

Il Client, in pratica, implementa un giocatore. All'inizio si collega direttamente al Main Server e decide se fare la registrazione, il login, o uscire. Se decide di fare la registrazione o il login, gli arriverà il numero di porte dove ascoltano i vari Game Server. Il Client sceglie un Game Server che implementa l'Escape Room a cui vuole giocare. Se il Game Server non ha ancora fatto partire una partita verrà collegato. Qua potrà scegliere nuovamente se giocare ("*begin*") o uscire ("*quit*"). Se sceglierà di giocare potrà decidere quali comandi usare per muoversi all'interno dell'Escape Room e vincere il gioco. Il Client ogni tot. secondi manda un "*timeout*" che gli permette di rimanere aggiornato sullo stato della partita e sui messaggi pendenti. In questo modo l'interazione dei Client e i messaggi risultano praticamente real-time. Ogni giocatore può interagire quando vuole e il suo non interagire non blocca gli altri giocatori. Al termine della partita può scegliere se giocare nuovamente, oppure se uscire e provare un'altra Escape Room su un altro Game Server, o se uscire anche dal Main Server.

Main Server

Il Main Server è unico e si occupa delle operazioni di registrazione, login, logout e uscita. Inoltre si occupa anche della creazione dei Game Server e del loro collegamento con un Client. Il Main Server implementa un meccanismo di I/O Multiplexing per la gestione contemporanea di più socket. Si utilizza un file `.txt` come database (chiamato per l'appunto "`database.txt`"), contenente username e password attraverso il seguente formato: "`username password \n`". La password viene inviata già criptata da Client. Durante la registrazione, si controlla che lo username inviato non sia già presente all'interno di `database.txt`. Una volta verificato ciò vengono inseriti username e password. Il login verifica che lo username e la password inviate coincidano con quelle contenute in `database.txt`. Si controlla anche che l'utente non abbia già fatto il login utilizzando una lista che memorizza gli username di tutti gli utenti che hanno fatto il login (o la registrazione). Il logout serve proprio a togliere l'utente da questa lista. Dopo la registrazione o il login, ci si può collegare ad un Game Server. Verranno inviate le porte sulle quali i Game Server sono in ascolto. Se non vi sono Game Server in ascolto il Client dovrà aspettare che vengano creati attraverso il comando: "`start <port>`". Una volta digitato questo comando saremo chiamati a decidere quale Escape Room implementare. Una volta digitato il codice dell'Escape Room che vorremo implementare, verrà chiamata una `fork()` e così sarà il processo figlio a gestire il Game Server. Il processo padre continuerà a gestire il Main Server e comunicherà con i figli attraverso i segnali. Il comando "`stop`", manda un segnale a tutti i Game Server attivi e li chiude nel caso non ci sia una partita in corso, grazie ad una struttura fondamentale (`server_descriptor`) che si preoccupa di memorizzare tutti i dati necessari per questo compito. Come richiesto dalle specifiche di progetto se un Game Server ha fatto partire il gioco, si dovrà aspettare fino al termine della partita. Il Client sceglierà la porta del Game Server che implementerà l'Escape Room che vorrà giocare. Così potrà collegarsi in autonomia al Game Server. Il Main Server infine ha una porta che viene scelta nel momento in cui è eseguita l'applicazione. Il Main Server usa protocolli "`text`" per inviare le stringhe e protocolli "`binary`" per mandare codici, status, lunghezze delle stringhe da inviare prima di inviarle, eccetera ... tramite protocolli "`binary`".

Game Server

I Game Server usano un meccanismo di I/O Multiplexing e si occupano di implementare un Escape Room. Ci sono due Escape Room implementabili al momento, ma ciò è estendibile. Le Escape Room sono quella Medievale (con codice 0), ambientata in un castello e quella dell'Antico Egitto (con codice 1), ambientata in una piramide. Il Game Server si occupa dell'implementazione del gioco in sé e per sé. All'inizio il Client ha solo due comandi: *begin* e *quit*. Il comando "*begin*" indicherà che il Client è pronto. Tuttavia se il Client è da solo, può decidere di giocare in single-player, altrimenti dovrà aspettare che gli altri Client siano tutti pronti. Una volta che almeno due Client hanno indicato di essere pronti, partirà un timer di pochi secondi e una volta scaduto si potrà finalmente giocare. Il comando "*quit*" invece fa uscire dal Game Server, tuttavia chiede al Client se vuole uscire pure dal Main Server e in caso affermativo manda una richiesta di logout al Main Server. Una volta iniziata la partita verranno stampati i comandi possibili da utilizzare per interagire con il gioco, con una breve descrizione di quello che fanno. Dopo di che verrà inviato il nome e la descrizione della prima location. Da qui il gioco sarà partito e ci si potrà muovere all'interno delle location dell'Escape Room scelta, cercando di prendere oggetti, risolvendo indovinelli e accumulando punti. L'Escape Room, le location, gli oggetti e gli indovinelli sono tutti implementati tramite apposite strutture. Come richiesto dalle specifiche all'inizio della partita viene attivato un timer. Se il timer scade prima che i giocatori abbiano raggiunto l'uscita, i giocatori hanno perso. Nell'Escape Room Medievale per vincere bisogna giungere alla fine del percorso e risolvere l'indovinello finale per prendere l'uscita dal castello. Nell'Escape Room Egizia bisogna risolvere i vari indovinelli, stare attenti alle trappole e arrivare ad un punteggio tale che permetta di interagire con la tomba del faraone per poter risolvere l'ultimo enigma e scappare dalla piramide. Al termine di ogni comando valido, sarà inviato il tempo rimanente, i giocatori presenti e i rispettivi punteggi. Il Game Server inoltre permette ai giocatori di scambiarsi dei messaggi per comunicare tra loro. Se il gioco è già partito, il Game Server rifiuterà tutte le connessioni da parte di nuovi Client. I Game Server usano protocolli "text" per inviare le stringhe e protocolli "binary" per mandare codici e lunghezze delle stringhe inviate tramite i protocolli "text", in particolare è prassi inviare la lunghezza della stringa prima, poi la stringa e infine riceve un codice che indica il risultato dell'allocazione dello spazio per la stringa da parte del Client.

Gestione delle strutture di gioco

Le strutture del gioco sono:

- **server_descriptor**: Si occupa di tenere le informazioni rilevanti al Server implementato (la porta su cui ascolta, l'ID del processo su cui gira se è un Game Server e così via);
- **player_descriptor**: Si occupa di tenere le informazioni rilevanti del giocatore e quindi del Client;
- **escape_room_descriptor**: Contiene il nome, la descrizione, l'ID e un vettore di location che compongono l'Escape Room;
- **location**: Contiene nome, descrizione, oggetti della location e delle door;
- **door**: Contiene la location successiva e se è bloccata;
- **object**: Contiene nome, descrizione, enigma e varie indicazioni di utilizzo dell'oggetto;
- **riddle**: Contiene l'enigma dell'oggetto, con tipo, domanda e risposta;
- **message**: È una lista dei messaggi pendenti che possono essere di sistema o utente;
- **users**: È una lista di utenti che hanno effettuato il login.