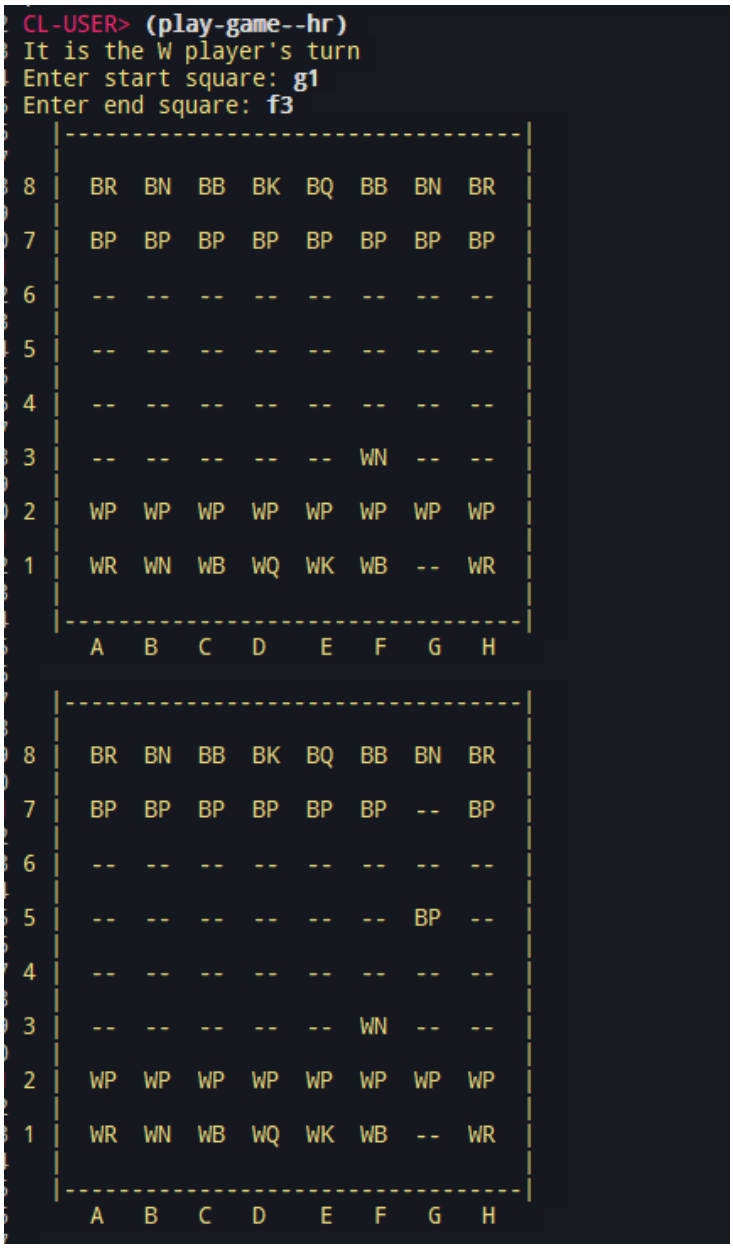
# Chess AI Player Task 14: Create the Game Playing Interface

**Abstract**: The goal of this task was to create a way for a person to enter the moves that they would like to make while playing a game. The program asks for the starting location of the piece that is going to move and then asks for the destination . It checks if the move is allowed then moves the desired piece.

**Demo:**



```
It is the W player's turn
Enter start square: f3
Enter end square: g5
   |---------------------------------|
   |                                 |
8  |  BR  BN  BB  BK  BQ  BB  BN  BR |
   |                                 |
7  |  BP  BP  BP  BP  BP  BP  --  BP |
   |                                 |
6  |  --  --  --  --  --  --  --  -- |
   |                                 |
5  |  --  --  --  --  --  --  WN  -- |
   |                                 |
4  |  --  --  --  --  --  --  --  -- |
   |                                 |
3  |  --  --  --  --  --  --  --  -- |
   |                                 |
2  |  WP  WP  WP  WP  WP  WP  WP  WP |
   |                                 |
1  |  WR  WN  WB  WQ  WK  WB  --  WR |
   |                                 |
   |---------------------------------|
     A   B   C   D   E   F   G   H

   |---------------------------------|
   |                                 |
8  |  BR  BN  BB  BK  BQ  BB  BN  BR |
   |                                 |
7  |  BP  BP  BP  BP  BP  BP  --  -- |
   |                                 |
6  |  --  --  --  --  --  --  --  BP |
   |                                 |
5  |  --  --  --  --  --  --  WN  -- |
   |                                 |
4  |  --  --  --  --  --  --  --  -- |
   |                                 |
3  |  --  --  --  --  --  --  --  -- |
   |                                 |
2  |  WP  WP  WP  WP  WP  WP  WP  WP |
   |                                 |
1  |  WR  WN  WB  WQ  WK  WB  --  WR |
   |                                 |
   |---------------------------------|
     A   B   C   D   E   F   G   H
```

**Code:**

```
( defun bking-in-playp ()
  ( square-in-list-p bking *black-pieces* )
)

( defun wking-in-playp ()
  ( square-in-list-p wking *white-pieces* )
)




( defun parse-square ( square-string )
  ( list
    ( - ( char-code ( aref square-string 0 ) ) ( char-code
#\a ) )
    ( 1- ( parse-integer ( subseq square-string 1 ) ) )
  )
)

( defun play-turn--hh ( color &aux curr-square csr csf
dest-square dsr dsf )
   ( if ( game-overp )
      ( progn
        ( format t "GAME OVER" )
        nil
      )
     ( progn
       ( format t "It is the ~A player's turn~%" color )
        ( format t "Enter start square: " )
       ( setf curr-square ( parse-square ( string-trim " "
(read-line))))
```

```
       ( setf csr ( car curr-square ) )
       ( setf csf ( car ( cdr curr-square ) ) )
       ( format t "Enter end square: " )
       ( setf dest-square (parse-square (string-trim " "
(read-line))))
       ( setf dsr ( car dest-square ) )
       ( setf dsf ( car ( cdr dest-square ) ) )
       ( setf curr-square ( aref ( board *gameboard* ) csf
csr ) )
       ( setf dest-square ( aref ( board *gameboard* ) dsf
dsr ) )
       ( move curr-square dest-square )
       ( play-turn--hh ( oppo-color color ) )
      )
    ( progn
      ( format t "GAME OVER ~A WINS" color )
      nil
     )
  )
)

( defun get-king ( color )
  ( cond
    ( ( eq color 'w ) wking )
     ( ( eq color 'b ) bking )
  )
)

( defun game-overp ()
  ( if ( and ( bking-in-playp ) ( wking-in-playp ) )
     nil
     ( progn
     ( if ( bking-in-playp )
```

```
      ( format t "B PLAYER WINS~%" )
      ( format t "W PLAYER WINS~%" )
    )
    t
    )
  )
)

( defun play-game-hh ()
  ( play-turn--hh 'w )
)
```