

Laboratorio III - A.A. 2022/2023

Relazione sul Progetto d'esame

Gianni Pan

Matricola: 615945

Obiettivo del progetto:

Implementazione di una variante del famoso gioco WORDLE, divenuto virale alla fine del 2021.

Il progetto contiene le seguenti classi:

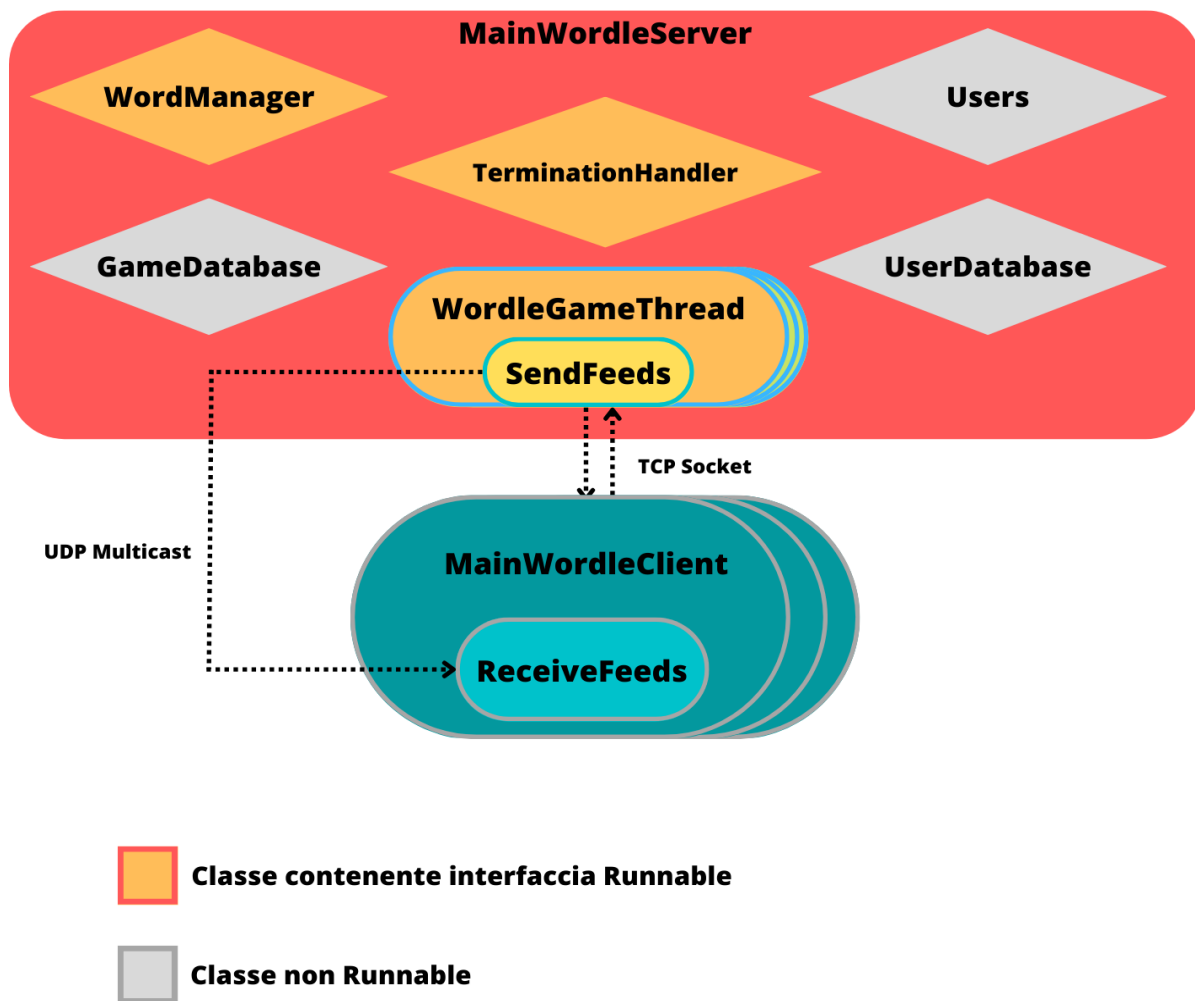
- MainWordleClient
- MainWordleServer
 - WordleGame
 - Utenti
 - WordManager
- TerminationHandler

E i seguenti file:

- database.json
- game.json
- words.txt
- client.properties
- server.properties

Parte I

Lo Schema del progetto.



Descrizione breve delle classi usate:

MainWordleServer:

- Legge e inizializza le variabili di configurazione dal file `server.properties`.
- Lancia un thread `scheduleWordSelection` che esegue `WordManager` ogni tot secondi. Il tempo che intercorre da un cambio di parola all'altro è definito nel file `server.properties`.
- Costruisce un `FixedThreadPool` di thread `WordleGame` per gestire le connessioni con i client. Entra in un ciclo in cui aspetta richieste di connessione da parte di client tramite metodo `accept()`.

MainWordleClient:

- Instaura una connessione TCP con il `MainWordleServer`. Di conseguenza il `MainWordleServer` assegna al client un thread `WordleGame`.
- Si unisce al gruppo Multicast nel quale riceverà le notifiche condivise dagli altri utenti su richiesta dell'utente stesso.
- Una volta connesso entra nel ciclo principale nel quale manda i comandi dell'utente per:
 - Registrazione e autenticazione utente.
 - Funzionalità sbloccate dopo autenticazione effettuata.
 - Tentativi durante una partita di WORDLE.

Status:

- È un tipo enumerativo che contiene gli stati in cui il client e `WordleGame` transitano durante la connessione.

WordleGame:

- Viene assegnato uno per connessione di client attraverso MainWordleServer.
- Gestisce le richieste del client inviando risposte elaborate sul tipo di richiesta.
- All'interno del ciclo principale
 - Riceve le richieste del client,
 - Elabora la richiesta, esegue le azioni richieste,
 - Manda messaggi o avvisi che il client riconosce per avvertire dell'esito e per orientare il client in caso di cambio status.
- È responsabile anche dell'immagazzinamento e dell'invio delle notifiche condivise dagli altri utenti.

Utenti:

- Contiene vari metodi che operano sul file database.json attraverso controllo e aggiornamento sui dati.

WordManager:

- Contiene il metodo run() nel quale, attraverso lo scheduledWordSelection avviato dal MainWordleServer, viene scelta e aggiornata la parola da indovinare.
- Contiene vari metodi che operano sul file game.json, il quale contiene la parola da indovinare e un array di giocatori che hanno già effettuato la loro partecipazione.

Parte II

Scelte di progettazione e descrizione delle strutture dati usate.

Properties File:

Le specifiche richiedono che server e client debbano inizializzare le variabili di configurazione autonomamente. Per soddisfare la specifica, creiamo due file, `server.properties` e `client.properties`, contenenti opzioni di inizializzazione delle due classi principali.

User Database:

Per la realizzazione di una struttura dati che contenesse i vari utenti registrati ho creato un file denominato “`database.json`” sulla quale la classe `Utenti` agisce.

La classe `Utenti` fornisce metodi per il controllo e aggiornamento di determinati valori all’interno del file “`database.json`” a seconda delle richieste effettuate dall’utente.

Sostanzialmente i vari metodi della classe, sia per l’aggiornamento, sia per il controllo dei dati, eseguono una deserializzazione del file `.json` con l’obiettivo di prendere le informazioni interessate, aggiornare/controllare i campi di interesse e, nel caso di aggiornamento, scrivere il file aggiornato su “`database.json`”.

Game Database:

Per realizzare un sistema che tiene traccia della parola estratta e degli utenti che effettuano la propria partecipazione relativa alla parola stessa, ho creato un file denominato “game.json” sulla quale la classe WordManager agisce

La classe WordManager è di tipo runnable, il suo metodo run() viene eseguito da uno ScheduledExecutorService denominato scheduledWordSelection definito all’interno di MainWordleServer, il quale pesca periodicamente una nuova parola da indovinare dal file di testo words.txt e la sostituisce alla parola corrente, reinizializzando il file e di conseguenza resettando l’array di giocatori corrente.

Oltre al metodo runnable, la classe contiene metodi usati per la gestione del file “game.json”. Il funzionamento è molto simile a quello usato dalla classe Utenti: iniziamo deserializzando il file, aggiorniamo i campi di interesse e scriviamo il file aggiornato su “game.json”.

Notifiche condivisioni utente:

Le notifiche inviate contengono l’esito dell’ultima partita dell’utente condivisa, il numero di tentativi rimanenti della partita e una stringa di simboli formattata in modo tale che l’utente possa riconoscere i colori normalmente associati alle lettere all’interno di WORDLE.

Parte III

Manuale d'istruzioni.

Compilazione da terminale:

- Spostarsi nella cartella "Terminale".
- Per il client digitare:
 - `javac MainWordleClient.java`
- Per il server digitare:
 - `javac -cp ";gson-2.8.6.jar" MainWordleServer.java`

Esecuzione da terminale dei file compilati:

- Per il client digitare:
 - `java MainWordleClient`
- Per il server digitare:
 - `java -cp ";gson-2.8.6.jar" MainWordleServer.java`

Esecuzione da terminale dei file .jar:

- Per il client digitare:
 - `java -jar WordleClient.jar`
- Per il server digitare:
 - `java -jar WordleServer.jar`

Interazione con il terminale lato client.

Digitare i comandi sulla CLI. I comandi digitati verranno inviati, valutati e, se validi e digitati correttamente, eseguiti. I comandi disponibili verranno visualizzati a schermo ogni volta che lo stato dell'utente cambierà (e.g. Utente loggato, cambio stato e metto a schermo i comandi disponibili ad autenticazione effettuata).

Interazione con il terminale lato server.

Per avviare la terminazione del server bloccato sulla accept(), digitare CTRL+C per avviare la procedura di terminazione del server.

Lista comandi disponibili:

Durante autenticazione:

- register(nome, password)
- login(nome, password)
- exit – Per la chiusura del client.

Durante fase post-autenticazione:

- play – per iniziare una partita di WORDLE.
- mostrastat – per ricevere le statistiche dell'utente aggiornate all'ultima partita giocata.
- condividi – per condividere i propri risultati della partita.
- mostrafeed – per mostrare le notifiche condivise dagli altri giocatori.
- logout – uscita dal profilo attuale.

Durante una partita di WORDLE:

- Digitare una parola di lunghezza 10, se valida riceverà l'indizio, altrimenti verrà emesso un avviso.
- Al termine della partita, ritornerà alla fase post-autenticazione.