Kuncheng Feng
CSC 466 progress report

# Improved Display Method

## Abstract

In consideration of the feedback received during the last progress report, the display method now displays consistent symbols for when a cell is hit, missed, or has residence. Besides this I took the liberty to add in a new method that displays both the markers and the player's board on the same line in hope of saving terminal space. Also, I have updated the Cell class' field, instead of saving the residence's number, it now refers to the residence's (ship) instance itself, this makes giving feedback to the human player much easier, the actual values displayed for the cell remain unchanged.

## Display method

When the board display method is called, it presents the top line, loops through the rows and calls each of their display methods, then it displays the bottom line.

```
(defmethod display((b board) &aux width rows)
    (setf width (board-width b))
    (setf rows (board-rows b))
    (displayTop width)
    (loop for row in rows do
        (display row)
    )
    (newLine width)
)
```

When the row display method is called, it display the bufferline on top, loops through the cells in that row and calls each of their display methods, then display the '|' symbol to conclude the last cell block and terminate the line.

```
(defmethod display((r row) &aux cells rowLength rowNumber)
    (setf cells (row-cells r))
    (setf rowLength (length cells))

    ; +---+---+---+
    (newLine rowLength)

    ; Numbers on the left
    (setf rowNumber (row-number r))
    (if (< 9 rowNumber)
        (format t " ~A" rowNumber)
        (format t " ~A " rowNumber)
    )

    ; |   |   |
    (loop for cell in cells do
        (display cell)
    )
    (format t "| ~%")
)
```

When the cell display method is called, it checks if it needs to display any special symbols, ‘x’ for hit, ‘o’ for miss, ‘ ‘ for empty, if not it will call the ship class’s display method.

```
(defmethod display((c cell) &aux resident explored hit)
    (setf resident (cell-resident c))
    (setf explored (cell-explored c))
    (setf hit (isCellHit c))
    (cond
        ; Cell has been explored but no hit
        ((and explored (not hit))
            (format t "| o ")
        )
        ; Cell has been explored and there is a ship
        ((and explored hit)
            (format t "| x ")
        )
```

```
        ; Cell is not explored but there is a ship
        ((and (not explored) (not (equal resident nil)))
            (display resident)
        )
        ; Cell is both unexplored and have no ship
        (t
            (format t "|   ")
        )
    )
)
```

Lastly, the ship's display method shows the number representation of its type.

```
(defmethod display((s ship) &aux rep)
    (setf rep (get '*shipRep* (ship-type s)))
    (format t "| ~A " rep)
)

(setf (symbol-plist '*shipRep*) '(carrier 5 battleship 4 cruiser 3 submarine 2
destroyer 1))
```

An simple demo

```
[21]> (display Board)
     A   B   C   D   E   F   G   H   I   J
   +---+---+---+---+---+---+---+---+---+---+
 0 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
 1 |   |   | o |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
 2 | 5 | 5 | x | 5 | 5 |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
 3 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
 4 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
 5 |   |   |   |   |   | o |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
 6 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
 7 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
 8 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
 9 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
NIL
```

There also exists displayMarks methods that are 99% similar, it just does not display the residence.

## A method that display both on a same line

Board layer

```
(defmethod displayBoth((this board) (other board) &aux width thisRows
otherRows)
    (setf width (board-width this))
    (setf thisRows (board-rows this))
    (setf otherRows (board-rows other))

    (displayBothTop width)
    (dotimes (n (length thisRows))
        (displayBoth (nth n thisRows) (nth n otherRows))
    )
    (displayBothNewLine width)
)
```

Row layer

```
(defun displayBothTop(rowLength)
    ; Left board
    (format t "   ")
    (dotimes (n rowLength)
        (format t "  ~A " (cellToLetter n))
    )
    (format t " ")

    (middleBuffer)

    ; Right board
    (format t "   ")
    (dotimes (n rowLength)
        (format t "  ~A " (cellToLetter n))
    )
    (format t " ~%")
)

(defun displayBothNewLine(rowLength)
    ; Left board
```

```
    (format t "    ")
    (dotimes (n rowLength)
        (format t "+---")
    )
    (format t "+")

    (middleBuffer)

    ; Right board
    (format t "    ")
    (dotimes (n rowLength)
        (format t "+---")
    )
    (format t "+~%")
)

(defmethod displayBoth((this row) (other row) &aux thisCells otherCells
rowLength rowNumber)
    ; Both rows should be of equal length.
    (setf thisCells (row-cells this))
    (setf otherCells (row-cells other))
    (setf rowLength (length thisCells))
    (setf rowNumber (row-number this))

    ; Display a new line
    (displayBothNewLine rowLength)

    ; Display the left board (marks)
    (if (< 9 rowNumber)
        (format t " ~A" rowNumber)
        (format t " ~A " rowNumber)
    )
    (loop for cell in otherCells do
        (displayMarks cell)
    )
    (format t "|")

    (middleBuffer)
```

```
    ; Display the right board
    (if (< 9 rowNumber)
        (format t " ~A" rowNumber)
        (format t " ~A " rowNumber)
    )
    (loop for cell in thisCells do
        (display cell)
    )
    (format t "|~%")
)

(defun middleBuffer()
    ; Middle buffer space (4x space)
    (format t "    ")
)
```

The Cell layer uses the method previously presented.

A simple demo:
(Command [1] is to load everything)

```
[2]> (setf board (newBoard 10 10))
#<BOARD #x1A4092CD>
[3]> (setf otherBoard (newBoard 10 10))
#<BOARD #x1A4251E5>
[4]> (setf ship (newShip 'carrier))
#<SHIP #x1A4291C5>
[5]> (placeShip 1 0 5 0 ship board)
(#<CELL #x1A3E8775> #<CELL #x1A3E8791> #<CELL #x1A3E87AD> #<CELL #x1A3E87C9> #<CELL #x1A3E87E5>)
[6]> (fireAtBoard 5 5 otherBoard)
(5 5)
[7]> (displayBoth board otherBoard)
     A   B   C   D   E   F   G   H   I   J          A   B   C   D   E   F   G   H   I   J
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 0 |   |   |   |   |   |   |   |   |   |   |    0 |   | 5 | 5 | 5 | 5 | 5 |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 1 |   |   |   |   |   |   |   |   |   |   |    1 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 2 |   |   |   |   |   |   |   |   |   |   |    2 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 3 |   |   |   |   |   |   |   |   |   |   |    3 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 4 |   |   |   |   |   |   |   |   |   |   |    4 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 5 |   |   |   |   |   | o |   |   |   |   |    5 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 6 |   |   |   |   |   |   |   |   |   |   |    6 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 7 |   |   |   |   |   |   |   |   |   |   |    7 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 8 |   |   |   |   |   |   |   |   |   |   |    8 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 9 |   |   |   |   |   |   |   |   |   |   |    9 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
NIL
```