option

**input** *<filename>*

This command is used to redirect the console input I/O stream from the keyboard to a file. Crafty will then read commands from this file, rather than from the keyboard, and execute them just as though they were typed in. Such a command file \*must\* be terminated by an **"exit"** command (no quotes) as the last command in the file. This reverts the input stream back to the keyboard, and prompts you for another command or move.

This command might be used to configure Crafty for a specific time control, by putting the appropriate time control commands in the file, or to customize the hash table sizes as needed.

**read** | **reada** [*<filename>*]

This command will read input, and extract the chess moves and make them to set up the position at the end of the game. It first resets the chess board to the initial position (read command only) and then extracts the PGN tags (if present) from the front of the input. The rest of the input is parsed for chess moves (comments and similar things are culled automatically) and the moves are made and added to the game history. Once this is done, you can back up, go forward, or play from any point in the game. If you specify a *<filename>* everything is read from the file, otherwise it is read from the console keyboard.

The **reada** command reads moves, but appends them to the current game history/position rather than resetting to the initial chess position. This lets you read in a game, then use reada to manually add some more moves to see the resulting position.

**reset** *<n>*

This command lets you back up in the current game to any move of your choice. **reset** *<n>* backs up the game to move *<n>* with the same side on move. If you want to first change the side to move, use the [**white**|**black**](https://craftychess.com/documentation/craftydoc.html#black%7Cwhite) command, then use the **reset** command to back up to the right move. Note that you can also go forward as well, just so there are moves in the current game history.

# Strategy

1. input console instruction
2. read from game1 to move 24
   1. saves the time it takes to reach the depth with other parameters
   2. the engine writes to a file with java filewriter
   3. the engine then proceeds to the next one
3. The engine repeat step 2 until it reaches game30

# how should look like

input gamelog.txt

# gamelog.txt content

**initiate option settings**

**read** | **reada** [*Game1.txt*]

Make move and search depth x

reset

**read** | **reada** [*Game2.txt*]