Kuncheng Feng CSC 466 Presentation

Tier List Player Plus Plus

Abstract:

This player is built upon the "TierListPlayerPlus.l", with the only difference being that when placing ships, it will try to avoid placing them next to each other, as I personally think that it will improve the chances of success.

Code:

The only difference in code:

```
(defmethod playerPlaceShips((player tierListPlayerPlusPlus) &aux board
ships)
  (setf ships (player-ships player))
  (setf board (player-board player))
  ; Loop through the ships that have to be placed
  (loop for ship in ships do
      (noneAdjacentPlaceShip ship board)
  )
; noneAdjacentPlaceShip method can be found in "Ship.l"
```

The new method:



The "getNoneAdjacentPosition" will get the ship size and guess 2 locations, it will keep guessing until the locations are valid



The locations are considered valued if it passes the previous checks, (correct size, not diagonal, not out of bound, not occupied), and if none of the locations in between has a ship that is placed adjacent to it.

```
(defmethod checkNoneAdjacents(x1 y1 x2 y2 b &aux result)
  (setf result t)
  (if (= x1 x2)
    ; Ship is placed vertically
      (loop for y from y1 to y2 do
            (setf result (and result (checkCellAdjacent x1 y b)))
      )
      ; Ship is placed horizontally
      (loop for x from x1 to x2 do
            (setf result (and result (checkCellAdjacent x y1 b)))
      )
      )
      result
}
```

"checkCellAdjacent" method simply check if all neighbors of this cell are either nil or empty, if it's not empty it fail to pass the check.



"notNilAndNotEmpty" method checks what its title suggests.



Demo:

I feel the best way to demo is to compare it with randomly placed ships. A board with randomly placed ships:

```
[6]> (setf ships (generateShips))
(#<SHIP #x1A510F75> #<SHIP #x1A510BA1> #<SHIP #x1A5107CD> #<SHIP
#x1A5103F9> #<SHIP #x1A510025>)
[7]> (setf board (newBoard 10 10))
#<BOARD #x1A52D925>
[8]> (loop for ship in ships do (randomlyPlaceShip ship board))
NIL
[9]> (display board)
        B C
                D
                    Е
                        F
                            G
                                Н
                                    Ι
                                      J
    А
                                     -+--+
   +---+
                         -+--+--+-
                  -+
                      -+-
NIL
```

A board with none of the ships adjacently placed:

```
[10]> (setf ships (generateShips))
(#<SHIP #x1A54C1A9> #<SHIP #x1A54BDD5> #<SHIP #x1A54BA01> #<SHIP
#x1A54B62D> #<SHIP #x1A54B259>)
[11]> (setf board (newBoard 10 10))
#<BOARD #x1A568B5D>
[12]> (loop for ship in ships do (noneAdjacentPlaceShip ship board))
NIL
[13]> (display board)
    A B C D E F
                          G
                             Н
                                 I J
  +---+--+--+--+--
                             ---+--+
0
      1 1
                             5 5
                              -+--+
 2
 5
 6
7
                      4
                             2
 8
                             2
                                      -+
            -+
9
                                      - +
         -+--+-
                    -+--+--+--+-
NIL
```

Let's see how well it play against other Als: Against Random Player:

```
[14]> (getStatistics)
Available AIs:
1 - RANDOMPLAYER
2 - RANDOMPLAYERPLUS
3 - RANDOMPLAYERPLUSPLUS
4 - TIERLISTPLAYER
5 - TIERLISTPLAYERPLUS
6 - TIERLISTPLAYERPLUSPLUS
Enter a corresponding number to choose AI 1: 1
Enter a corresponding number to choose AI 2: 6
Enter the number of iterations: 100
100 games played:
Player 1 (RANDOMPLAYER) victories: 0
Player 2 (TIERLISTPLAYERPLUSPLUS) victories: 100
Draws: 0
NIL
```

Against Random Player Plus:



Against Random Player Plus Plus

Note that even though the RandomPlayer++ can't beat the + version, it fairs better against this one.

[16]> (getStatistics) Available AIs: **1** - RANDOMPLAYER 2 - RANDOMPLAYERPLUS **3** - RANDOMPLAYERPLUSPLUS **4** - TIERLISTPLAYER **5** - TIERLISTPLAYERPLUS **6** - TIERLISTPLAYERPLUSPLUS Enter a corresponding number to choose AI 1: 3 Enter a corresponding number to choose AI 2: 6 Enter the number of iterations: 100 100 games played: Player 1 (RANDOMPLAYERPLUSPLUS) victories: 28 Player 2 (TIERLISTPLAYERPLUSPLUS) victories: 70 Draws: 2 NIL

```
Against Tier List Player
```

```
[17]> (getStatistics)
Available AIs:
1 - RANDOMPLAYER
2 - RANDOMPLAYERPLUS
3 - RANDOMPLAYERPLUSPLUS
4 - TIERLISTPLAYER
5 - TIERLISTPLAYERPLUS
6 - TIERLISTPLAYERPLUSPLUS
Enter a corresponding number to choose AI 1: 4
Enter a corresponding number to choose AI 2: 6
Enter the number of iterations: 100
100 games played:
Player 1 (TIERLISTPLAYER) victories: 31
Player 2 (TIERLISTPLAYERPLUSPLUS) victories: 69
Draws: 0
NIL
```

Against Tier List Player Plus

```
[18]> (getStatistics)
Available AIs:
1 - RANDOMPLAYER
2 - RANDOMPLAYERPLUS
3 - RANDOMPLAYERPLUSPLUS
4 - TIERLISTPLAYER
5 - TIERLISTPLAYERPLUS
6 - TIERLISTPLAYERPLUSPLUS
Enter a corresponding number to choose AI 1: 5
Enter a corresponding number to choose AI 2: 6
Enter the number of iterations: 100
100 games played:
Player 1 (TIERLISTPLAYERPLUS) victories: 53
Player 2 (TIERLISTPLAYERPLUSPLUS) victories: 44
Draws: 3
NIL
```

Against itself

<pre>[19]> (getStatistics)</pre>
Available AIs:
1 - RANDOMPLAYER
2 - RANDOMPLAYERPLUS
3 - RANDOMPLAYERPLUSPLUS
4 - TIERLISTPLAYER
5 - TIERLISTPLAYERPLUS
6 - TIERLISTPLAYERPLUSPLUS
Enter a corresponding number to choose AI 1: 6
Enter a corresponding number to choose AI 2: 6
Enter the number of iterations: 100
100 games played:
Player 1 (TIERLISTPLAYERPLUSPLUS) victories: 50
Player 2 (TIERLISTPLAYERPLUSPLUS) victories: 46
Draws: 4
NIL

Just a side note:

While trying to get the Tier List Plus Plus player to play against itself, sometimes it will get stuck in infinite loops.

Upon hours of debugging, I found out that it simply was taking way too long to place ships on the board, as sometimes it is hard to get consecutive random locations that have no adjacent ships.

However I found a simple fix, under the "getStatistics" function under the "Main.l" file, where I generate the ships, I simply reverse the list of ships generated:



Due to that "generateShips" method returns a list of ship instances, with smaller ship instances at the beginning, it imposes a big challenge for AI as it now needs to place bigger and bigger ships on a board with less and less available spaces.

With the "reverse" the bigger ships instances will be put in front, it will end up being doable to find smaller consecutive spaces on an increasingly smaller board