Program Interface

After running the program, you will be in the engine's interface. There are several commands you may use to communicate with the program:

- load startpos
- load fen <string>
- print board
- move <from> <to> <type>
- search <depth>
- perft <depth>
- eval
- help
- exit

Commands: **load startpos**, **load fen <string>, print board**

Currently, there is no position loaded on the board and typing **print board** will output a blank board. To load a position, you may either type **load startpos** to load the start position, or type **load fen <string>** to load any given position. A FEN (Forsyth-Edwards Notation) string is the standard notation used to describe a particular chess position. For example, you may type **load fen r1bqkb1r/1ppp1ppp/p1n2n2/4p3/B3P3/5N2/PPPP1PPP/RNBQK2R w KQkq - 2 5** to load a position in the Ruy Lopez opening. Ensure the FEN string has the proper formatting. Once a position is loaded, typing **print board** will output the board to the screen.

Capitalized letters indicate white pieces, and lowercase letters indicate black pieces. Pieces are represented by their letter: P=pawn, R=rook, N=knight, B=bishop, Q=queen, K=king. Additionally, it is white to move in this position, indicated by the second token of the FEN string “w”.

Commands: **move <from> <to> <type>**

Moving a piece on the board requires the starting square, ending square, and move type. Move type is an indicator of certain “special” moves. Special moves include:

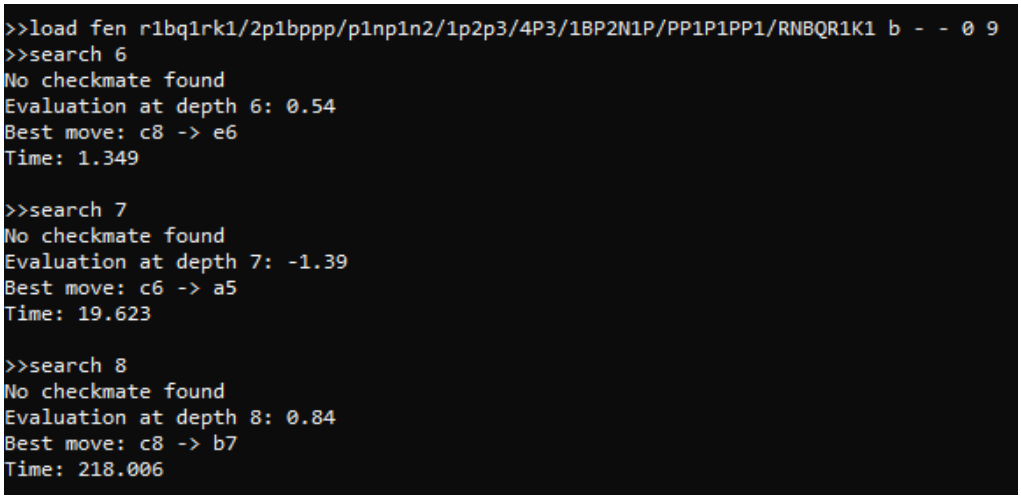
Type 1: Moving a pawn forward two spaces
Type 2: En Passant
Type 3: Castling
Type 4: Promotion to a queen
Type 5: Promotion to a knight
Type 6: Promotion to a bishop
Type 7: Promotion to a rook

If no move type is given, a default move type of 0 is assumed (a normal move). For example, if you wish to move the b1 knight to c3 in the above position, simply type **move b1 c3**. However, if you wish to castle in this position, you must type **move e1 g1 3**. Before making the move, the program will check if the move is legal. Examples of illegal moves are moving into check, moving a white piece when it is black to move, and making a special move without the move type. After making the move, you may type **print board** again to see the new position.

Note: If you wish to see the list of legal moves, type **perft 1**. This command will output the start and end square of all legal moves, but not the move type. This command is described in more depth below.

Commands: **search <depth>**

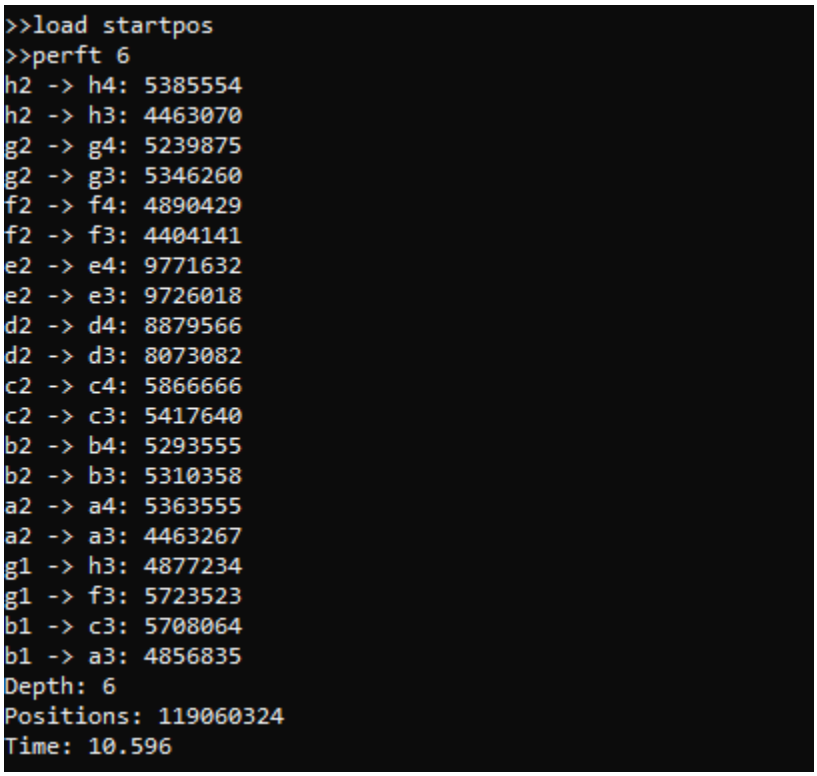
This command provides access to the core functionality of the program. The internal working of this command is complex, and is described in the Search section of the documentation. It is used to find the “best move” in the current position, as determined by the evaluation function. As the depth increases there is a higher chance the engine will find a very good move, but the time it takes to compute the best move increases exponentially. I recommend sticking with a depth of 7. Using depth of 8 may take minutes to compute if in a complicated middle game like the position below.



Note: Using a depth of above 7 or 8 is not recommended as you will most likely need to force quit the program.

Commands: **perft <depth>**

Perft is an abbreviation for performance test. This command calculates the total number of positions that occur after the given number of moves. It is a crucial tool for testing, optimizing, and debugging my program.



The number of positions given by this performance test may be compared to the known values at various depths, providing a quantitative measure for the accuracy of the engine's move generation. The output also provides a list of all legal moves, and the number of positions reached after each move. The total number of positions reached is the summation of the positions reached (at depth minus one) after each move. This enables the pinpointing of incorrect moves and proved highly effective to debug the move generation algorithm.