

Relazione Progetto

Traccia 3 – CHATGAME

Sofia Tosi 0000914693

Laura Leonardi 0000915873

Serafino Pandolfini 0000914941

INDICE

- 1 – Introduzione
- 2 – Descrizione
- 3 – Dettagli implementativi
- 4 – Librerie utilizzate

1 – INTRODUZIONE

Il progetto implementa un'architettura client-server per il supporto di un Multiplayer Playing game testuale in rete locale.

2 – DESCRIZIONE

Abbiamo sfruttato il principio della chat per realizzare un gioco fantasy. I giocatori sono accolti dal master (chatgame_server.py) che richiederà l'inserimento del nome e assegnerà un ruolo ad ogni giocatore (la lunghezza del ruolo+nome può essere al massimo di 25 caratteri).

I giocatori saranno al massimo 10 e al minimo 1 e possono comunicare tra loro grazie a una chat.

Quando viene premuto il bottone “START” nella GUI del server non sarà più possibile aggiungere altri giocatori(chatgame_client.py).

A questo punto il master propone una situazione a cui si può reagire secondo tre opzioni. Due di esse conducono a una domanda mentre una a un trabocchetto che fa uscire dal gioco. Se si risponde correttamente alla domanda verrà aggiunto un punto, in caso contrario verrà tolto. La prima situazione sarà uguale per tutti, successivamente le strade dei giocatori si divideranno. Durante il gioco la classifica è continuamente visualizzata in alto a destra, nella GUI di ogni giocatore, ordinata per punteggio decrescente.

Le domande e le situazioni presentano elementi surreali e fantastici tipici dei giochi di ruolo.

Il gioco può terminare o allo scadere del tempo, o quando tutti i giocatori sono caduti nel trabocchetto.

Se il gioco terminerà allo scadere del tempo, comparirà una finestra con il punteggio del giocatore e l’esito (“Vittoria” / “Sconfitta”).

Se un giocatore che è stato eliminato ha totalizzato un punteggio superiore a tutti gli altri giocatori, risulta comunque il vincitore del gioco.

Nello svolgimento del gioco ogni 20 secondi viene mandato un messaggio che indica il tempo restante.

Scelte effettuate:

- Utilizzo dei dizionari per controllare in maniera più efficiente i dati, essendo i dizionari pensati per gestire coppie di elementi chiave-valore.
- Utilizzo delle matrici per la gestione delle situazioni trattandosi di insiemi di dati complessi da gestire diversamente. Infatti, in alcune situazioni sono richiesti solamente specifici elementi della struttura, che risulterebbe molto complesso gestire in altro modo.
- Gestione dell’interazione delle domande tramite bottoni per semplificare l’inserimento dei dati da parte del giocatore.

- Gestione di alcuni caratteri speciali, quali ad esempio “~” e “,” che se inviati in chat potrebbero venire interpretati in modo errato dalle funzioni di gestione dei messaggi.
- Sono state implementate due code (a partire da liste) che impediscono alle situazioni e alle domande che sono state estratte recentemente di venire nuovamente selezionate.
- Abbiamo scelto una durata per il gioco relativamente breve (100 secondi) e di mandare messaggi ogni 20 secondi che avvisano il giocatore del tempo restante per valorizzare la capacità di rispondere alle domande sotto pressione.

Riportiamo di seguito una foto dell'interfaccia del server e del client:



Salve! Digita il tuo Nome seguito dal tasto Invio!
Benvenuto Barbaro Sofia!
Paladino Laura si è unito all chat!
Stregone Serafino si è unito all chat!
MASTER: Benvenuti nel Sottomondo avventurieri!
MASTER: Il vostro scopo è sopravvivere a una serie di situ
MASTER: Per ogni situazione superata vi verrà posta una
MASTER: Se risponderete correttamente guadagnerete u
MASTER: Se sbaglierete invece perderete un punto.
MASTER: Alla fine della partita chi ha più punti vince!
MASTER: Ogni partita dura 100 secondi.
MASTER: Buon Gioco!!
Ti trovi davanti a 3 porte magiche, quale scegli?
La porta ti conduce a un albero interrogativo
MASTER: Mancano ancora 80 secondi!
MASTER: Mancano ancora 60 secondi!

PUNTEGGI:

- Barbaro Sofia: 0
- Paladino Laura: -1
- Stregone Serafino: -1

Porta rossa

Porta blu

Porta gialla

Invio

Cosa usi per allontanare un vampiro?

Pomodoro

Aglio



3 – DETTAGLI IMPLEMENTATIVI

Abbiamo implementato una connessione Client-Server utilizzando come porta la 53000, mentre per la gestione dell'indirizzo IP viene richiesto un inserimento da parte dell'utente dell'IP desiderato, che verrà mostrato nella GUI del server.

Sono state utilizzate le funzioni `rcv()` e `send()` della libreria socket per gestire l'invio dei messaggi con un `buffer size` di lunghezza 1024 byte.

Abbiamo gestito tutte le eccezioni che si verificano frequentemente con il sistema di `try-except` quali, ad esempio, l'eccezione che viene generata se si tenta di eseguire il client prima del server o se si prova a far partire il gioco prima che tutti i client aperti si siano autenticati con il nome utente.

Strutture utilizzate:

- ampio utilizzo di dizionari, per esempio nella gestione dei client e dei relativi indirizzi dal lato del server e nella gestione del punteggio lato client.
- utilizzo di due file (“domande.txt”, “situazioni.txt”) per ottenere le domande e le situazioni da utilizzare all’interno del gioco, generati dal file “generatore_domande.py”
- ampio uso delle liste
- utilizzo di una matrice per la gestione delle situazioni
- utilizzo di tuple, ad esempio come valore di dizionari

Thread utilizzati:

- un thread per il server
- un thread per ogni client
- un thread per la gestione del tempo

4 – LIBRERIE UTILIZZATE

- messagebox, tkinter
- AF_INET, socket, SOCK_STREAM
- time, sleep
- threading, Thread
- random

