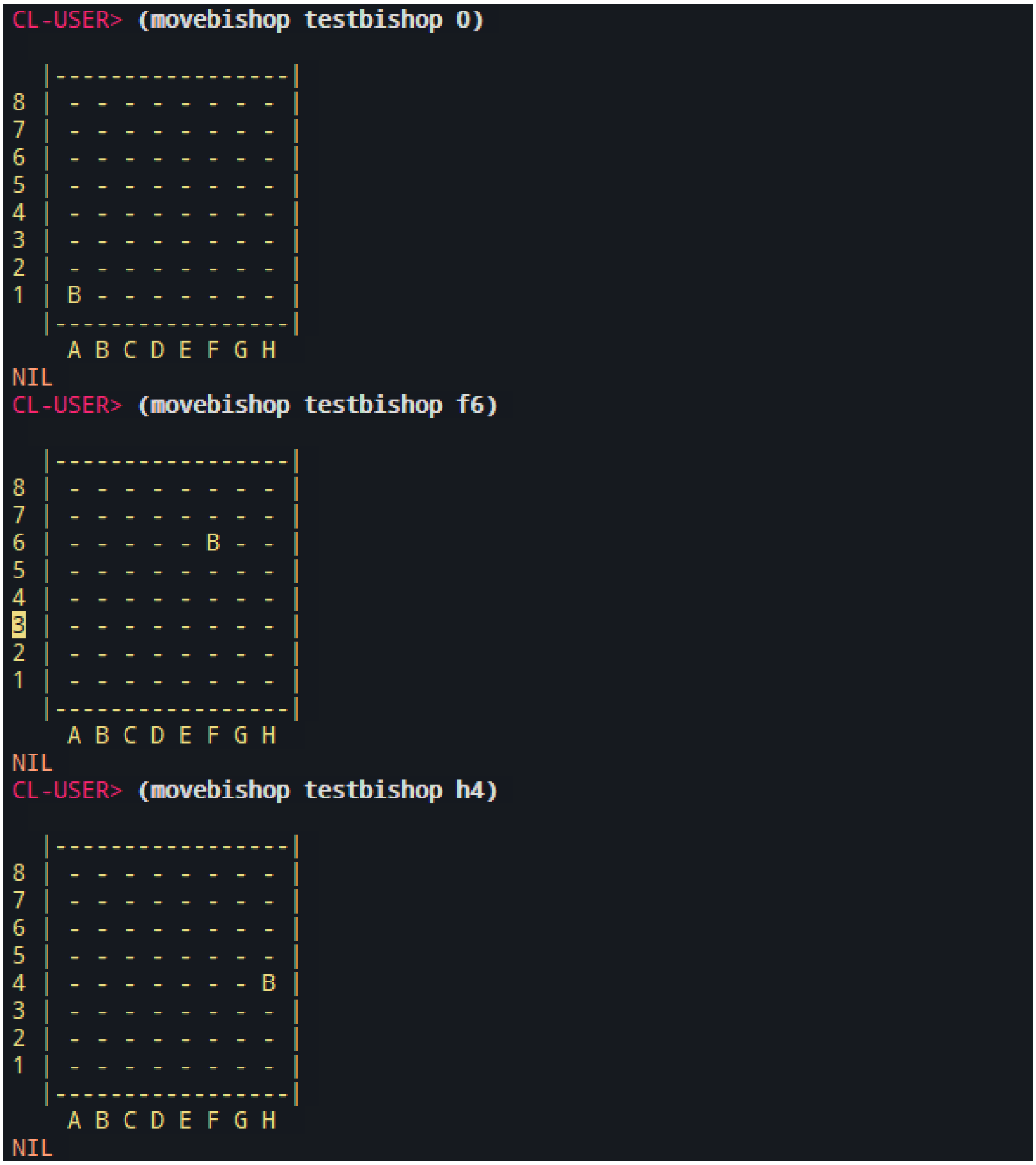
Chess AI Player Task 7: Bishop Piece

Abstract: The goal of this task was to take the created bishop piece and to have a human move the piece around the board. Also I am combining the random moves task to this task so it will also contain moving the piece around the board randomly. The piece by finding the diagonal and anti diagonal that it is currently in. It then randomly picks one of the squares in the combined list of the diagonal and anti diagonal.

Demo:

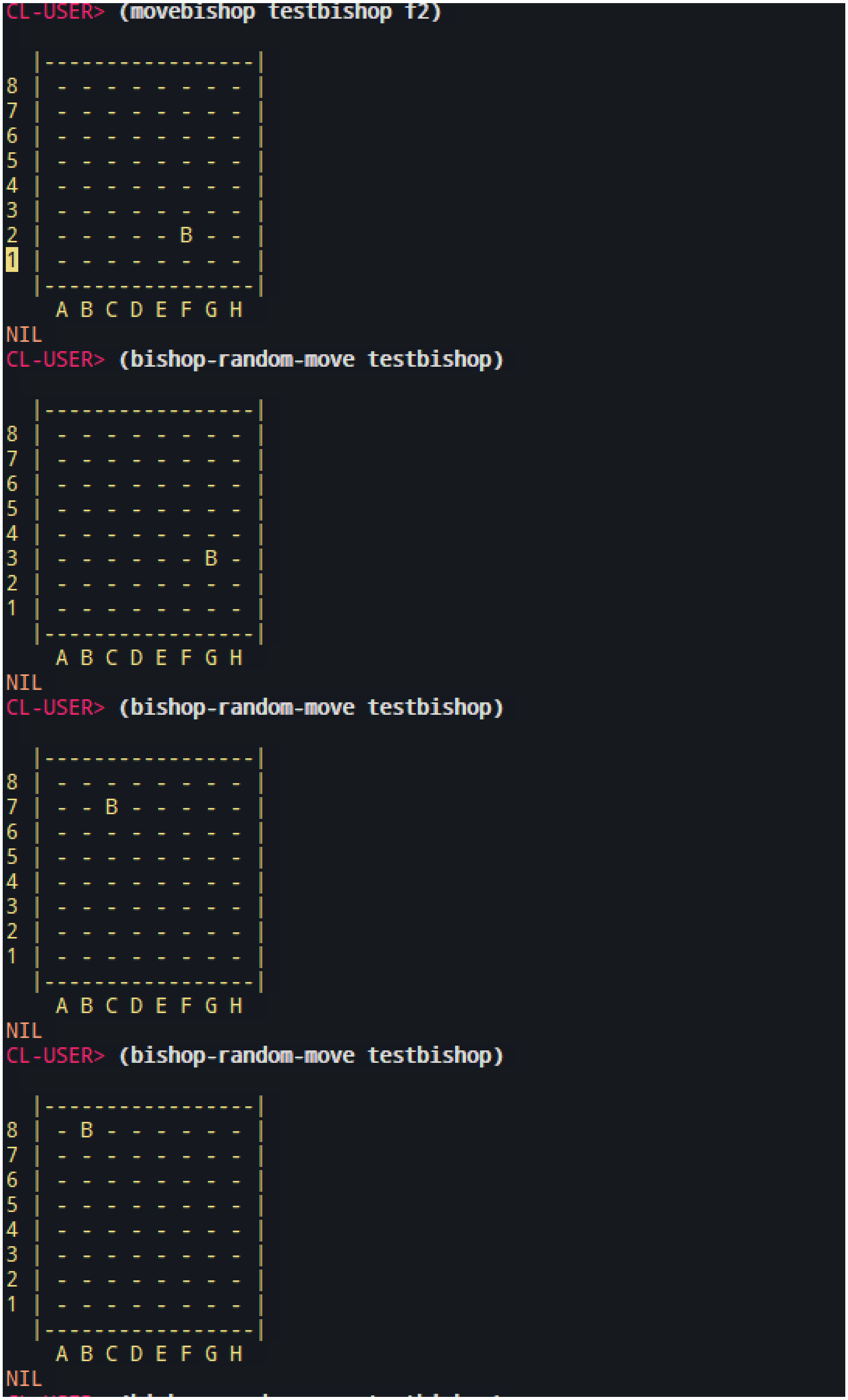


```
CL-USER> (movebishop testbishop 0)

  |-----------------|
8 | - - - - - - - - |
7 | - - - - - - - - |
6 | - - - - - - - - |
5 | - - - - - - - - |
4 | - - - - - - - - |
3 | - - - - - - - - |
2 | - - - - - - - - |
1 | B - - - - - - - |
  |-----------------|
    A B C D E F G H
NIL
CL-USER> (movebishop testbishop f6)

  |-----------------|
8 | - - - - - - - - |
7 | - - - - - - - - |
6 | - - - - - B - - |
5 | - - - - - - - - |
4 | - - - - - - - - |
3 | - - - - - - - - |
2 | - - - - - - - - |
1 | - - - - - - - - |
  |-----------------|
    A B C D E F G H
NIL
CL-USER> (movebishop testbishop h4)

  |-----------------|
8 | - - - - - - - - |
7 | - - - - - - - - |
6 | - - - - - - - - |
5 | - - - - - - - - |
4 | - - - - - - - B |
3 | - - - - - - - - |
2 | - - - - - - - - |
1 | - - - - - - - - |
  |-----------------|
    A B C D E F G H
NIL
```



```
CL-USER> (movebishop testbishop f2)

  |-----------------|
8 | - - - - - - - - |
7 | - - - - - - - - |
6 | - - - - - - - - |
5 | - - - - - - - - |
4 | - - - - - - - - |
3 | - - - - - - - - |
2 | - - - - - B - - |
1 | - - - - - - - - |
  |-----------------|
    A B C D E F G H
NIL
CL-USER> (bishop-random-move testbishop)

  |-----------------|
8 | - - - - - - - - |
7 | - - - - - - - - |
6 | - - - - - - - - |
5 | - - - - - - - - |
4 | - - - - - - - - |
3 | - - - - - - B - |
2 | - - - - - - - - |
1 | - - - - - - - - |
  |-----------------|
    A B C D E F G H
NIL
CL-USER> (bishop-random-move testbishop)

  |-----------------|
8 | - - - - - - - - |
7 | - - B - - - - - |
6 | - - - - - - - - |
5 | - - - - - - - - |
4 | - - - - - - - - |
3 | - - - - - - - - |
2 | - - - - - - - - |
1 | - - - - - - - - |
  |-----------------|
    A B C D E F G H
NIL
CL-USER> (bishop-random-move testbishop)

  |-----------------|
8 | - B - - - - - - |
7 | - - - - - - - - |
6 | - - - - - - - - |
5 | - - - - - - - - |
4 | - - - - - - - - |
3 | - - - - - - - - |
2 | - - - - - - - - |
1 | - - - - - - - - |
  |-----------------|
    A B C D E F G H
NIL
```

Code:

```
(setf d1 '(6 7))
(setf d2 '(5 14 23))
(setf d3 '(4 13 22 31))
(setf d4 '(3 12 21 30 39))
(setf d5 '(2 11 20 29 38 47))
(setf d6 '(1 10 19 28 37 46 55))
(setf d7 '(0 9 18 27 36 45 54 63))
(setf d8 '(8 17 26 35 44 53 62))
(setf d9 '(16 25 34 43 52 61))
(setf d10 '(24 33 42 51 60))
(setf d11 '(32 41 50 59))
(setf d12 '(40 49 58))
(setf d13 '(48 57))

(setf ad1 '(1 8))
(setf ad2 '(2 9 16))
(setf ad3 '(3 10 17 24))
(setf ad4 '(4 11 18 25 32))
(setf ad5 '(5 12 19 26 33 40))
(setf ad6 '(6 13 20 27 34 41 48))
(setf ad7 '(7 14 21 28 35 42 49 56))
(setf ad8 '(15 22 29 36 43 50 57))
(setf ad9 '(23 30 37 44 51 58))
(setf ad10 '(31 38 45 52 59))
(setf ad11 '(39 46 53 60))
(setf ad12 '(47 54 61))
(setf ad13 '(55 62))



(defun get-diagonal (n)
  (cond
    ((member n d1) d1)
    ((member n d2) d2)
    ((member n d3) d3)
    ((member n d4) d4)
    ((member n d5) d5)
    ((member n d6) d6)
    ((member n d7) d7)
    ((member n d8) d8)
    ((member n d9) d9)
    ((member n d10) d10)
    ((member n d11) d11)
    ((member n d12) d12)
    ((member n d13) d13)
    )
  )

(defun get-antidiagonal (n)
  (cond
    ((member n ad1) ad1)
    ((member n ad2) ad2)
    ((member n ad3) ad3)
    ((member n ad4) ad4)
    ((member n ad5) ad5)
    ((member n ad6) ad6)
    ((member n ad7) ad7)
    ((member n ad8) ad8)
    ((member n ad9) ad9)
    ((member n ad10) ad10)
    ((member n ad11) ad11)
    ((member n ad12) ad12)
    ((member n ad13) d13)
    )
  )

(defun bishop-legal-move (cs ds)
  (or (zerop (mod (- cs ds) 9))
      (zerop (mod (- cs ds) 7))))

(defmethod movebishop ((obj bishop) (x integer))
  (setf (desired-square obj) x)
  (cond
    ((bishop-legal-move (current-square obj) (desired-square obj))
     (setf (aref the-board (desired-square obj)) 4)
     (setf (aref the-board (current-square obj)) 0)
     (setf (current-square obj) x)
     (display-current-board))))


(defun get-bishop-tiles (n)
  (append (remove n (get-diagonal n)) (remove n (get-antidiagonal
n))))

(defun bishop-get-move (n)
  (nth (random (length (get-bishop-tiles n))) (get-bishop-tiles n)))

(defmethod bishop-random-move ((obj bishop))
  (movebishop obj (bishop-get-move (current-square obj))))
```