

Compito di Introduzione alla Programmazione Web – gennaio 2023

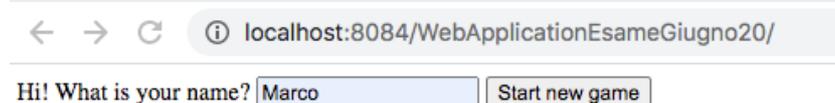
Si richiede di realizzare una versione on-line del gioco “indovina la parola” (descritto nel seguito). Si accede al gioco via web. Il gioco è a coppie, che sono gestite dal server web che deve essere in grado di gestire tante coppie simultanee.

Il gioco consiste nel cercare di indovinare una parola di 10 lettere scelta a caso dal server a partire da un vocabolario predefinito e non noto ai giocatori. Inizialmente nessuna lettera della parola da indovinare è nota, e tutte le lettere della parola sono sostituite da un underscore. Durante il gioco il giocatore propone una lettera: se questa è presente nella parola da indovinare, tutte le occorrenze di quella lettera sono rese visibili. Ad esempio, se la parola da indovinare è ACCOCCOLAI, inizialmente viene mostrata come _____ (10 underscore). Se il giocatore propone la lettera C (maiuscole e minuscole sono considerate equivalenti), la parola di indovinare viene mostrata come _CC_CC_____.

Il punteggio del giocatore, che è zero inizialmente, viene aumentato di 10 per ogni lettera indovinata, diminuito di 1 per ogni lettera errata, e aumentato di 50 nel caso di parola indovinata.

Inizio:

Quando arriva un nuovo utente, gli viene offerta una videata iniziale che chiede il nome dell'utente (obbligatorio) e viene presentato un bottone “start new game”.



Alla pressione del bottone di Start, se esiste già un giocatore con quel nome, il nuovo giocatore viene identificato dal nome stesso seguito da un numero (es. Marco1, Marco2 ecc.).

Per iniziare il gioco, il primo giocatore deve attendere che giunga un avversario. Gli viene quindi mostrata una pagina che dice “Attendi”, e che ogni secondo prova a verificare se nel frattempo la situazione è cambiata (ovvero se un nuovo giocatore è arrivato).

Quando arriva un secondo giocatore, la coppia di avversari viene formata ed il gioco inizia. (Eventuali altri giocatori verranno accoppiati per altre sessioni di gioco).

A questo punto ciascun giocatore vede nel browser:

- Il proprio nome e quello dell'avversario (Ciao Marco, giochi contro Luca2).
- lo stato della parola da indovinare (con le lettere note, e degli underscore a sostituire quelle incognite, come discusso prima),
- un campo tramite il quale il giocatore può tentare di indovinare una lettera
- un campo tramite il quale il giocatore può tentare di indovinare l'intera parola
- la lista delle lettere già tentate dal giocatore stesso ma non corrette (non vengono mostrati i tentativi dell'avversario).
- il punteggio suo, e quello dell'avversario (entrambi posti a zero).

Quando il giocatore sottomette un tentativo (lettera o parola) la pagina viene ricaricata nel nuovo stato ottenuto a seguito del tentativo.

Inoltre la pagina verifica ogni k secondi se nel frattempo lo stato è cambiato a causa di azioni dell'avversario (questi potrebbe aver variato il proprio punteggio provando a indovinare altre lettere o la parola): se vi sono state variazioni la pagina si aggiorna. Il parametro k è settabile nella Webapp senza che questa debba essere ricompilata. (Suggerimento: ogni cambiamento di stato del gioco comporta una variazione di punteggio: quindi controllando se il punteggio dell'avversario è cambiato possiamo scoprire che vi è stata una variazione causata da una sua azione).

Quando la parola viene indovinata, o l'ultima lettera mancante viene scoperta, lo stato del gioco passa alla fase terminale: ad entrambi i giocatori viene mostrata la parola indovinata, viene proclamato vincitore il giocatore con il punteggio più elevato, e appare un bottone "start new game" che ha lo stesso ruolo di quello con la uguale dicitura discusso all'inizio.

Non ci si preoccupi di gestire i casi in cui un utente abbandona il gioco.

Suggerimento: per gestire la business logic relativa alla parola da indovinare, si utilizzi la classe Word scaricabile da <https://latemar.science.unitn.it/esameWeb/Word.java>.

La classe fornisce;

- un costruttore Word(), che sceglie a caso la parola da indovinare da un dizionario predefinito
- un metodo String getShownWord(), che restituisce la parola da indovinare sostituendo con _ tutte le lettere non note.
- un metodo boolean guessLetter(char letter) che verifica se la lettera proposta è tra quelle ancora da indovinare, restituendo falso se la lettera non è tra quelle mancanti e vero altrimenti: in questo secondo caso aggiunge la lettera proposta a quelle note (ma poi il programmatore dovrà chiamare getShownWord per avere la parola aggiornata)
- Un metodo boolean guessWord(String k) che verifica se k è la parola da indovinare.
- Un main di test che ne mostra il funzionamento.

La classe considera equivalenti le maiuscole e le minuscole.

Per testare il codice si dovrà fare ricorso a due browsers.

Possibile traccia di esecuzione (potete decidere di strutturare altrimenti il vostro codice, ma questo è un suggerimento per una possibile implementazione:)

Si definisca una classe Gioco avente come campi nomeGiocatore1, nomeGiocatore2, punteggioGiocatore1, punteggioGiocatore2, word, un booleano che dica se il Gioco può iniziare (perché entrambi i giocatori sono arrivati), un booleano che dica se il gioco è terminato: per ogni coppia di giocatori si manterrà nella WebApp un'istanza di questa classe.

Una waiting room sarà vuota (null) se nessun giocatore è in attesa, e conterrà una istanza di Gioco (incompleta) se vi è un giocatore in attesa dell'arrivo di un avversario.