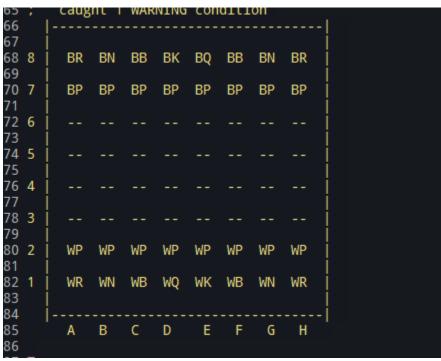# Chess AI Player Task 12: Populate Board with Starting positions

Abstract: The goal of this task is to populate the board with all of the pieces that will be used in the game. This task leaves the pieces at their initial start spot for each game.

Demo:

```
65 ;    caught 1 WARNING condition
66    |------------------------------------|
67    |                                    |
68 8  |   BR  BN  BB  BK  BQ  BB  BN  BR    |
69    |                                    |
70 7  |   BP  BP  BP  BP  BP  BP  BP  BP    |
71    |                                    |
72 6  |   --  --  --  --  --  --  --  --    |
73    |                                    |
74 5  |   --  --  --  --  --  --  --  --    |
75    |                                    |
76 4  |   --  --  --  --  --  --  --  --    |
77    |                                    |
78 3  |   --  --  --  --  --  --  --  --    |
79    |                                    |
80 2  |   WP  WP  WP  WP  WP  WP  WP  WP    |
81    |                                    |
82 1  |   WR  WN  WB  WQ  WK  WB  WN  WR    |
83    |                                    |
84    |------------------------------------|
85        A   B   C   D   E   F   G   H
86
```

Code:

```
( defun make-wpieces ()
    ( setf wrook1 (make-instance 'rook) )
    ( setf wrook2 (make-instance 'rook) )

    ( setf wbishop1 (make-instance 'bishop) )
    ( setf wbishop2 (make-instance 'bishop) )

    ( setf wknight1 (make-instance 'knight) )
    ( setf wknight2 (make-instance 'knight) )

    ( setf wking (make-instance 'king) )
    ( setf wqueen (make-instance 'queen) )

    ( setf wpawn1 ( make-instance 'pawn ) )
    ( setf wpawn2 ( make-instance 'pawn ) )
    ( setf wpawn3 ( make-instance 'pawn ) )
    ( setf wpawn4 ( make-instance 'pawn ) )
    ( setf wpawn5 ( make-instance 'pawn ) )
    ( setf wpawn6 ( make-instance 'pawn ) )
    ( setf wpawn7 ( make-instance 'pawn ) )
    ( setf wpawn8 ( make-instance 'pawn ) )


    ( setf ( cs wrook1 ) a1)
    ( setf ( cs wrook2 ) h1)

    ( setf ( cs wknight1 ) b1)
    ( setf ( cs wknight2 ) g1)

    ( setf ( cs wking ) e1)
    ( setf ( cs wqueen ) d1)
```

```
( setf ( cs wpawn1 ) a2 )
( setf ( cs wpawn2 ) b2 )
( setf ( cs wpawn3 ) c2 )
( setf ( cs wpawn4 ) d2 )
( setf ( cs wpawn5 ) e2 )
( setf ( cs wpawn6 ) f2 )
( setf ( cs wpawn7 ) g2 )
( setf ( cs wpawn8 ) h2 )

( setf ( cs wbishop1 ) c1)
( setf ( cs wbishop2 ) f1)

( setf ( occupier (cs wbishop1) ) wbishop1)
( setf ( occupier (cs wbishop2) ) wbishop2)

( setf ( occupier (cs wknight1) ) wknight1)
( setf ( occupier (cs wknight2) ) wknight2)

( setf ( occupier (cs wrook1) ) wrook1)
( setf ( occupier (cs wrook2) ) wrook2)

( setf ( occupier (cs wking) ) wking)
( setf ( occupier (cs wqueen) ) wqueen)

( setf ( occupier (cs wpawn1) ) wpawn1 )
( setf ( occupier (cs wpawn2) ) wpawn2 )
( setf ( occupier (cs wpawn3) ) wpawn3 )
( setf ( occupier (cs wpawn4) ) wpawn4 )
( setf ( occupier (cs wpawn5) ) wpawn5 )
( setf ( occupier (cs wpawn6) ) wpawn6 )
( setf ( occupier (cs wpawn7) ) wpawn7 )
( setf ( occupier (cs wpawn8) ) wpawn8 )
```

```lisp
    ( setf ( color wpawn1 ) 'w )
    ( setf ( color wpawn2 ) 'w )
    ( setf ( color wpawn3 ) 'w )
    ( setf ( color wpawn4 ) 'w )
    ( setf ( color wpawn5 ) 'w )
    ( setf ( color wpawn6 ) 'w )
    ( setf ( color wpawn7 ) 'w )
    ( setf ( color wpawn8 ) 'w )

    ( setf ( color wrook1 ) 'w )
    ( setf ( color wrook2 ) 'w )

    ( setf ( color wbishop1 ) 'w )
    ( setf ( color wbishop2 ) 'w )

    ( setf ( color wknight1 ) 'w )
    ( setf ( color wknight2 ) 'w )

    ( setf ( color wqueen ) 'w )
    ( setf ( color wking ) 'w )
    ( setf *white-pieces* ( list wking wqueen wknight1
wknight2 wbishop1 wbishop2 wrook1 wrook2 wpawn1 wpawn2
wpawn3 wpawn4 wpawn5 wpawn6 wpawn7 wpawn8 ) )
)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;


( defun make-bpieces ()
    ( setf brook1 (make-instance 'rook) )
    ( setf brook2 (make-instance 'rook) )
```

```
( setf bbishop1 (make-instance 'bishop) )
( setf bbishop2 (make-instance 'bishop) )

( setf bknight1 (make-instance 'knight) )
( setf bknight2 (make-instance 'knight) )

( setf bking (make-instance 'king) )
( setf bqueen (make-instance 'queen) )

( setf bpawn1 ( make-instance 'pawn ) )
( setf bpawn2 ( make-instance 'pawn ) )
( setf bpawn3 ( make-instance 'pawn ) )
( setf bpawn4 ( make-instance 'pawn ) )
( setf bpawn5 ( make-instance 'pawn ) )
( setf bpawn6 ( make-instance 'pawn ) )
( setf bpawn7 ( make-instance 'pawn ) )
( setf bpawn8 ( make-instance 'pawn ) )


( setf ( cs brook1 ) a8)
( setf ( cs brook2 ) h8)

( setf ( cs bknight1 ) b8)
( setf ( cs bknight2 ) g8)

( setf ( cs bking ) d8)
( setf ( cs bqueen ) e8)

( setf ( cs bpawn1 ) a7 )
( setf ( cs bpawn2 ) b7 )
( setf ( cs bpawn3 ) c7 )
( setf ( cs bpawn4 ) d7 )
( setf ( cs bpawn5 ) e7 )
```

```
( setf ( cs bpawn6 ) f7 )
( setf ( cs bpawn7 ) g7 )
( setf ( cs bpawn8 ) h7 )

( setf ( cs bbishop1 ) c8)
( setf ( cs bbishop2 ) f8)

( setf ( occupier (cs bbishop1) ) bbishop1)
( setf ( occupier (cs bbishop2) ) bbishop2)

( setf ( occupier (cs bknight1) ) bknight1)
( setf ( occupier (cs bknight2) ) bknight2)

( setf ( occupier (cs brook1) ) brook1)
( setf ( occupier (cs brook2) ) brook2)

( setf ( occupier (cs bking) ) bking)
( setf ( occupier (cs bqueen) ) bqueen)

( setf ( occupier (cs bpawn1) ) bpawn1 )
( setf ( occupier (cs bpawn2) ) bpawn2 )
( setf ( occupier (cs bpawn3) ) bpawn3 )
( setf ( occupier (cs bpawn4) ) bpawn4 )
( setf ( occupier (cs bpawn5) ) bpawn5 )
( setf ( occupier (cs bpawn6) ) bpawn6 )
( setf ( occupier (cs bpawn7) ) bpawn7 )
( setf ( occupier (cs bpawn8) ) bpawn8 )

( setf ( color bpawn1 ) 'b )
( setf ( color bpawn2 ) 'b )
( setf ( color bpawn3 ) 'b )
( setf ( color bpawn4 ) 'b )
( setf ( color bpawn5 ) 'b )
```

```
    ( setf ( color bpawn6 ) 'b )
    ( setf ( color bpawn7 ) 'b )
    ( setf ( color bpawn8 ) 'b )

    ( setf ( color brook1 ) 'b )
    ( setf ( color brook2 ) 'b )

    ( setf ( color bbishop1 ) 'b )
    ( setf ( color bbishop2 ) 'b )

    ( setf ( color bknight1 ) 'b )
    ( setf ( color bknight2 ) 'b )

    ( setf ( color bqueen ) 'b )
    ( setf ( color bking ) 'b )

    ( setf *black-pieces* ( list bking bqueen bknight1
bknight2 bbishop1 bbishop2 brook1 brook2 bpawn1 bpawn2
bpawn3 bpawn4 bpawn5 bpawn6 bpawn7 bpawn8 ) )
)
```