

## Documentazione progetto – Martina Fabiani 615788

Il progetto viene mandato in esecuzione dal comando `./exec2024.sh` lanciato da terminale. Si aprono inizialmente 3 terminali corrispondenti a: server, client ed other. Per iniziare è necessario avviare il server digitando su stdin il comando `'start'`, dopo di che il server si pone in ascolto. A questo punto client e other si connettono con il server: su `'client'` è implementata la partita dell'utente mentre su `'other'` è implementata la funzionalità speciale, dunque esso si metterà in attesa di essere utilizzato dal giocatore attraverso l'apposito comando. La comunicazione tra i dispositivi e il server avviene tramite **socket TCP**, in quanto non è richiesta velocità da parte del server ma ci interessa che sia affidabile.

Lo scambio di messaggi è implementato attraverso due funzioni definite nel file `'utility.c'`. Queste funzioni si occupano prima dello scambio della lunghezza del messaggio e successivamente dello scambio del messaggio vero e proprio utilizzando il **protocollo text**. Questo protocollo è vantaggioso quando si lavora principalmente con stringhe, come nel nostro caso, poiché evita la necessità di conversioni e problemi di endianess. Questa scelta aumenta il traffico nella rete ma rende più agevole lo scambio.

Per gestire la partita sono state create 3 strutture dati definite nel file `'partite.h'`. Quando il giocatore, dopo aver effettuato l'accesso, inizia una nuova partita viene allocata memoria per una nuova struttura `'Partite'` nella quale viene salvato l'username del giocatore, lo scenario scelto, gli oggetti raccolti e i token conquistati.

Nella struttura `'Argomento'` invece sono presenti tutti gli oggetti e stanze di questa escape room con la relativa descrizione ed il loro status, ovvero se sono stati sbloccati o meno dall'utente. Infine ad ogni `'Argomento'` bloccato è associata una struttura `'Enigma'` contenente l'enigma corrispondente e la relativa risposta.

Lo scenario `'Museo Egizio'` è composto da 7 oggetti, due dei quali sono porte che danno accesso a 2 location diverse. Gli enigmi totali sono 4, tutti di tipologie diverse: completamento parole con lettere mancanti, domanda con risposta a scelta multipla, individuazione codice lucchetto e completamento di una sequenza numerica.

Infine per aggiungere oggetti nello zaino, se quest'ultimo risulta pieno, è stata adottata una politica FIFO.

### Server

Il server una volta avviato accetta da stdin solo il comando `'start'`, per porsi in ascolto di richieste di connessione e il comando `'stop'`, per essere spento. È implementato con **I/O Multiplexing** in quanto deve gestire più connessioni, come ad esempio socket di comunicazione, socket di connessione e stdin, che non vengono chiuse subito ma vengono mantenute aperte; dunque ho ritenuto questa implementazione più efficiente e pratica nel mio caso di utilizzo.

Una volta connesso con i dispositivi, il server si occupa di gestire i comandi inseriti dal giocatore principale nello stdin. Quando comunica con il client principale, il server riceve una stringa contenente il comando ed eventuali argomenti, che suddivide inserendo le varie parti nelle apposite variabili. In base al comando ricevuto viene mandata in esecuzione la relativa funzione implementata nel file `'funzioni_cmd.c'`. Alla funzione, nella maggior parte

dei casi, viene mandato come argomento il puntatore alla struttura della partita corrente, inizializzato al momento dell'avvio della partita, l'id del socket di comunicazione e, se presente, l'argomento del comando digitato dall'utente.

Per distinguere i due dispositivi connessi, il server, al momento del login da parte dell'utente, salva in una variabile l'id del socket di comunicazione con il client; così, quando viene richiesto l'aiuto dell'host speciale, è facilmente individuabile l'id del socket di comunicazione con l'host speciale, dato che in questo caso specifico le connessioni massime sono 2.

## Client

Il client, dopo essersi connesso al server, procede con la fase di login/registrazione (i dati del profilo vengono salvati su un file di testo *utenti.txt*), se l'username inserito è già registrato nel file o la password di login è errata, il giocatore non potrà accedere (l'username è una chiave primaria). L'utilizzo del file consente la persistenza dei dati e garantisce maggiore robustezza ad eventuali guasti, permettendone nel caso un ripristino efficiente.

Dopo aver fatto l'accesso l'utente potrà scegliere lo scenario in cui giocare e iniziare la partita vera e propria. Il client entra in un ciclo infinito dove l'utente dovrà inserire i comandi per poter giocare. All'inizio di questo ciclo viene avviato anche il timer della partita della durata di 10 minuti, gestito lato client.

Prima di inviare i comandi inseriti al server, viene controllata la corretta sintassi sul client per evitare traffico inutile e sovrabbondante di messaggi. Solo dopo si ha un effettivo invio del messaggio al server che si occuperà della gestione del comando.

Il client esce dal ciclo e chiude la connessione solo in tre casi: se l'utente digita il comando 'end', se scade il tempo e quindi perde oppure se l'utente completa l'escape room e vince.

## Other

Dopo essersi connesso al server, si blocca in attesa che il giocatore principale chieda del tempo bonus digitando il comando **'time'**.

Quando riceve il messaggio di richiesta d'aiuto dal server, il secondo utente potrà inserire 3 numeri a suo piacimento di qualsiasi grandezza, dunque non necessariamente potranno essere d'aiuto al giocatore (ad esempio può inserire 3 zeri).

Dopo aver inviato i numeri digitati al server, la connessione viene chiusa, dato che questa funzionalità può essere usata una sola volta durante la partita.

### *Funzionalità speciale: **time***

La funzione a piacere sviluppata consente al giocatore di ottenere del tempo bonus per poter completare la missione. Digitando il comando **'time'** il giocatore indirettamente chiede aiuto all'host speciale, il quale viene "risvegliato" dal server ed il secondo utente dovrà inserire 3 numeri, che saranno i minuti bonus che il giocatore principale potrà ricevere.

Solo dopo che l'host speciale ha inviato i 3 numeri al server, al giocatore principale appariranno i 3 possibili pacchi da scegliere, i quali conterranno i minuti bonus proposti dal secondo giocatore.

Una volta effettuata la scelta da parte dell'utente, vengono svelati i minuti contenuti nel pacco scelto e aggiunti al timer della partita. Il giocatore non potrà più usare questa funzionalità durante la partita in corso.