

Progetto di laboratorio – Programmazione III - 2015/2016

Si sviluppi un'applicazione java che permette a due utenti di giocare a battaglia navale. L'applicazione deve gestire una sessione di gioco a cui partecipano due giocatori.

- L'interfaccia utente deve offrire una finestra principale per ciascun giocatore, dedicata all'interazione con il sistema, che riassume lo stato del gioco indicando, per ciascuno dei giocatori, quante navi ha a disposizione e, a fine gioco, chi è il vincitore.
- La finestra di gioco (di ciascun giocatore) deve visualizzare:
 - o L'identificatore del giocatore
 - o L'elenco delle navi del giocatore non ancora affondate e l'elenco di quelle affondate
 - o L'elenco delle navi affondate e non dell'avversario.
 - o Un pannello contenente una griglia che visualizza la posizione delle navi del giocatore (affondate e non – indicare visivamente lo stato delle navi)
 - o Un pannello contenente una griglia che visualizza le celle della griglia dell'avversario su cui il giocatore ha già sparato e la posizione delle navi del suo avversario già affondate
 - o Elementi di interfaccia utente che permettono al giocatore di specificare le coordinate a cui sparare per colpire l'avversario
 - o Elementi di interfaccia utente che permettono al giocatore di sapere se, avendo sparato un colpo sull'avversario, ha colpito una nave oppure l'esito è "acqua".
- L'applicazione deve permettere ai giocatori di sparare a turno fino a quando le navi di uno dei due sono state tutte affondate. Al termine del gioco l'applicazione deve indicare chi ha vinto.

Si consideri una sola tipologia di nave, che viene affondata colpendola una sola volta (in altre parole, non si considerino navi più grandi, come l'ammiraglia, che richiedono più attacchi per essere affondate).

Requisiti tecnici:

- L'applicazione deve essere basata su **architettura MVC**, con Controller + viste e Model, seguendo il pattern Observer Observable. Si noti che non deve esserci comunicazione diretta tra viste e model: ogni tipo di comunicazione tra questi due livelli deve essere mediato dal controller o supportata dal pattern Observer Observable.
- L'applicazione deve controllare l'inserimento di input utente e permettere all'utente di correggere eventuali input errati (per es., click del mouse con coordinate al di fuori dello spazio di gioco, click quando non è il turno dell'utente, etc.).
- L'applicazione deve utilizzare **parallelizzare le attività** che non necessitano di esecuzione sequenziale.
- L'applicazione deve essere **distribuita** (in particolare le due finestre per l'interazione sistema-giocatori devono essere gestite in macchine virtuali diverse).

Requisiti dell'interfaccia utente:

- L'interfaccia utente deve essere:
 - Comprensibile (**trasparenza**). In particolare, a fronte di errori, deve segnalare il problema all'utente.
 - Ragionevolmente efficiente per permettere all'utente di eseguire le operazioni con un numero minimo di click e di inserimenti di dati.
 - Deve essere implementata utilizzando il linguaggio Java e in particolare SWING, Thread java e RMI.

NOTE:

- *Si raccomanda di prestare molta attenzione alla progettazione dell'applicazione per facilitare il parallelismo nell'esecuzione delle istruzioni e la distribuzione su JVM diverse.*
- *Si ricorda che il progetto può essere svolto in gruppo (max 4 persone) o individualmente. Se lo si svolge in gruppo la discussione deve essere fatta dall'intero gruppo in soluzione unica. La discussione potrà essere fatta nelle date di appello orale del corso oppure su appuntamento, concordando la data con il docente via email.*