

## Introduzione alla Programmazione Web – Appello di Gennaio '22

Un gioco consiste nella sfida tra due utenti (ciascuno agente su un diverso browser). Lo scopo è di essere il primo a comporre un convoglio assegnato estraendolo a caso da un pool predefinito.


Il pool contiene varie istanze di diversi tipi di rotabile, ciascuno con un dato valore:

TIPO	Valore	Numerosità nel pool	Immagine
Loco	25	5	
WRz	20	5	
Az	15	10	
Bz	10	15	
Dz	5	5	

- 1) Un utente che accede al sistema effettuando un login con username riceve una dotazione iniziale di 100 fiorini ed un obiettivo che corrisponde alla composizione di un convoglio, scelto in modo casuale tra le seguenti:

Aurora	Loco+Az+Az+Bz+Bz
Colosseum	Loco+Az+WRz+Bz+Bz+Dz
Mediolanum	Loco+Az+Az+Az
Celerone	Loco+Loco+Dz+Dz+Dz

- 2) L'utente collegato vede una pagina che dichiara la composizione da realizzare, e i soldi disponibili. Può effettuare una sola azione premendo uno dei bottoni disponibili. Inizialmente è presente e abilitato solo il bottone "Acquista. C'è un link di Logout per abbandonare il gioco.

 localhost:8084/EsameGennaio22/CreateSession?username=marco

**connected as marco - session id: 750D5AEF34C5E0293EB736FBE32B6AAC - [Logout](#)**

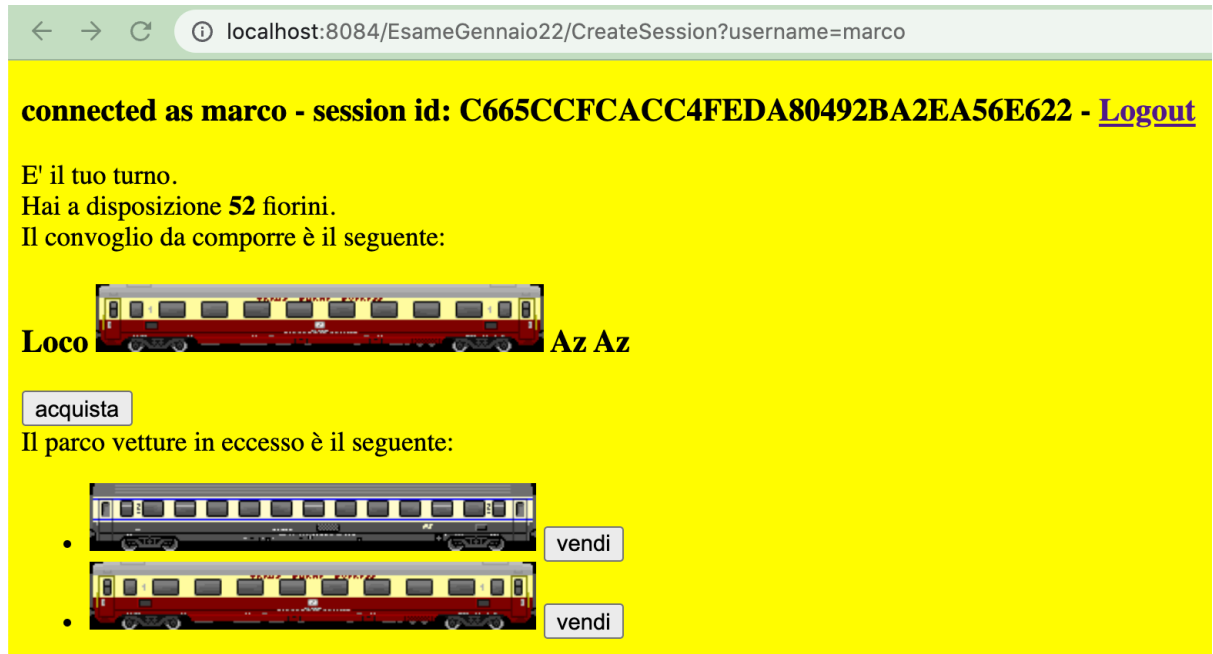
E' il tuo turno.  
Hai a disposizione **100** fiorini.  
Il convoglio da comporre è il seguente:

**Loco Az Az Az**

Il parco vetture in eccesso è il seguente:

- 3) Il sistema accetta il login di due giocatori, e poi non ne accetta altri fino al termine di una partita, dando ad eventuali ulteriori richieste di login un messaggio che dice "Gioco occupato, riprova più tardi".

- 4) “Acquista” ottiene dal banco un rotabile a caso, pagandolo 16 fiorini (indipendentemente da quello che si ottiene) che vengono detratti dalla dotazione. Il banco assegna il rotabile estratto al giocatore, e lo rimuove dal pool. Se c’è un posto in composizione per il rotabile, questo viene posizionato e visualizzato lì, altrimenti viene posto in una lista “Rotabili in eccesso”. Ciascuno dei rotabili nella lista ha a fianco un proprio bottone “Vendi”.



- 5) Dopo un’azione di acquisto o vendita causata dalla pressione dei relativi bottoni, la pagina si aggiorna:
- tutti i bottoni vengono disabilitati
  - il controllo passa all’altro utente.
  - La pagina inizia un polling verso il server, lanciando una richiesta ogni due secondi. Il server risponde con un “NO” finché l’altro utente non termina il suo turno, e con un “OK” quando si può procedere, un “HAI VINTO” se l’altro utente si è ritirato facendo logout, o un “HAI PERSO” se l’altro utente ha completato la propria composizione.
    - Se la risposta è “HAI VINTO” o “HAI PERSO”, questa viene notificata all’utente che terminerà con un logout.
    - Se la risposta è “NO” nulla cambia e il polling continua.
    - Se la risposta è “OK” tutti i bottoni vengono riabilitati, ma se la dotazione non contiene almeno i 16 fiorini necessari per acquistare un rotabile, il bottone “Acquista” rimane disabilitato.
- 6) Il tasto “Vendi” restituisce al pool il corrispondente rotabile, e all’utente viene accreditato il valore dello stesso. Può quindi accadere che un utente abbia un tesoro di valore superiore a quello iniziale, dato che l’acquisto avviene ad un prezzo standard (16 fiorini) mentre la vendita avviene al prezzo corrispondente al valore del rotabile. Ovviamente il rotabile venduto sparisce dalla lista “Rotabili in eccesso”.

- 7) Quando la composizione è completa, al giocatore viene mostrata una pagina “Hai vinto”, mentre l’altro giocatore riceverà tramite il polling la notifica “Hai perso”. Le sessioni dei due utenti vengono distrutte, il pool torna allo stato originario, e vengono accettati due nuovi partecipanti al gioco.

NOTA: Le immagini (di tipo bmp) sono scaricabili da [latemar.science.unitn.it/EsameWeb/disegniFS.zip](http://latemar.science.unitn.it/EsameWeb/disegniFS.zip). Unzipate e mettete la directory disegniFS che contiene i files grafici in WebPages. Assicuratevi che il filtro non intercetti le immagini.

A questo punto potete includere le immagini con `` ecc.

## Possibile traccia per la soluzione

NOTA: il compito è molto lungo, non mi aspetto che lo completiate. È però importante che ne realizziate almeno il motore di controllo, che è il cuore dell'applicazione. Potete procedere come segue:

- 1) La webApp tiene:
  - a. una lista con gli utenti attivi (ad esempio una mappa che associa username e session). Tale lista è raggiungibile da tutte le servlet.
  - b. l'identificazione dell'utente di turno (inizialmente in primo che si registra).
- 2) Si definisca:
  - a. Una pagina **login.html** che richiede lo username, e che rimanda ad una servlet di validazione (chiamiamola **CreateSession**)
  - b. La servlet **CreateSession**:
    - i. produce:
      1. Un messaggio di errore se la sessione esiste già
      2. Un messaggio di errore se la webapp ha già due utenti attivi
      3. Un messaggio di errore se lo username scelto è già tra gli utenti attualmente attivi
    - ii. Se non vi è alcun errore crea una sessione, e aggiunge alla lista utenti attivi username e sessione. Controlla se il turno è già definito: se non lo è setta il turno uguale allo username (quindi il primo login prende il turno).
    - iii. Richiama (per ora uno scheletro de) la pagina principale del gioco, nel banner della quale scrive lo username e il sessionId, un link "logout" che richiama la servlet **InvalidateSession**.  
Il corpo della pagina mostra (per ora solo) il turno.
  - c. Un **filtro** che intercetti tutto, eccetto la pagina login.html e la servlet **CreateSession**, e che se non esiste una sessione rimanda a login.html. Nota: per gestire le pagine che possono bypassare il filtro, si consiglia di controllare se la path della pagina chiamante (`httpRequest.getServletPath()`) è una di quelle permesse, e in tal caso non effettua i controlli che possono rimandare alla pagina di login.
  - d. La servlet **InvalidateSession**:
    - i. rimuove l'utente corrente dalla lista degli utenti attivi
    - ii. se vi è un altro utente attivo, setta il turno a lui, se non ci sono altri utenti attivi setta il turno a null.
    - iii. Invalida la sessione
    - iv. Dà il messaggio "Ciao *username*, sei stato scollegato".
- 3) Si testi la struttura così definita, verificando il funzionamento nei vari casi. Per lavorare con due diversi utenti si usino due diversi browser. Per fare un tentativo di collegamento con un terzo utente si può usare una "Incognito Window" da Chrome, che mantiene cookies separati da quelli delle finestre principali.  
In particolare, si verifichi che:
  - a. Inizialmente chiamando index.jsp o index.html si finisce alla pagina di login

- b. Collegandosi con uno username si entra, e si giunge alla pagina principale del gioco che dice che è il nostro turno
  - c. Collegandosi da un altro browser con lo stesso username si ha un messaggio di errore perché il nome è già in uso
  - d. Collegandosi da un altro browser con username diverso si entra e si giunge alla pagina principale del gioco che dice che è il turno dell'avversario
  - e. Collegandosi con un terzo browser (o incognito window) dice che non ci sono posti disponibili
  - f. Facendo "abbandona" sul primo o il secondo browser, ci si può poi ricollegare con un nome non in uso.
- 4) Si arricchisca la pagina di gioco con un bottone "play" che è abilitato quando è il proprio turno, e disabilitato altrimenti. La pressione del bottone "play" passa il turno all'altro giocatore, disabilita se stesso, modifica la scritta relativa al turno e attiva un sistema di polling. Quest'ultimo che controlla ogni due secondi con il server se è tornato il proprio turno, nel qual caso aggiorna la scritta relativa al turno e riabilita il bottone.
- 5) Si salvi a questo punto il progetto, che andrà consegnato in questo stato.

## SECONDA PARTE

- 6) Da qui in avanti si proceda su una copia del progetto, implementando i dettagli del gioco. Al termine, se avete sviluppato anche la seconda parte, consegnate entrambi i progetti.