

Chess AI Player Task 9: Pawn Piece

Abstract: The goal of this task was to take the created pawn 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. This piece works by checking if it is a white or a black piece. This determines if the piece is moving up or down the board and depending on the color it either adds or subtracts 8 from the current square to get the new square.

Demo:

```
8 |-----|
7 | - - - P - - - |
6 | - - - - - - - |
5 | - - - - - - - |
4 | - - - - - - - |
3 | - - - - - - - |
2 | - - P - - - - |
1 | - - - - - - - |
  |-----|
  A B C D E F G H
T
CL-USER> (movepawn wpawn c3)

8 |-----|
7 | - - - P - - - |
6 | - - - - - - - |
5 | - - - - - - - |
4 | - - - - - - - |
3 | - - P - - - - |
2 | - - - - - - - |
1 | - - - - - - - |
  |-----|
  A B C D E F G H
NIL
CL-USER> (movepawn bpawn d6)

8 |-----|
7 | - - - - - - - |
6 | - - - P - - - |
5 | - - - - - - - |
4 | - - - - - - - |
3 | - - P - - - - |
2 | - - - - - - - |
1 | - - - - - - - |
  |-----|
  A B C D E F G H
NIL
CL-USER> (movepawn wpawn c4)

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

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

T
CL-USER> (pawn-random-move wpawn)

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

NIL
CL-USER> (pawn-random-move bpawn)

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

NIL
CL-USER> (pawn-random-move wpawn)

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

NIL

Code:

```
(defun white-pawn-legal-move (cs ns)
  (cond
    ((= (- ns cs) 8))))

(defun black-pawn-legal-move (cs ns)
  (cond
    ((= (- cs ns) 8))))

(defmethod movebpawn ((obj pawn) (x integer))
  (setf (desired-square obj) x)
  (cond
    ((black-pawn-legal-move (current-square obj) (desired-square
obj))
      (setf (aref the-board (desired-square obj)) 6)
      (setf (aref the-board (current-square obj)) 0)
      (setf (current-square obj) x)
      (display-current-board))))

(defmethod movewpawn ((obj pawn) (x integer))
  (setf (desired-square obj) x)
  (cond
    ((white-pawn-legal-move (current-square obj) (desired-square
obj))
      (setf (aref the-board (desired-square obj)) 6)
      (setf (aref the-board (current-square obj)) 0)
      (setf (current-square obj) x)
      (display-current-board))))

(defmethod movepawn ((obj pawn) (x integer))
  (cond
    ((eq (color obj) 'b)
      (movebpawn obj x))
    ((eq (color obj) 'w)
      (movewpawn obj x))))

(defmethod bpawn-random-move ((obj pawn))
  (movebpawn obj (- (current-square obj) 8)))
```

```
(defmethod wpawn-random-move ((obj pawn))  
  (movewpawn obj (+ (current-square obj) 8)))
```

```
(defmethod pawn-random-move ((obj pawn))  
  (cond  
    ((eq (color obj) 'b)  
     (bpawn-random-move obj))  
    ((eq (color obj) 'w)  
     (wpawn-random-move obj))))
```