

# Laboratorio di Reti

## WORDLE: un gioco di parole 3.0

### Progetto di Fine Corso A.A. 2022/23

#### 1. Descrizione del gioco

Il progetto consiste nella implementazione di WORDLE, un gioco di parole web-based, divenuto virale alla fine del 2021. In questa sezione descriveremo le regole generali del gioco, in quella successiva le specifiche della versione di WORDLE da implementare.

Il gioco consiste nel trovare una parola inglese formata da 5 lettere, impiegando un numero massimo di 6 tentativi. WORDLE dispone di un vocabolario di parole di 5 lettere, da cui estrae casualmente una parola SW (Secret Word), che gli utenti devono indovinare. Ogni giorno viene selezionata una nuova SW, che rimane invariata fino al giorno successivo e che viene proposta a tutti gli utenti che si collegano al sistema durante quel giorno. Quindi esiste una sola parola per ogni giorno e tutti gli utenti devono indovinarla, questo attribuisce al gioco un aspetto sociale. L'utente propone una parola GW (Guessed Word) e il sistema inizialmente verifica se la parola è presente nel vocabolario. In caso negativo avverte l'utente che deve immettere un'altra parola. In caso la parola sia presente, il sistema fornisce all'utente tre tipi d'indizi, per ogni lettera  $l$  di GW, il sistema indica:

- se  $l$  è stata indovinata e si trova nella posizione corretta rispetto a SW
- se  $l$  è stata indovinata, ma si trova in una posizione diversa in SW
- se  $l$  non compare in SW

In WORDLE, questi indizi vengono presentati all'utente colorando le lettere in GW con colori diversi, come mostrato in Fig.1

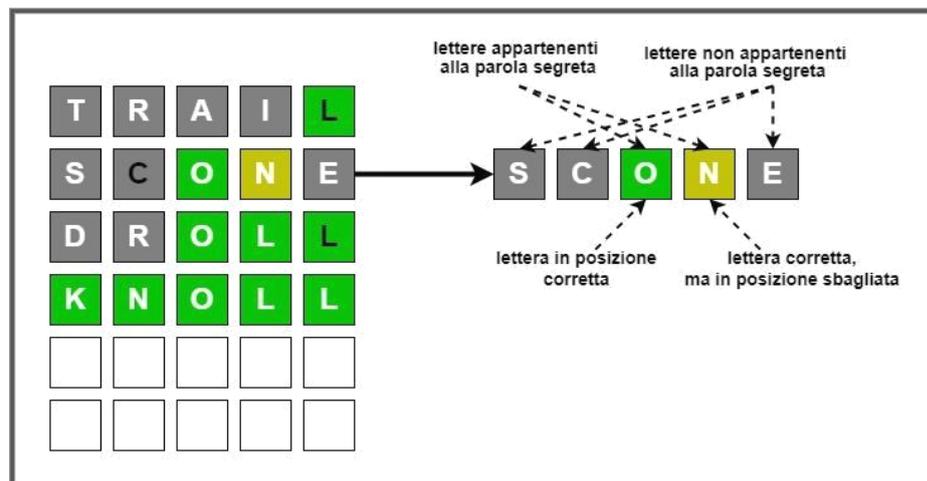


Fig.1 Uno scenario di gioco

L'utente, utilizzando questi indizi, immette un'altra parola. Il gioco termina se l'utente individua la parola segreta oppure se i tentativi sono terminati. Il sistema memorizza, per ogni utente, le seguenti statistiche e le mostra a termine di ogni gioco:

- numero partite giocate
- percentuale di partite vinte

- lunghezza dell'ultima sequenza continua (streak) di vittorie
- lunghezza della massima sequenza continua (streak) di vittorie
- guess distribution: la distribuzione di tentativi impiegati per arrivare alla soluzione del gioco, in ogni partita vinta dal giocatore.

Infine, per incrementare l'aspetto social del gioco, WORDLE fornisce all'utente, a fine di ogni gioco, la possibilità di condividere i tentativi effettuati (sia che il gioco sia terminato con successo o meno), sulle più importanti social network. Per evitare spoiler, vengono condivisi solo i colori delle lettere proposte in ogni tentativo. Ad esempio, per l'esempio riportato in Fig.1, le informazioni che l'utente può condividere sono mostrate in Fig.2. Nell'intestazione è presente il numero del gioco (le parole proposte da WORDLE sono numerate progressivamente) e il numero di tentativi prima della fine del gioco.

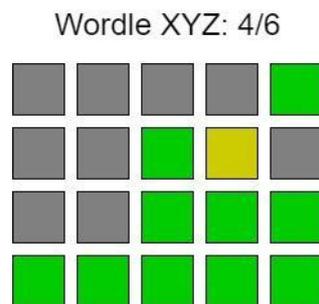


Fig.2: condivisione dei risultati

## 2. Funzionalità

Si richiede d'implementare una versione semplificata di WORDLE, che conservi però la logica di base del gioco. In particolare, la fase di condivisione dei risultati non richiederà l'interazione con una social network, ma sarà realizzata mediante la definizione e l'uso di un gruppo multicast a cui partecipano tutti i giocatori registrati al gioco. Inoltre l'implementazione richiesta utilizzerà un vocabolario di parole di 10 lettere (che verrà fornito) e un numero di tentativi massimo pari a 12.

Distingueremo in seguito l'insieme di funzionalità di base da quelle aggiuntive. L'insieme di funzionalità di base dovranno essere implementate sia dagli studenti del vecchio ordinamento che da quelli del nuovo ordinamento, quelle aggiuntive solo dagli studenti del vecchio ordinamento. Inoltre l'implementazione di alcune funzionalità di base sarà diversa per gli studenti del nuovo rispetto a quelli del vecchio ordinamento, come specificato nella sezione 2.1 e 2.2.

### 2.1 Funzionalità di base (per gli studenti di entrambe gli ordinamenti)

Il gioco deve essere implementato mediante due componenti principali, che interagiscono usando diversi protocolli e paradigmi di comunicazione di rete. Le componenti sono le seguenti:

- **WordleClient.** Gestisce l'interazione con l'utente, tramite una CLI (Command Line Interface) (l'interfaccia grafica è opzionale), comunica con il WordleServer per eseguire le azioni richieste dall'utente. Le principali operazioni sono le seguenti;
  - *register(username, password):* registrazione a WORDLE. L'utente deve fornire username e una password. Il server risponde con un codice che può indicare l'avvenuta registrazione, oppure, se lo username è già presente, o se la password è vuota, restituisce un messaggio d'errore. Lo username dell'utente deve essere univoco. Come specificato in seguito, le registrazioni sono tra le informazioni da persistere lato server.

- *login(username, password)*: login di un utente già registrato per accedere al servizio. Il server risponde con un codice che può indicare l'avvenuto login, oppure, se l'utente ha già effettuato la login o la password è errata, restituisce un messaggio d'errore.
  - *logout(username)*: effettua il logout dell'utente dal servizio.
  - *playWORDLE()*: richiesta d'iniziare il gioco indovinando l'ultima parola estratta dal server. Il server controlla se l'utente ha già partecipato al gioco per quella parola, nel qual caso invia un messaggio di errore, altrimenti il server invia un messaggio in cui indica al client che può iniziare l'invio delle Gussed Word.
  - *sendWord()*: invio da parte del client di una Gussed Word al server. Il server risponde indicando se la parola è presente nel vocabolario e, in questo caso, fornendo gli indizi. Se la parola non è presente, viene inviato un codice particolare, però il tentativo non viene contato tra i 12 tentativi consentiti.
  - *sendMeStatistics()*: richiesta delle statistiche dell'utente aggiornata dopo l'ultimo gioco
  - *share()* richiesta di condividere i risultati del gioco su un gruppo sociale. Come descritto nella sezione 3, il gruppo sociale verrà implementato come un gruppo di multicast
  - *showMeSharing()*: mostra sulla CLI le notifiche inviate dal server riguardo alle partite degli altri utenti.
- **WordleServer**. Gestisce la fase di registrazione e di login degli utenti, memorizza tutti gli utenti registrati, propone periodicamente una nuova parola, interagisce con i diversi client che vogliono partecipare al gioco fornendo gli indizi per la ricerca della parola segreta, gestisce le statistiche degli utenti, provvede, su richiesta dell'utente, a inviare a tutti gli utenti iscritti al gruppo di condivisione, le informazioni relative alla partita conclusa da un utente.

## 2.2 Funzionalità aggiuntive (per gli studenti del vecchio ordinamento)

Gli studenti del vecchio ordinamento dovranno implementare anche le seguenti funzionalità

- **WordleServer**. Il server deve
  - Mantenere una classifica ordinata degli utenti. Il punteggio attribuito a ogni utente è calcolato in funzione del numero di parole indovinate e del numero di tentativi prima della conclusione vittoriosa del gioco. Il calcolo del punteggio per l'inserimento in classifica avviene moltiplicando il numero di partite vinte per il numero medio di tentativi impiegati per raggiungere la soluzione.
  - Quando l'utente termina una sessione di gioco, con successo o con fallimento, deve fornire al client la traduzione italiana della parola segreta, ottenuta accedendo al servizio presente alla URL  
*https://mymemory.translated.net/doc/spec.php*, tramite una chiamata *HTTP GET*.
- **WordleClient**. il client deve
  - Essere avvertito ogni volta c'è un aggiornamento nelle prime tre posizioni della classifica. La classifica viene visualizzata on demand sul client mediante il messaggio *showMeRanking()*

## 3. Specifiche per l'implementazione

Nella realizzazione del progetto devono essere utilizzate molte delle tecnologie illustrate durante il

corso. In particolare:

- L'utente interagisce con WORDLE mediante un client che utilizza un'interfaccia a linea di comando. E' facoltativa l'implementazione di un'interfaccia grafica.
- Fase di registrazione
  - (studenti nuovo ordinamento): questa fase viene implementata instaurando una connessione TCP con il server
  - (studenti vecchio ordinamento) questa fase viene implementata mediante RMI
- Fase di login: deve essere effettuata come prima operazione, dopo che è stata effettuata la registrazione. In ogni sessione di login, ogni utente può tentare d'indovinare più SW, a seconda della durata della session. Ogni tentativo d'indovinare una parola si intende concluso se l'utente esegue il logout.
  - (solo studenti vecchio ordinamento) In seguito alla login il client si registra a un servizio di notifica del server per ricevere aggiornamenti sulla classifica degli utenti. Il servizio di notifica deve essere implementato con il meccanismo di RMI callback. Il client mantiene una struttura dati per tenere traccia delle modifiche avvenute nelle prime posizioni della classifica utenti, che è aggiornata in seguito alla ricezione della callback.
- Dopo previa login effettuata con successo, l'utente interagisce, secondo il modello client-server (richieste/risposte), con il server sulla connessione TCP persistente creata, inviando uno dei comandi elencati nella sezione 2.1. Tutte le operazioni sono effettuate su questa connessione TCP
- Ogni client, dopo la fase di login, si unisce a un gruppo di multicast di cui fa parte anche il server. La condivisione dell'esito di una partita, richiesta dal client mediante il comando *share()*, viene inviata dal server ai client mediante un messaggio UDP su questo gruppo di multicast. Il client deve essere sempre in attesa di questi messaggi di notifica da parte del server e deve memorizzare le notifiche in una sua struttura dati.
- Il server può essere realizzato con JAVA I/O e threadpool oppure può effettuare il multiplexing dei canali mediante NIO (eventualmente con threadpool per la gestione delle richieste). Il server definisce opportune strutture dati per memorizzare le informazioni relative agli utenti e persiste lo stato del sistema. Quando il server viene riavviato tali informazioni sono utilizzate per ricostruire lo stato del sistema. Le informazioni devono essere memorizzate su file utilizzando il formato JSON.
- Il periodo di tempo che intercorre tra la pubblicazione di una parola e la pubblicazione della parola successiva è definito come parametro di configurazione. Si ricorda che un utente può partecipare a più giochi (secret words) durante la stessa sessione di login, ma una volta che ha tentato d'indovinare una secret word (con esito positivo o negativo), deve attendere l'estrazione della successiva secret word per giocare di nuovo.
- La visualizzazione dei suggerimenti del server sulla CLI può essere effettuata associando a ogni colore diverso una diversa lettera (esempio grigio:'X', verde: '+', giallo: '?', si ricorda che il significato dei colori è quello mostrato in Fig.1 e Fig.2)

#### 4. Modalità di svolgimento e di consegna del progetto

Il progetto deve essere eseguito individualmente.

Il materiale da consegnare comprende:

- Il codice dell'applicazione e di eventuali programmi utilizzati per il test delle sue funzionalità.
- la relazione in formato pdf.

Per quanto riguarda il codice, si tenga presente che:

- deve compilare correttamente da riga di comando (ovvero invocando direttamente il compilatore *javac*). In caso contrario, il progetto non verrà considerato valido.
- deve essere ben commentato.
- le classi che contengono un metodo main devono contenere "main" nel nome, es. ServerMain.java; per le altre classi non ci sono vincoli, ma nomi mnemonici sono ovviamente apprezzati.
- oltre al codice sorgente, è necessario consegnare un file JAR eseguibile per ogni applicazione (es. un file JAR per il client e uno per il server).
- i parametri di input delle applicazioni (numeri di porta, indirizzi, valori di timeout, ecc.) devono essere letti automaticamente da appositi file di configurazione testuali da consegnare assieme al resto del codice (due file separati per client e server). Non è consentito leggere i parametri in modo "interattivo" (ovvero facendo in modo che sia il programma a chiederli dopo essere stato avviato).
- in caso di progetti realizzati con Eclipse, IntelliJ IDEA o altri IDE, è obbligatorio consegnare solamente il codice sorgente, rimuovendo eventuali altri file (o directory) creati dall'IDE per gestire il progetto.
- eventuali librerie esterne utilizzate (jar) vanno allegate al progetto.

Per quanto riguarda la relazione, essa deve essere consegnata in formato pdf e deve contenere:

- la definizione delle scelte effettuate nei punti del progetto lasciati alla personale interpretazione
- una definizione delle strutture dati utilizzate sia lato server che lato client
- uno schema generale dei thread attivati sia lato server che lato client
- una descrizione delle eventuali primitive di sincronizzazione utilizzate dai thread per accedere a strutture dati condivise
- una sezione d'istruzioni su come compilare ed eseguire il progetto (librerie esterne usate, argomenti da passare al codice, sintassi dei comandi per eseguire le varie operazioni...). Questa sezione deve essere un manuale d'istruzioni semplice e chiaro per gli utilizzatori del sistema.
- l'organizzazione e la chiarezza della relazione influiranno sul voto finale.

Relazione e codice sorgente devono essere consegnati su Moodle in un unico archivio compresso in formato zip.