Kuncheng Feng
CSC 466 Presentation

Random Player Plus

Abstract

The class “RandomPlayerPlus” is an improved version of “RandomPlayer” playing machine, this player will not explore the same location twice, and upon achieving a hit, it will move all unexplored neighbors into a preferred list, and prioritize hitting them on the next term.

Code

This player keeps track of three more things than the “randomPlayer.l”, an unexplored list, an explored list, and a preferred list.

```
(defclass randomPlayerPlus()
    (
        (name :accessor player-name :initform 'RandomPlayerPlus)
        (thisBoard :accessor player-board :initarg :thisBoard)
        (otherBoard :accessor player-otherBoard :initarg :otherBoard)
        (ships :accessor player-ships :initarg :ships)
        (unexplored :accessor player-unexplored :initarg :unexplored)
        (explored :accessor player-explored :initform nil)
        (preferred :accessor player-preferred :initform nil)
    )
)


; Constructor
--------------------------------------------------------------------------
---------------------
(defmethod newRandomPlayerPlus((this board) (other board) (ships list)
&aux width height unexplored)
    (setf width (board-width this))
    (setf height (length (board-rows this)))
    (setf unexplored (generateAllLocations width height))
    (make-instance 'randomPlayerPlus
        :thisBoard this
```

```
        :otherBoard other
        :ships ships
        :unexplored unexplored
    )
)
```

Like all player instance in my game, they need three big methods:

- playerPlaceShip
- playerOpenFire
- isPlayerDefeated

However only the playerOpenFire method is shown here, as the other two are identical to the ones in “randomPlayer.l”)

When asked to open fire, it grabs the next location, fires at it, and modifies the three lists based on the result (hit/miss).

```
(defmethod playerOpenFire((p randomPlayerPlus) &aux location hit)
   (setf location (getNextLocation p))
   (setf hit (fireAtLocation location (player-otherBoard p)))
   (modifyLists location hit p)
)
```

```
(defmethod getNextLocation((p randomPlayerPlus) &aux unexplored preferred
location)
    ; Gather the player's information
    (setf unexplored (player-unexplored p))
    (setf preferred (player-preferred p))

    ; Get a random location either from the preferred list or the
unexplored list.
    ; On theory the game should end before or when both of them become
empty.
    (setf location nil)
    (if (equal preferred nil)
        (setf location (randomFromList unexplored))
        (setf location (randomFromList preferred))
    )

    location
)
```

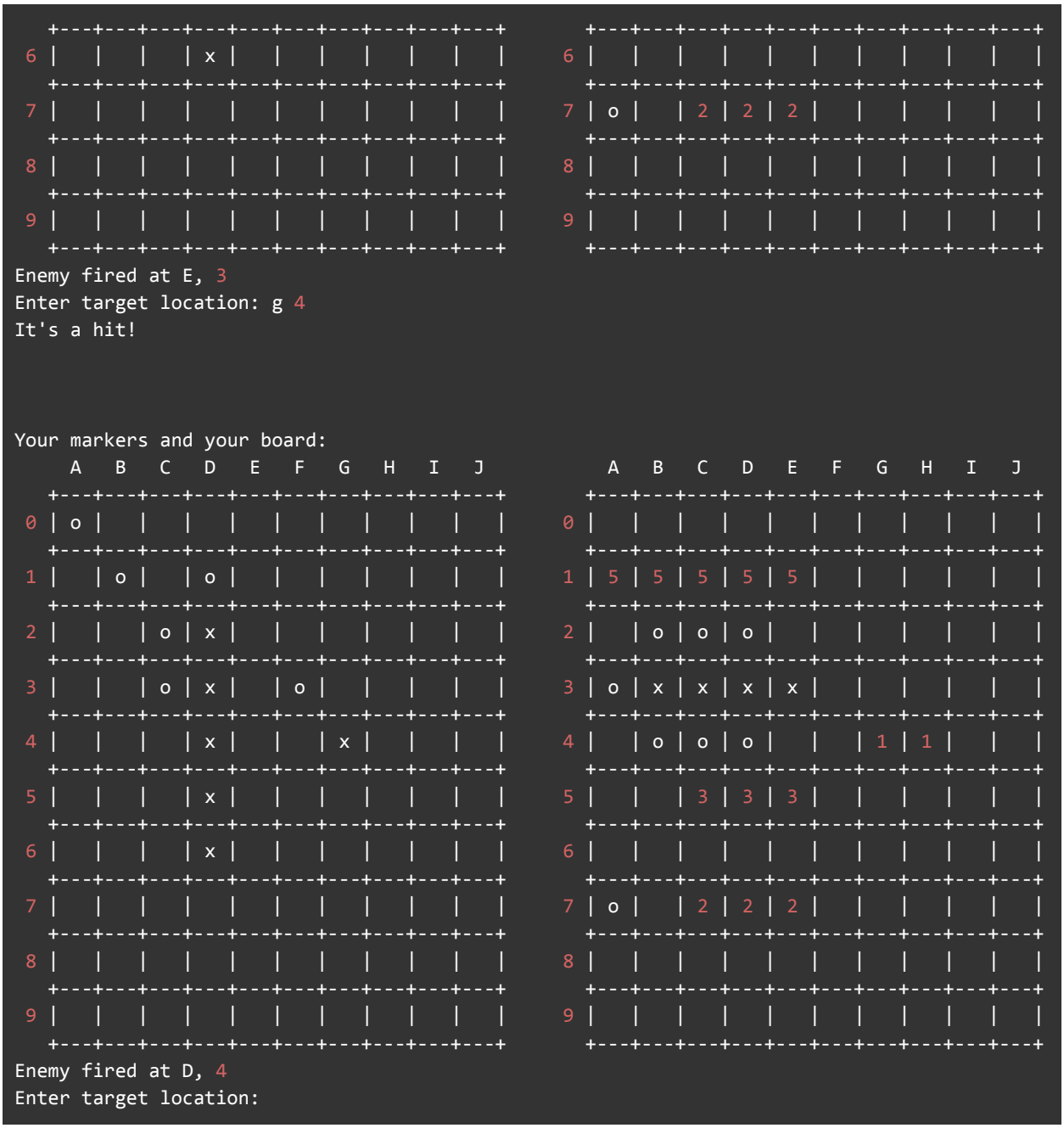
The modifyLists function moves the current location into the explored list, and if it achieved a hit, move all unexplored neighbors into preferred list.

```
(defmethod modifyLists((l location) hit (p randomPlayerPlus) &aux
unexplored explored preferred)
    ; Get the basic list in place
    (setf unexplored (player-unexplored p))
    (setf explored (player-explored p))
    (setf preferred (player-preferred p))

    ; Remove the current location from the (unexplored / preferred) list,
and add it to explored
    (setf unexplored (remove l unexplored))
    (setf preferred (remove l preferred))
    (setf explored (cons l explored))

    ; If the location hit something, move its adjacent locations from
unexplored to preferred.
    ; That means it will ignore the locations that are already in
preferred or explored list.
    (if hit
        (setf preferred (append (getAdjacents l unexplored) preferred))
    )

    ; Write back the modified values
    (setf (player-unexplored p) unexplored)
    (setf (player-explored p) explored)
    (setf (player-preferred p) preferred)

    nil
)
```

Demo

(Snippets of a game i played against)

```
Enter target location: d 6
You sunk a CARRIER.



Your markers and your board:
     A   B   C   D   E   F   G   H   I   J          A   B   C   D   E   F   G   H   I   J
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 0 | o |   |   |   |   |   |   |   |   |   |    0 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 1 |   | o |   | o |   |   |   |   |   |   |    1 | 5 | 5 | 5 | 5 | 5 |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 2 |   |   | o | x |   |   |   |   |   |   |    2 |   | o | o | o |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 3 |   |   | o | x |   |   |   |   |   |   |    3 | o | x | x | x | 4 |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 4 |   |   |   | x |   |   |   |   |   |   |    4 |   | o | o |   |   |   | 1 | 1 |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 5 |   |   |   | x |   |   |   |   |   |   |    5 |   |   | 3 | 3 | 3 |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 6 |   |   |   | x |   |   |   |   |   |   |    6 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 7 |   |   |   |   |   |   |   |   |   |   |    7 | o |   | 2 | 2 | 2 |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 8 |   |   |   |   |   |   |   |   |   |   |    8 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 9 |   |   |   |   |   |   |   |   |   |   |    9 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
Enemy fired at C, 4

Enter target location: f 3
It missed.



Your markers and your board:
     A   B   C   D   E   F   G   H   I   J          A   B   C   D   E   F   G   H   I   J
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 0 | o |   |   |   |   |   |   |   |   |   |    0 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 1 |   | o |   | o |   |   |   |   |   |   |    1 | 5 | 5 | 5 | 5 | 5 |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 2 |   |   | o | x |   |   |   |   |   |   |    2 |   | o | o | o |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 3 |   |   | o | x |   | o |   |   |   |   |    3 | o | x | x | x | x |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 4 |   |   |   | x |   |   |   |   |   |   |    4 |   | o | o |   |   |   | 1 | 1 |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 5 |   |   |   | x |   |   |   |   |   |   |    5 |   |   | 3 | 3 | 3 |   |   |   |   |   |
```



We can see from the demo that the AI was able to concentrate its pattern after it landed a hit, now my battleship (represented with a 4) had been sunk.

Let’s put it to test against other AI:

```
[1]> (load "Main.l")
;; Loading file Main.l ...
;;  Loading file Cell.l ...
;;  Loaded file Cell.l
;;  Loading file Row.l ...
;;  Loaded file Row.l
;;  Loading file Board.l ...
;;  Loaded file Board.l
;;  Loading file Ship.l ...
;;  Loaded file Ship.l
;;  Loading file Location.l ...
;;  Loaded file Location.l
;;  Loading file HumanPlayer.l ...
;;  Loaded file HumanPlayer.l
;;  Loading file RandomPlayer.l ...
;;  Loaded file RandomPlayer.l
;;  Loading file RandomPlayerPlus.l ...
;;  Loaded file RandomPlayerPlus.l
;;  Loading file RandomPlayerPlusPlus.l ...
;;  Loaded file RandomPlayerPlusPlus.l
;;  Loading file TierListPlayer.l ...
;;  Loaded file TierListPlayer.l
;; Loaded file Main.l
T
[2]> (getStatistics)
Available AIs:
1 - RANDOMPLAYER
2 - RANDOMPLAYERPLUS
3 - RANDOMPLAYERPLUSPLUS
4 - TIERLISTPLAYER
Enter a corresponding number to choose AI 1: 1
Enter a corresponding number to choose AI 2: 2
Enter the number of iterations: 100

100 games played:
Player 1 (RANDOMPLAYER) victories: 0
Player 2 (RANDOMPLAYERPLUS) victories: 100
Draws: 0
```

```
NIL
[3]> (getStatistics)
Available AIs:
1 - RANDOMPLAYER
2 - RANDOMPLAYERPLUS
3 - RANDOMPLAYERPLUSPLUS
4 - TIERLISTPLAYER
Enter a corresponding number to choose AI 1: 2
Enter a corresponding number to choose AI 2: 2
Enter the number of iterations: 100

100 games played:
Player 1 (RANDOMPLAYERPLUS) victories: 47
Player 2 (RANDOMPLAYERPLUS) victories: 53
Draws: 0
NIL
[4]>
```

Nice! It was able to completely destroy the braindead random player, and goes evenly against itself.