

## Project Tasks 2 and 3: Establish Consecutive Order of Bowls and Write Code for Single Move

By Carrie Corcoran

This is the second and third tasks for my semester-long project of creating heuristic Mancala players. It involved extensive reworking of the initial board model into a set of property lists.

### Code:

```
;-----Universal Constants and variables

(setf *side-B* '(B6 B5 B4 B3 B2 B1))
(setf *side-A* '(A1 A2 A3 A4 A5 A6))


(setf *home-A* 0)
(setf *home-B* 0)


(setf (symbol-plist 'A1) '(next-for-a A2 next-for-b A2 opposite
B6 value 4))

(setf (symbol-plist 'A2) '(next-for-a A3 next-for-b A3 opposite
B5 value 4))

(setf (symbol-plist 'A3) '(next-for-a A4 next-for-b A4 opposite
B4 value 4))

(setf (symbol-plist 'A4) '(next-for-a A5 next-for-b A5 opposite
B3 value 4))

(setf (symbol-plist 'A5) '(next-for-a A6 next-for-b A6 opposite
B2 value 4))

(setf (symbol-plist 'A6) '(next-for-a AH next-for-b B1 opposite
B1 value 4))


(setf (symbol-plist 'AH) '(next-for-a B1 value 0))


(setf (symbol-plist 'B1) '(next-for-a B2 next-for-b B2 opposite
A6 value 4))

(setf (symbol-plist 'B2) '(next-for-a B3 next-for-b B3 opposite
A5 value 4))
```

```
(setf (symbol-plist 'B3) '(next-for-a B4 next-for-b B4 opposite
A4 value 4))
```

```
(setf (symbol-plist 'B4) '(next-for-a B5 next-for-b B5 opposite
A3 value 4))
```

```
(setf (symbol-plist 'B5) '(next-for-a B6 next-for-b B6 opposite
A2 value 4))
```

```
(setf (symbol-plist 'B6) '(next-for-a A1 next-for-b BH opposite
A1 value 4))
```

```
(setf (symbol-plist 'BH) '(next-for-b A1 value 0))
```

```
;-----Functions to access each space's
value
```

```
(defun B6 ()
  (get 'B6 'value)
)
```

```
(defun B5()
  (get 'B5 'value)
)
```

```
(defun B4 ()
  (get 'B4 'value)
)
```

```
(defun B3 ()
  (get 'B3 'value)
)
```

```
(defun B2 ()
  (get 'B2 'value)
)
```

```
(defun B1 ()  
  (get 'B1 'value)  
)
```

```
(defun A6()  
  (get 'A6 'value)  
)
```

```
(defun A5()  
  (get 'A5 'value)  
)
```

```
(defun A4()  
  (get 'A4 'value)  
)
```

```
(defun A3()  
  (get 'A3 'value)  
)
```

```
(defun A2()  
  (get 'A2 'value)  
)
```

```
(defun A1()  
  (get 'A1 'value)  
)
```

```
(defun AH())
```

```

    (get 'AH 'value)
)

(defun BH()
  (get 'BH 'value)
)

;-----Sets board to start game state
(defun reset-board()
  (setf (get 'BH 'value) 0)
  (setf (get 'B1 'value) 4)
  (setf (get 'B2 'value) 4)
  (setf (get 'B3 'value) 4)
  (setf (get 'B4 'value) 4)
  (setf (get 'B5 'value) 4)
  (setf (get 'B6 'value) 4)
  (setf (get 'AH 'value) 0)
  (setf (get 'A1 'value) 4)
  (setf (get 'A2 'value) 4)
  (setf (get 'A3 'value) 4)
  (setf (get 'A4 'value) 4)
  (setf (get 'A5 'value) 4)
  (setf (get 'A6 'value) 4)
)

;-----Function to add 1 stone to a bowl
(defun add-one-stone (space)
  (setf (get space 'value) (+ 1 (get space 'value))))
)

;-----Recursive function to run a
single sequence (a move can be multiple sequences)

```



```

    (t
      (setf start-space (get space 'next-for-b))
    )
  )
  (setf new-space (single-sequence player start-space start-
value))
  (draw-board)
  (setf new-value (get new-space 'value))
  (cond
    ((eq1 new-space 'ah)
      t
    )
    ((eq1 new-space 'bh)
      t
    )
    ((> new-value 1)
      (single-move player new-space)
    )
    ((and (member new-space *side-A*) (eq1 player 'a))
      (setf opp (get new-space 'opposite))
      (setf (get 'AH 'value) (+ (get opp 'value) (get 'AH
'value))))
      (setf (get opp 'value) 0)
      (draw-board)
      nil
    )
    ((and (member new-space *side-B*) (eq1 player 'b))
      (setf opp (get new-space 'opposite))
      (setf (get 'BH 'value) (+ (get opp 'value) (get 'BH
'value))))
      (setf (get opp 'value) 0)

```

```
        (draw-board)
      null
    )
    (t
      nil
    )
  )
)
```

## Demo:

```
[1]> (load "mancala.l")
;; Loading file mancala.l ...
;; Loaded file mancala.l
T
[2]> (get 'A1 'next-for-a)
A2
[3]> (get 'A1 'next-for-b)
A2
[4]> (get 'A6 'next-for-a)
AH
[5]> (get 'A6 'next-for-b)
B1
[6]> (get 'B3 'next-for-a)
B4
[7]> (get 'B3 'next-for-b)
B4
[8]> (get 'B6 'next-for-a)
A1
[9]> (get 'B6 'next-for-b)
BH
[10]> (get 'A4 'opposite)
B3
[11]> (get 'B3 'opposite)
A4
```

[12]> (draw-board)

| HB | B6 | B5 | B4 | B3 | B2 | B1 |    |
|----|----|----|----|----|----|----|----|
| —  | 4  | 4  | 4  | 4  | 4  | 4  | —  |
| 0  | —  | —  | —  | —  | —  | —  |    |
|    |    |    |    |    |    |    | 0  |
|    | 4  | 4  | 4  | 4  | 4  | 4  |    |
| —  | —  | —  | —  | —  | —  | —  | —  |
|    | A1 | A2 | A3 | A4 | A5 | A6 | HA |

NIL

[13]> (single-move 'a 'a3)

| HB | B6 | B5 | B4 | B3 | B2 | B1 |    |
|----|----|----|----|----|----|----|----|
| —  | 4  | 4  | 4  | 4  | 4  | 4  | —  |
| 0  | —  | —  | —  | —  | —  | —  |    |
|    |    |    |    |    |    |    | 1  |
|    | 4  | 4  | 0  | 5  | 5  | 5  |    |
| —  | —  | —  | —  | —  | —  | —  | —  |
|    | A1 | A2 | A3 | A4 | A5 | A6 | HA |

T

[14]> (single-move 'a 'a1)

| HB | B6 | B5 | B4 | B3 | B2 | B1 |    |
|----|----|----|----|----|----|----|----|
| —  | 4  | 4  | 4  | 4  | 4  | 4  | —  |
| 0  | —  | —  | —  | —  | —  | —  |    |
|    |    |    |    |    |    |    | 1  |
|    | 0  | 5  | 1  | 6  | 6  | 5  |    |
| —  | —  | —  | —  | —  | —  | —  | —  |
|    | A1 | A2 | A3 | A4 | A5 | A6 | HA |

| HB | B6 | B5 | B4 | B3 | B2 | B1 |    |
|----|----|----|----|----|----|----|----|
| —  | 4  | 4  | 5  | 5  | 5  | 5  | —  |
| 0  | —  | —  | —  | —  | —  | —  |    |
|    |    |    |    |    |    |    | 2  |
|    | 0  | 5  | 1  | 6  | 0  | 6  |    |
| —  | —  | —  | —  | —  | —  | —  | —  |
|    | A1 | A2 | A3 | A4 | A5 | A6 | HA |



| HB | B6 | B5 | B4 | B3 | B2 | B1 |    |
|----|----|----|----|----|----|----|----|
| —  | 5  | 5  | 0  | 5  | 5  | 5  | —  |
| 0  | —  | —  | —  | —  | —  | —  |    |
|    |    |    |    |    |    |    | 2  |
|    | 1  | 6  | 2  | 6  | 0  | 6  |    |
| —  | —  | —  | —  | —  | —  | —  | —  |
|    | A1 | A2 | A3 | A4 | A5 | A6 | HA |

| HB | B6 | B5 | B4 | B3 | B2 | B1 |    |
|----|----|----|----|----|----|----|----|
| —  | 5  | 5  | 0  | 5  | 5  | 5  | —  |
| 0  | —  | —  | —  | —  | —  | —  |    |
|    |    |    |    |    |    |    | 2  |
|    | 1  | 6  | 0  | 7  | 1  | 6  |    |
| —  | —  | —  | —  | —  | —  | —  | —  |
|    | A1 | A2 | A3 | A4 | A5 | A6 | HA |

| HB | B6 | B5 | B4 | B3 | B2 | B1 |    |
|----|----|----|----|----|----|----|----|
| —  | 5  | 5  | 0  | 5  | 0  | 5  | —  |
| 0  | —  | —  | —  | —  | —  | —  |    |
|    |    |    |    |    |    |    | 7  |
|    | 1  | 6  | 0  | 7  | 1  | 6  |    |
| —  | —  | —  | —  | —  | —  | —  | —  |
|    | A1 | A2 | A3 | A4 | A5 | A6 | HA |

NIL