

Progettazione Yahtzee

La progettazione riguarda un applicativo di rete del gioco Yahtzee ovvero “un gioco di dadi basato su combinazioni e strategia, in cui i giocatori lanciano cinque dadi per ottenere il punteggio più alto in 13 turni”.

- 1) Come linguaggio di programmazione scelto, ho scelto Java.
- 2) Per gestire le comunicazioni tra i vari componenti del sistema userò la libreria “java.net” che include le classi “java.net.Socket” e “java.net.ServerSocket”. Queste classi sono necessarie per creare connessioni dirette tra client e server senza dipendenze esterne. Per l’invio e la ricezione dei dati, userò la libreria “java.io” con le classi “java.io.ObjectInputStream” e “java.io.ObjectOutputStream”. Queste classi sono necessarie per convertire gli oggetti in dati e viceversa, garantendo che tutti i messaggi scambiati abbiano una struttura chiara e gestibile.
- 3) Funzionalità essenziali:
 - sistema di autenticazione che permette agli utenti di registrarsi e di effettuare il login in modo sicuro;
 - gestione delle partite di Yahtzee, singleplayer o multiplayer. Il server coordina le connessioni dei client, i lanci dei dadi, il calcolo del punteggio e i turni;
 - contatore dei turni, Yahtzee dura massimo 13 turni, dopo il tredicesimo turno la partita finisce e viene deciso il vincitore.Funzionalità facoltative:
 - classifica dei giocatori in base al numero di vittorie ottenute;
 - stato della partita in corso (tiri rimasti, punteggi mancanti).
- 4) Per la comunicazione in rete utilizzerò i socket basati sul protocollo TCP. Il TCP mi garantisce la consegna affidabile e in ordine dei pacchetti, necessario per mantenere sotto controllo lo stato di ogni partita e sincronizzare correttamente i turni. Yahtzee è un gioco che richiede che ogni comando venga ricevuto e processato nell’ordine corretto. TCP garantendo l’integrità della trasmissione si adatta meglio di UDP per questo caso.
- 5) Per proteggere i dati sensibili, durante la fase di registrazione e di login utilizzerò una chiave asimmetrica, per garantire sicurezza nelle informazioni più sensibili come la password e per stabilire una chiave simmetrica da usare per il resto della comunicazione tra client e server. La chiave simmetrica verrà scelta con l’algoritmo AES a 256 bit. Questo mi permette di gestire anche una grande quantità di client senza dover far lavorare troppo il sistema data la pesantezza della chiave asimmetrica.

Inoltre mi piacerebbe implementare un’interfaccia grafica usando Swing e un tutorial per capire come si può giocare a Yahtzee, questa per l’UX di chi userà l’applicativo. Per la parte di

sicurezza volevo implementare una limitazione dei tentativi di accesso per ridurre il rischio di attacchi bruteforce e dei meccanismi per evitare attacchi di SQL injection e buffer overflow.

SCHEMA AD ALTO LIVELLO DELL'ARCHITETTURA DEL SISTEMA

Le due componenti principali sono il client e il server.

Il client consente agli utenti di giocare a Yahtzee. Possono registrarsi, fare il login, eseguire i comandi di gioco e visualizzare i punteggi e la classifica dei giocatori. Il client comunica con il server con il protocollo TCP.

Il server e' responsabile dell'autenticazione dei giocatori e la logica del gioco. Elabora le richieste dei client, coordina le partite multigiocatore e la logica randomica del lancio dei dadi e aggiorna in tempo reale la classifica dei giocatori. Inviando poi gli aggiornamenti ai client.

Il flusso dei dati e': client fa una richiesta al server, server riceve, elabora, aggiorna (tipo stato della partita o classifica) e invia la risposta ai client.