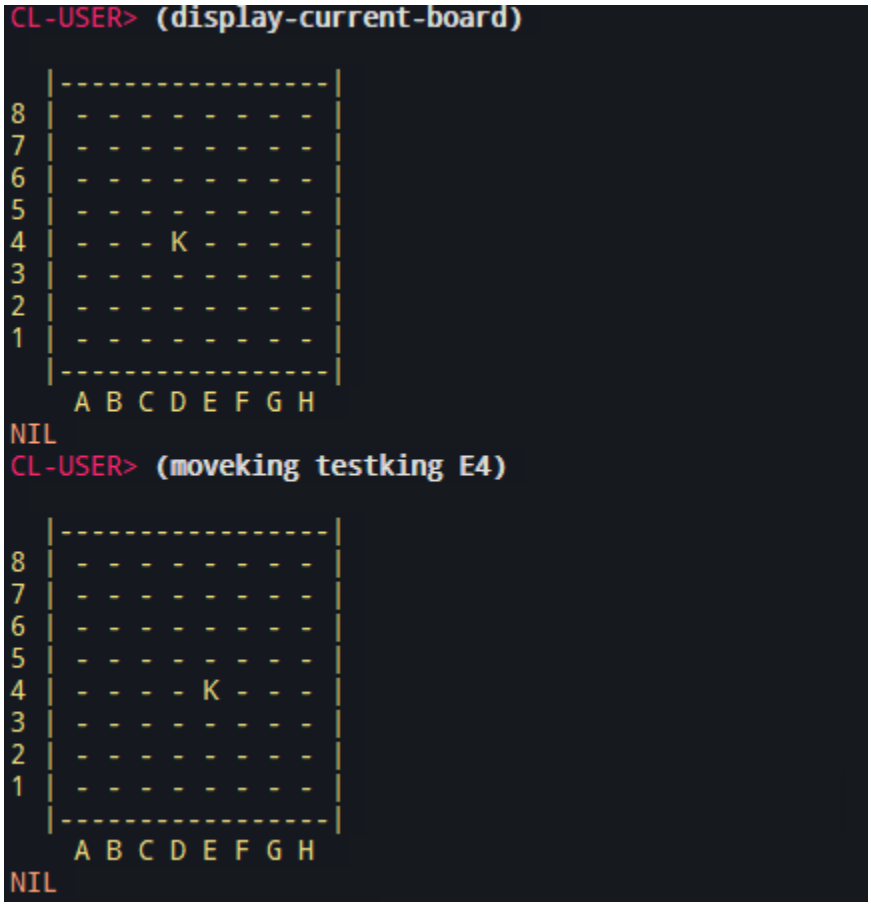
Chess AI Player Task Five: King Piece

Abstract: The goal of this task was to create a king piece and be able to have a user move the king around the board.

Demo:




CL-USER> (moveking testking f3)
|-----------------|
8 | - - - - - - - - |
7 | - - - - - - - - |
6 | - - - - - - - - |
5 | - - - - - - - - |
4 | - - - - - - - - |
3 | - - - - - K - - |
2 | - - - - - - - - |
1 | - - - - - - - - |
|-----------------|
A B C D E F G H
NIL
CL-USER> (moveking testking e3)
|-----------------|
8 | - - - - - - - - |
7 | - - - - - - - - |
6 | - - - - - - - - |
5 | - - - - - - - - |
4 | - - - - - - - - |
3 | - - - - K - - - |
2 | - - - - - - - - |
1 | - - - - - - - - |
|-----------------|
A B C D E F G H
NIL
CL-USER> (moveking testking d4)
|-----------------|
8 | - - - - - - - - |
7 | - - - - - - - - |
6 | - - - - - - - - |
5 | - - - - - - - - |
4 | - - - K - - - - |
3 | - - - - - - - - |
2 | - - - - - - - - |
1 | - - - - - - - - |
|-----------------|
A B C D E F G H
NIL
CL-USER>

Code:

```
(defclass king ()
  ((color
    :initform 'b
    :accessor king-color)
   (current-square
    :initform 34
    :accessor current-square)
   (new-sqaure
    :initform nil
    :accessor desired-sqaure)))


(defun king-legal-move (cs ns)
  (setf king-addition '(-9 -8 -7 -1 1 7 8 9))
  (cond
    ((number-in-list-p (- cs ns) king-addition))))

(setf testking (make-instance 'king))
(setf (current-square testking) 27)
(setf (desired-sqaure testking) nil)
(get-square 27 3)

(defun number-in-list-p (num lst)
  (if (member num lst)
      t
      nil))


(defmethod moveking ((obj king) (x integer))
  (setf (desired-sqaure obj) x)
  (cond
    ((king-legal-move (current-square obj) (desired-sqaure obj))
     (setf (aref the-board (desired-sqaure obj)) 3)
     (setf (aref the-board (current-square obj)) 0)
     (setf (current-square obj) x)
     (display-current-board))
    )
```

```
)
```