

Dipartimento di Ingegneria e Scienze dell’Informazione e Matematica

CORSO DI LAUREA IN INFORMATICA

Insegnamento Laboratorio di programmazione ad oggetti

Da sostituire con nome dell’applicazione

|  |  |  |  |
| --- | --- | --- | --- |
| **Membri del team** | | | |
| **Cognome e nome** | **Matricola** | | **Indirizzo e-mail** |
| **Ruggieri Alex** | 260640 | | alex.ruggieri@student.univaq.it |
| Rosini Stefano | 258841 | | stefano.rosini@student.univaq.it |
|  | |  | | |
|  | | | | |
|  | |  | | |

A.A. 2022/2023

Sommario

[Capitolo 1 – Caso di studio 2](#_Toc38393257)

[1.1 – Descrizione dell’applicazione 2](#_Toc38393258)

[1.2 - Funzionalità 3](#_Toc38393259)

[1.3 Domain Model 4](#_Toc38393260)

# Capitolo 1 – Caso di studio

## – Descrizione dell’applicazione

SCACCHI INFORMATICI A CHIAMATA

La nostra app di gioco permette di immergersi nel mondo affascinante degli scacchi. Supporta tutte le regole standard degli scacchi, consentendo di giocare contro un avversario virtuale, contro un amico o magari assistere ad una partita tra avversari virtuali!

L’app include una serie di funzionalità per migliorare l’esperienza di gioco. Permette di visualizzare ad ogni turno i pezzi che si possono muovere, visualizzare le mosse disponibili per ogni pedina selezionata, annullare le mosse (fino ad un massimo di 5 in tutta la partita) e di conseguenza ripeterle, rilevare una eventuale situazione di scacco, patta (50 turni tra entrambi i giocatori senza aver mangiato nessuna pedina) e scacco matto. Infine, permette la resa di un giocatore oppure fermare e salvare una partita per poi riprendere da dove la si era terminata.

Altre funzionalità sono offerte durante la scelta della partita da riprendere tra quelle salvate. Permette all’utente di visualizzare in modo decrescente le partite salvate in base al numero di mosse effettuate in ogni partita, visualizzare in modo decrescente le partite salvate in base al numero complessivo di pezzi sulla scacchiera ed infine, visualizzare in modo crescente le partite in base al valore dei pezzi presenti sulla scacchiera.

L’app di gioco degli scacchi in java è stata sviluppata con forte attenzione alla stabilità e alle prestazioni, garantendo una risposta rapida e fluida agli input e un’esperienza di gioco senza interruzioni.

L’app non permette l’utilizzo di mosse speciali, quindi è consigliata per gli utenti principianti!

## 1.2 – Funzionalità

L’app è destinata a due tipi di giocatori, ovvero Utente e CPU. Di seguito, indichiamo le varie funzionalità dell’app del gioco degli scacchi.

* **Partita Utente vs CPU:** il sistema fornisce la possibilità all’utente di poter sfidare un avversario virtuale. Quest’ultimo sceglie pedine ed effettua mosse casuali utilizzando la classe “Random()” fornita dalla libreria “java.util”, rispettando però le regole stabilite del gioco degli scacchi.

* **Partita Utente vs Utente:** il sistema fornisce la possibilità all’utente di poter sfidare un altro utente in locale. Per scegliere la pedina e muoverla, l’utente dovrà inserire degli input sul terminale di comando. Questo viene reso possibile mediante l’utilizzo della classe Scanner() fornita dalla libreria “java.util”.
* **Osservare una partita tra CPU:** il sistema fornisce la possibilità all’utente di poter visualizzare una partita tra avversari virtuali. Come detto in precedenza, questi sceglieranno e muoveranno pedine in modo completamente randomica.
* **Visualizzazione pedine disponibili:** il sistema permette all’utente mediante il suo turno, di visualizzare le pedine disponibile e che presentano dei movimenti validi. Ogni giocatore, ha una lista di pedine a lui associate, mediante l’utilizzo dell’interfaccia List() fornita da “java.util”. L’algoritmo deterministico che permette la visualizzazione, si basa sullo scorrimento della lista associata al giocatore, dove per ogni pedina presente si applica un algoritmo di verifica dei movimenti validi.
* **Visualizzazione movimenti disponibili:** Il sistema fornisce all’utente la visualizzazione da terminale delle mosse disponibili di una determinata pedina scelta precedentemente. L’algoritmo deterministico che permette la visualizzazione è basato sull’utilizzo di una “HashBasedTable” fornita dalla libreria “Guava” e una lista che implementa l’interfaccia list che verrà utilizzata come contenitore dei possibili movimenti. L’algoritmo, quindi, rileva la posizione della pedina dalla table ed effettua, in base alla tipologia della pedina scelta, il calcolo di tutte le possibili posizioni valide (rispettando le regole del gioco). Infine, inserisce quest’ultime nella lista che rappresenta l’output dell’algoritmo.
* **Annullamento mosse effettuate:** il sistema fornisce agli utenti la possibilità di annullare l’ultimo movimento effettuato, per un massimo di 5 volte in tutta la partita. Tutti i movimenti effettuati nella partita vengono inseriti in una lista. Per effettuare l’annullamento di tali movimenti viene utilizzato un metodo deterministico che prende in input la lista, restituisce il penultimo elemento, ed elimina l’ultimo elemento della lista.

* Rilevamento situazioni di scacco: Il sistema fornisce la possibilità di controllare se, dopo aver mosso una pedina, si verifica una situazione di scacco. Fondamentalmente questo controllo avviene due volte:

1. Nel momento in cui viene effettuata la mossa da parte dell’utente;
2. Nel momento in cui inizia il turno del giocatore.

L’algoritmo deterministico, in questo caso, prende in input l’ultima posizione del Re (Ricavata dalla table) ed in base ad essa viene richiamato il metodo che verifica le mosse valide di tutti i pezzi che sono presenti all’interno della table e che sono avversari del Re. Se nelle loro posizioni valide è presente la posizione del Re, allora l’algoritmo restituisce un booleano di valore “true” che rappresenta l’output.

* **Rilevamento situazioni di patta:** Il sistema fornisce la possibilità, nel caso in cui si effettuano al più 50 movimenti senza mangiare pedine oppure muovere pedoni, di terminare la partita. Il conteggio della patta avviene mediante l’utilizzo di un contatore statico.

* **Rilevamento situazioni di scacco matto: Il sistema fornisce la possibilità di controllare se, dopo una situazione di scacco, sia presente una situazione di scacco matto. Questo rilevamento avviene controllando la lista delle mosse valide del Re e se quest’ultima è vuota, allora l’algoritmo restituisce un booleano di valore “true”, sancendo quindi la fine della partita.**
* **Resa dell’utente:** Il sistema fornisce agli utenti la possibilità di poter effettuare la Resa ad ogni turno, terminando quindi la partita. Il termine della partita viene gestita mediante l’utilizzo di un’eccezione chiamata “InterruptedException” fornita da java.
* **Salvataggio delle partite:** Il sistema fornisce agli utenti la possibilità di poter salvare la partita ad ogni turno. Il salvataggio viene effettuato mediante l’utilizzo di una classe Container che contiene una lista di partite. Nel momento in cui l‘utente decide di voler salvare la partita, l’algoritmo effettua delle operazioni chiamate “Serializzazione” e “Deserializzazione”, il quale utilizzo viene garantito dall’interfaccia “Serializable” (Implementata da ogni classe interessata), che viene fornita dalla libreria java.io. La Serializzazione consiste nel salvare su un file un oggetto che viene trasformato in una stringa. La Deserializzazione consiste, invece, nell’estrarre la stringa dal file e riconvertirla in un oggetto sul quale poi effettuare le operazioni. L’algoritmo in questione, quindi, effettua prima una Deserializazzione del file, verificando se all’interno sono presenti delle partite. Se lo sono, allora la partita in questione viene aggiunta assieme alle altre, altrimenti viene salvata come prima partita nel file vuoto.
* **Ripresa di una partita tra quelle salvate:** Il sistema fornisce all’utente la possibilita di effettuare il caricamento delle partite. L’algoritmo che implementa la funzionalità effettua una Deserializzazione del file e restituisce la lista di tutte le partite salvate, permettendo quindi la scelta della partita desiderata.
* **Visualizzazione delle partite salvate in base ad un certo ordine:** Il sistema permette poter visualizzare l’elenco delle partite in base a tre criteri di ordinamento: Numero di pezzi nella scacchiera; Numero di mosse effettuate; Punteggio Complessivo. L’implementazione degli ordinamenti viene effettuata utilizzando delle classi che implementano l’interfaccia “Comparator” fornita dalla libreria java.util.

## 1.3 – Domain Model

Immagine che contiene testo, diagramma, Piano, Parallelo

Descrizione generata automaticamente