

# ESCAPE ROOM - DOCUMENTAZIONE PROGETTO RETI INFORMATICHE

Andrea Bochicchio – 642196

## - Informazioni implementazione

Il progetto è un'applicazione distribuita che realizza il gioco dell'Escape Room Musicale. I dati tra client e server sono trasferiti con il **protocollo TCP** perché è necessario che lo scambio dei messaggi sia affidabile, anche se ciò va a discapito della velocità.

Il server gestisce più client che giocano contemporaneamente. Tuttavia, poiché il giocatore deve esplorare la stanza e risolvere enigmi, il server trascorre la maggior parte del tempo in attesa di richieste. Questa informazione chiave ha portato alla decisione di utilizzare un server basato su **I/O multiplexing**. In questo modo è possibile gestire più connessioni contemporaneamente (a differenza del server iterativo) e rispondere ai client che inviano richieste, senza dover usare processi aggiuntivi che rimarrebbero in attesa per la maggior parte del tempo (come nei server concorrenti).

La dimensione dei dati inviati non è nota a priori, pertanto durante la comunicazione il destinatario deve ricevere il numero di byte da leggere dal socket e solo successivamente riceve il messaggio effettivo. In realtà, la comunicazione tra client e server è abbastanza standard poiché il messaggio è composto da un comando e uno o due parametri, quindi sarebbe stato possibile definire una dimensione massima. Tuttavia, ciò avrebbe limitato aggiornamenti futuri. Il vantaggio è che non sarebbe stato necessario effettuare due *send* e due *recv* (una per la dimensione e una per il messaggio effettivo), ma questo non è un problema in questa applicazione perché non è generato un traffico eccessivo.

Sono state implementate due strutture dati: *Formato\_cmd* per descrivere i dati da client a server e *Formato\_resp* per descrivere i dati da server a client. Queste sono molto utili per garantire una chiarezza durante la gestione dei comandi. Per lo scambio della dimensione del messaggio si utilizza il **Binary Protocol**; invece, per il messaggio effettivo, si converte la struttura in stringa, quindi si utilizza **Text Protocol**. Poiché la maggior parte dei dati inviati è di tipo testuale, questa risulta essere la scelta migliore, dato che semplifica anche il controllo della comunicazione.

Quando il giocatore riceve una risposta dal server, sono inviate anche informazioni aggiuntive, quali i token raccolti e mancanti, notifiche ecc... L'invio di queste informazioni potrebbe essere ridondante, dato che sarebbe stato possibile inviare questi dati solo in caso di variazione. Un caso particolare riguarda il tempo rimanente: il server avrebbe potuto inviare il tempo di inizio e poi il client avrebbe potuto calcolare il tempo rimanente ad ogni comando. In questo modo, se il tempo fosse scaduto, il client avrebbe impedito l'invio di ulteriori comandi al server (che avrebbe comunque effettuato un controllo aggiuntivo). Detto ciò, ho deciso di inviare comunque queste informazioni perché, solitamente, occupano poco spazio in confronto al messaggio effettivo, e quando si verificano variazioni, gestirle a livello client potrebbe causare una complessità inutile, mentre è più agevole per il server. Un esempio di variazione del tempo sono i minuti bonus aggiunti se un utente aiuta un altro in un indovinello (vedi *funzionalità a scelta*).

## - Informazioni sul gioco

L'utente può scegliere tra due scenari: "Stanza Rock", "Stanza Pop". In pratica, c'è un unico scenario con caratteristiche diverse a seconda della scelta effettuata.

Gli scenari sono istanziati durante l'esecuzione del codice. Le informazioni sono organizzate in strutture dati per facilitarne la gestione, garantire la scalabilità e rendere il sistema adatto al caricamento delle informazioni degli scenari da file, se richiesto in futuro.

Le credenziali dei giocatori registrati sono nel file *db\_utenti.txt*.

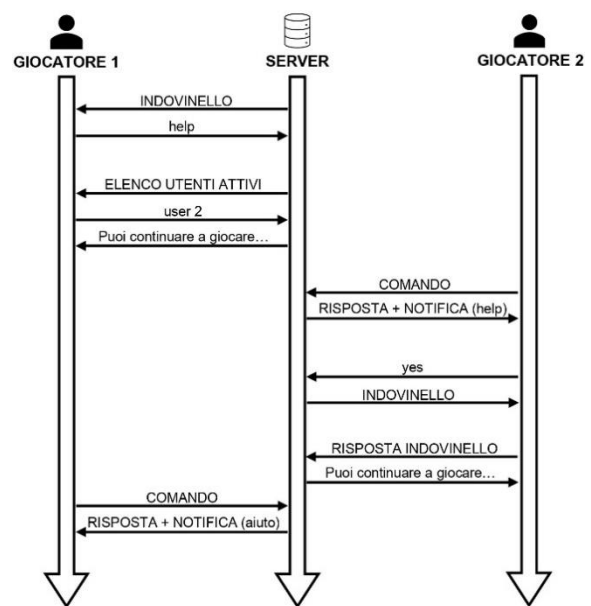
Nel gioco sono presenti degli oggetti bloccati da enigmi. Sono state sviluppate tre tipologie di enigmi: domanda a risposta multipla, indovinello con un numero limitato di risposte (il giocatore perde se esaurisce i tentativi), l'uso di oggetti in un ordine specifico. Solo dopo aver risolto l'enigma è possibile prendere l'oggetto.

Il giocatore può raccogliere fino a tre oggetti. Quando raggiunge questo limite e cerca di prendere un nuovo oggetto, il server automaticamente invia l'elenco degli oggetti già presi tra cui l'utente deve scegliere quale lasciare. Dopo aver fatto la scelta, può prendere l'oggetto richiesto inizialmente o un altro.

### - Funzionalità *help*

Quando il giocatore cerca di prendere un oggetto bloccato da un indovinello con un numero di tentativi limitati per la risposta, ha la possibilità di chiedere aiuto ad un altro giocatore. Questa funzionalità è disponibile una sola volta durante il gioco.

Nello specifico, il procedimento è il seguente: una volta ricevuto l'indovinello, il giocatore può digitare il comando *help* e il server fornirà l'elenco degli id e degli username degli utenti attivi. A questo punto, il giocatore seleziona l'utente a cui chiedere aiuto utilizzando il comando *user* seguito dall'id dell'utente scelto. Il giocatore può poi continuare a giocare e riceverà una notifica quando il destinatario ha deciso se aiutarlo. Il destinatario, quando invia un messaggio al server, oltre alla risposta al suo comando, riceverà una notifica con la richiesta di aiuto a cui potrà rispondere con *yes* o *no*. Se risponde in maniera affermativa riceverà l'indovinello dal server a cui può rispondere. In caso di risposta corretta, riceverà un bonus di tre minuti. Il giocatore che ha chiesto aiuto riceverà quindi la risposta suggerita dall'altro utente nelle notifiche.



La funzionalità a scelta si sarebbe potuta implementare facendo comunicare direttamente i due giocatori, offrendo l'importante vantaggio di una comunicazione in tempo reale.

Tuttavia, è necessario considerare che il server ha un ruolo importante perché è suo il compito di gestire gli utenti attivi, il limite di richieste di aiuto, il controllo della correttezza della risposta e l'assegnazione del tempo bonus. Per quanto appena detto, si è deciso di far gestire al server la comunicazione tra i due giocatori.

Attualmente questa funzionalità è stata implementata solo per richiedere aiuto per risolvere l'indovinello ma può essere estesa in modo da far collaborare i due giocatori in tutta le fasi della partita.