

Documentation

Target assessment level

Target assessment level of this work is 3.

Specification

What does the program do?

The program

- reads one input from the user, a string in FEN-notation ([wiki](#)) representing the chess position
- makes sure the string represents a valid position, and
 - displays the board in the terminal using chess symbols in unicode ([wiki](#))
 - displays the best move both on the board and as text, as well as the evaluation, according to [stockfish.online](#)

The user supplies the FEN string either as a program argument or, if none is given, from keyboard.

Data format

The input string is of the format

```
"board turn castlingRights enPassantTarget fullMove halfMove"  
rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
```

Spending one minute to check the wikipedia page is recommended.

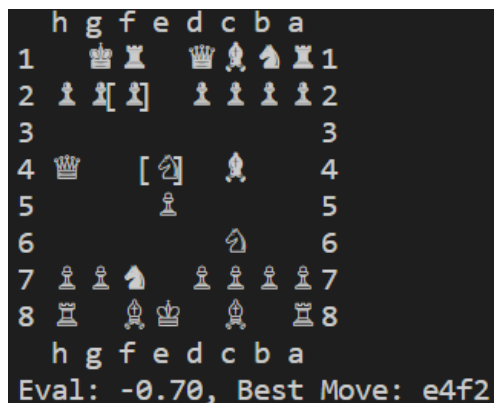
Correctness and exception handling

Typical test case

(1) When the program (file Main.java) is run with

```
java Main.java "r1b1kblr/pppp1Npp/2n5/4p3/2B1n2q/8/PPPP1PPP/RNBQ1RK1 b
kq - 2 6"
```

The output is correct, with the board successfully displayed followed by the stockfish evaluation and the best move of the position:

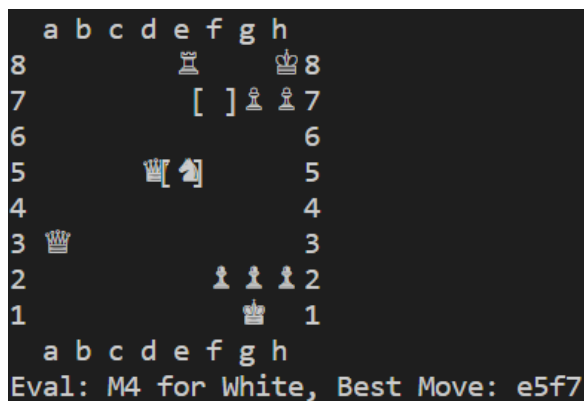


here as an image because I couldn't get monospace to work in google docs

(2) When the program (file Main.java) is run with

```
java Main.java "4r2k/6pp/8/3QN3/8/q7/5PPP/6K1 w - - 0 1"
```

The output is correct, with the board successfully displayed followed by the stockfish evaluation and the best move of the position:



here as an image because I couldn't get monospace to work in google docs

Exception handling (levels 2 and 3)

The following are all the possible exceptions / special cases and the way they are handled.

- The FEN given to the program is not valid: uses the FEN of the chess starting position.
 - The program will always identify a valid FEN as valid, but an invalid FEN with no white king, for example, may also be identified as valid.
- Taking user input fails: treats it as an invalid FEN
- Connection to the API fails: exception caught and printed, program exits.
- Getting data from the connection fails and/or parsing fails: leaves the failed portions empty in the output

Resource management (level 3)

The following resources are opened with try-with-resources -statements and are therefore closed automatically when the program no longer needs them, even in the case of an exception.

- Scanner for system input when reading user-supplied FEN.
- Scanner for connection input when using the API.

Additionally, the connection to the API is closed if no errors occur.

Additional stuff for level 3

- My program does something considerably more complex and sophisticated that would be required.
- The flow of the program is split into meaningful functions called by main().
- The role of defined functions is explained in comments.
- If useful, classes are defined for program-specific data types.