

# Manuale di Klotski

Juri Farruku, Elton Llacja, Alvise Molino, Tommaso Tonello

June 11, 2023

## 1 Descrizione del progetto

Il progetto è stato sviluppato adottando come pattern architetturale principale il **Model-View-Controller** e consta delle seguenti principali directory:

- **src**, contenente i file sorgente.
- **bin**, contenente i file **.class** che si ottengono dalla compilazione dei sorgenti.
- **test**, contenente i file usati come test per le classi del gioco.
- **junit**, contenente i test report.

Le directory **src** e **bin** sono a loro volta suddivise in sotto-directory, ovvero **model**, composta dai componenti logici del gioco, **controller**, contenente i controllori che eseguono determinate azioni generalmente impartite dall'utente, e la cartella **view**, al cui interno si trova il file che gestisce la parte relativa alla grafica.

Il software presenta diverse funzionalità che sono state sviluppate nei seguenti modi:

- **Configuration**: permette di scegliere una configurazione tra le 3 disponibili.
- **Reset**: permette di resettare la tavola di gioco in qualsiasi momento.
- **Save**: permette di salvare su file di testo uno stato della tabella di gioco e di poterla ripristinare successivamente.
- **Load**: permette di caricare uno stato precedentemente salvato se ci si trova sulla configurazione in cui è avvenuto il salvataggio.
- **Best Move**: risolve la prima configurazione mostrando ogni mossa dall'inizio alla fine.
- **Undo**: ripristina lo stato precedente a quello attuale, diminuendo anche il numero di mosse. Se si effettua un "Load" non si può usare tale funzione per tornare immediatamente indietro, in quanto un caricamento presuppone un'eliminazione della storia passata del gioco.

## 2 Installazione

Per utilizzare il software è necessario scaricare tutti i file contenuti all'interno della directory **KlotSki\_Ecl**, compilare tutti i file sorgente e avviare il software tramite il file all'interno della cartella `src/view`, ovvero **Main.java**. Tuttavia, poichè per la parte grafica sono state utilizzate le librerie di JavaFX, bisogna scaricare ed aggiungere al compilatore Java i file **.jar** che costituiscono il pacchetto `javafx`.

Se si sta usando Eclipse si lascia il seguente tutorial che può risultare utile ai fini di aggiungere le suddette librerie al compilatore Java: <https://pragmaticways.com/how-to-add-javafx-to-eclipse-the-easy-way/>

Il compilatore usato per il progetto è **Java SE 17**, mentre per JavaFX è stata usata la versione **17.0.7**.

## 3 Librerie

Le principali librerie utilizzate per lo sviluppo del software sono le seguenti:

- **java.io**: utilizzata per la lettura e scrittura di/su file tramite classi quali **FileReader** e **FileWriter** e relative funzioni.
- **javafx.scene**: utilizzata in molte circostanze in quanto versatile. Permette di creare la interfaccia grafica visualizzata e di modificarla aggiungendo elementi diversi, cambiandone la composizione grafica(colore, forme ecc).
- **javafx.animation**: utilizzata per creare animazioni che modifichino l'interfaccia grafica utilizzata dall'utente.
- **javafx.util**: contenente diverse utilities e funzionalità utili a sviluppare oggetti e/o classi.
- **javafx.event**: utilizzata per gestire eventi che avvengono sull'interfaccia grafica, ad esempio un tasto premuto o un pezzo della tavola che viene selezionato tramite puntatore mouse.
- **javafx.stage**: utilizzata per creare l'applicazione e perciò rendere visibili gli elementi grafici creati tramite la libreria `javafx.scene`