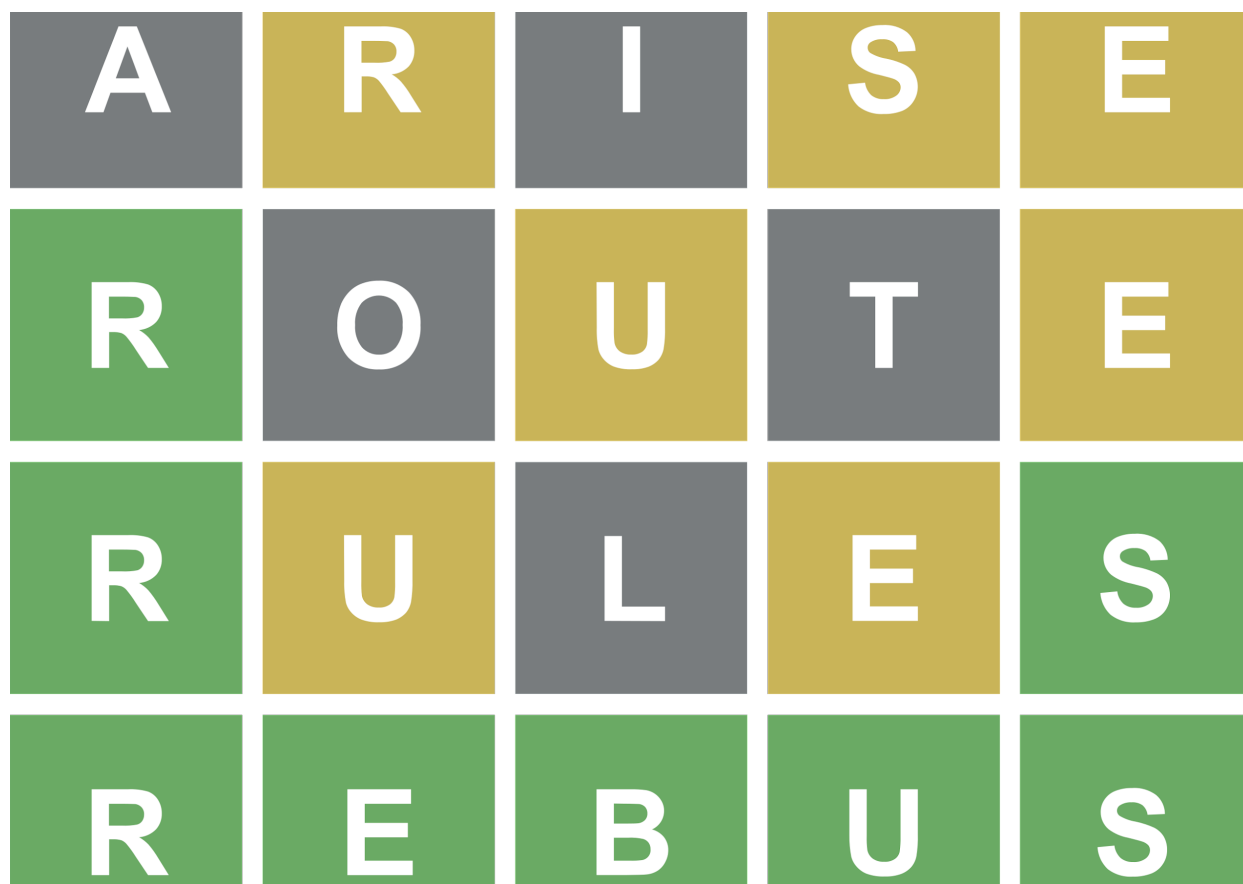


RELAZIONE PROGETTO WORDLE 3.0



Giuseppe Di Stefano 599065

INTRODUZIONE

Wordle 3.0 è una versione semplificata dell'omonimo gioco. In questa versione viene usato un vocabolario di parole in lingua inglese di 10 lettere ed inoltre viene data la possibilità di indovinare la Secret Word con un numero massimo di

tentativi pari a 12.

STRUTTURA DEL PROGETTO

Il progetto si basa su un programma Client-Server dove il Client legge da terminale le richieste fatte dai vari utenti, mandandole al Server su una connessione TCP e il Server risponde, eseguendo ove possibile le azioni richieste dai vari utenti. Le classi principali sono:

- **WordleServerMain:** classe che implementa il server. All'avvio legge il file di configurazione "serverproperties.txt", ripristina la mappa relativa agli utenti andando a leggere il file "raccoltaUtenti.json" sfruttando le funzionalità di gson. Stessa procedura per la classifica andando a leggere "classifica.json". Successivamente apre la connessione e resta in attesa di richieste da parte dei client. All'interno di **WordleServerMain** vengono avviati un **ThreadPool**, un **TimerTask** e un **TerminationThread**. Il **TimerTask** viene usato per estrarre la Secret Word, il **TerminationThread** per terminare l'esecuzione del programma (in quanto il server rimane bloccato sulla accept() in attesa di connessioni) in maniera corretta andando a salvare nei file JSON le strutture dati.
- **Handler:** classe che ha il compito di interagire con l'utente durante la partita. Infatti riceve le richieste dell'utente ed esegue le azioni richieste, comunicando al client l'esito.
- **WordleClientMain:** classe che implementa il client. All'avvio legge il file di configurazione "client.properties.txt" e successivamente tenta di aprire la connessione verso il server. Se tutto è andato a buon fine l'utente può cominciare a digitare i comandi da terminale, ricevendo indietro gli esiti dal server. Viene avviato un **Thread** per la ricezione delle notifiche UDP relative al gruppo sociale.
- **Utente:** classe che implementa il singolo utente.

FASI DI GIOCO E LISTA COMANDI

1. **register:** l'utente può registrarsi a Wordle 3.0 inserendo username e

password. Questa fase viene implementata sfruttando RMI, infatti il client utilizza il metodo **registrazione** di un interfaccia remota.

2. **login:** l'utente effettua l'accesso al proprio account inserendo username e password. Una volta loggato, l'utente lato client joina il gruppo sociale (implementato come gruppo multicast). Inoltre il client si registra a un servizio di notifica del server per ricevere aggiornamenti sulle prime 3 posizioni della classifica. Il servizio di notifica viene implementato con il meccanismo di RMI callback. Quindi viene usata l'interfaccia **NotifyEventInterface** dal server per notificare il client e l'interfaccia **ServerInterfaceCallback** dal client per registrarsi alla ricezione delle notifiche.
3. **logout:** l'utente decide di disconnettersi quindi viene chiusa la socket relativa al gruppo multicast. Inoltre il client non riceve più notifiche relative alla classifica.
4. **play wordle:** l'utente può iniziare la propria sessione di gioco.
5. **send word:** l'utente manda la parola che pensa sia corretta. Se la parola non è di 10 lettere o non esiste nel vocabolario viene avvertito e i tentativi effettuati rimangono sempre gli stessi. Se l'utente indovina la parola: vengono aggiornate le statistiche dell'utente, viene aggiornata la classifica(controllando che le prime 3 posizioni non siano cambiate) e vengono settate delle variabili che permettono al server di comprendere che quell'utente ha giocato per quella parola e deve aspettare che ne esca un'altra. Se l'utente non indovina vengono incrementati i tentativi effettuati e viene fornito l'indizio.
6. **translate:** l'utente una volta terminata la partita può decidere di ricevere la traduzione della Secret Word, grazie al servizio offerto dalla URL <https://mymemory.translated.net/doc/spec.php>.
7. **send me statistics:** una volta terminata la partita vengono mostrate le statistiche aggiornate dell'utente.
8. **share:** dopo aver terminato la partita l'utente può condividere il risultato con il gruppo sociale. La notifica arriva solo ai client con utente loggato.
9. **show me sharing:** con questo comando vengono mostrare le condivisioni fatte dal gruppo sociale.
10. **show me ranking:** viene mostrata la classifica.

11. **help**: viene mostrata la lista dei comandi che l'utente può effettuare.
12. **exit**: spegniamo il client. Quindi chiudiamo tutte le socket precedentemente aperte(lato client), il server rimane sempre attivo in attesa di altri client.

SCELTE IMPLEMENTATIVE

1. **ConcurrentHashMap<String, Utente>** dove andare a salvare i vari utenti. Usata per mantenere un buon livello di concorrenza ed una buona scalabilità
2. **ThreadPoolExecutor** usato per la gestione di più richieste da parte del Server.
3. **RandomAccessFile** per la lettura della Secret Word, all'interno del thread TimerTask che 'pesca' la parola randomicamente. E' utile il meccanismo di puntatore all'interno del file ad accesso randomico, infatti settando il file pointer con una seek che prende un offset random ma con parametri corretti ($0 \leq \text{offset} \leq 30824 * 11 \text{ byte}$), si riesce a leggere una parola di 10 lettere randomica nel file "words.txt".
4. **SynchronizedList<String>** dove andare a salvare le notifiche che si ricevono, dopo aver joinato il gruppo multicast.
5. **AtomicReference** per l'aggiornamento della Secret Word in modo atomico.
6. **SynchronizedSortedSet<Utente>** per implementare la classifica dei giocatori. Essendo sincronizzata ci dà una grossa mano nella gestione della concorrenza ed inoltre questa struttura dati ci permette di mantenere l'ordine della classifica in base al punteggio grazie al Comparator "UtenteComparator".

REGOLE DI GIOCO

- E' obbligatorio essere registrati ed aver effettuato il login.
- Per poter giocare bisogna digitare il comando "**play wordle**".
- Ogni volta che si tenta di indovinare la parola bisogna prima digitare il comando "**send word**".
- Non mettere dello spazio dopo aver digitato i comandi.

- Inserire i comandi **solo** dopo aver ricevuto la risposta dal server.
- Per condividere sul gruppo sociale, si deve aver prima **terminato** una partita.

COMPILAZIONE ED ESECUZIONE

Per compilare recarsi nella cartella **Wordle3.0**. Una volta all'interno della cartella digitare il comando `javac -cp lib/* src/code/*.java src/risorse/*.java -d bin` per la **compilazione** di tutti i file.

Successivamente recarsi nella cartella **bin**.

Digitare il comando `java -cp ".:/../lib/*" WordleServerMain` per l'**esecuzione del server** ed il comando `java -cp ".:/../lib/*" WordleClientMain` per l'**esecuzione del client**.

ESECUZIONE JAR

Per eseguire il file **jar** bisogna recarsi nella cartella **bin**.

Dopodiché digitare il comando `java -jar WordleServer.jar` per l'**esecuzione del server** ed il comando `java -jar WordleClient.jar` per l'**esecuzione del client**.