Kuncheng Feng
CSC 466
progress presentation #2

Changes from previous presentation:

- The game board now correctly calls its elements rows instead of columns.
- Each models' responsibility got shifted, the previous version's models have methods that should belong to higher hierarchies, leaving the board model with only 1 method and no demo at all.
- The order of cells in each row has been reversed.
- Board now displays in a more user friendly way.

Demo

```
[]> (demo--board)
Displaying a 5 x 5 board:
     1   2   3   4   5
   +---+---+---+---+---+
 A | 0 | 1 | 2 | 3 | 4 |
   +---+---+---+---+---+
 B | 0 | 1 | 2 | 3 | 4 |
   +---+---+---+---+---+
 C | 0 | 1 | 2 | 3 | 4 |
   +---+---+---+---+---+
 D | 0 | 1 | 2 | 3 | 4 |
   +---+---+---+---+---+
 E | 0 | 1 | 2 | 3 | 4 |
   +---+---+---+---+---+


Displaying a 10 x 10 board:
     1   2   3   4   5   6   7   8   9   10
   +---+---+---+---+---+---+---+---+---+---+
 A | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+
 B | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+
 C | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+
 D | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+
 E | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+
 F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+
 G | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+
```

```
 H | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+
 I | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+
 J | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
   +---+---+---+---+---+---+---+---+---+---+


Displaying a 15 x 15 board:
     1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 A | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 B | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 C | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 D | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 E | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 G | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 H | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 I | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 J | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 K | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 L | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 M | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 O | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14|
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
NIL
```

Codes

Main.l

```
; File: Main.l

(load "Board.l")
(load "Row.l")
(load "Cell.l")
```

Board.l

```
; File: Board.l
; Uses Row.l
(defclass board()
	(
		(rows :accessor board-rows :initarg :rows)
		(width :accessor board-width :initarg :width)
	)
)

; The board can support up to 26 rows, due to user experience reasons.
(defmethod newBoard(width height &aux rows)
	(setf rows (list))
	(dotimes (rowNum height)
		(setf rows (cons (newRow width rowNum) rows))
	)
	(setf rows (reverse rows))
	(make-instance 'board
		:rows rows
		:width width
	)
)

; All for user experience
(defmethod display((b board) &aux width rows)
	(setf width (board-width b))
	(setf rows (board-rows b))
	(displayTop width)
	(dotimes (n (length rows))
		(display (nth n rows))
	)
	(newLine width)
)

(defmethod columnToCell(columnNumber)
	(- columnNumber 1)
)

(defmethod cellToColumn(cellNumber)
	(+ cellNumber 1)
)


; Demo ----------------------------------------
```

```
(defmethod demo--Board()
	(format t "Displaying a 5 x 5 board: ~%")
	(display (newBoard 5 5))
	(format t "~%~%Displaying a 10 x 10 board: ~%")
	(display (newBoard 10 10))
	(format t "~%~%Displaying a 15 x 15 board: ~%")
	(display (newBoard 15 15))
)
```

Row.l

```
; File: Row.l
; Uses Cell.l
(defclass row()
	(
		(number :accessor row-number :initarg :number :initform 0)
		(cells :accessor row-cells :initarg :cells :initform nil)
	)
)

(defmethod newRow(width number &aux cells)
	(setf cells (list))
	(dotimes (cellNum width)
		(setf cells (cons (newCell number cellNum) cells))
	)
	(setf cells (reverse cells))
	(make-instance 'row
		:number number
		:cells cells
	)
)

; Display methods -----------------------
; Display the top line
(defmethod displayTop(boardWidth)
	(format t "   ")
	(dotimes (n boardWidth)
		(if (>= n 9)
			(format t "  ~A" (+ n 1))
			(format t "  ~A " (+ n 1))
		)
	)
	(format t " ~%")
)

; Repeating row displays -----------------
(setf letters '(a b c d e f g h i j k l m n o p q r s t u v w x y z))

(defmethod rowToLetter(rowNumber)
	(nth rowNumber letters)
)

(defmethod letterToRow(rowLetter)
	(position rowLetter letters)
)
```

```
(defmethod newLine(cellLength)
	(format t "   ")
	(dotimes (n cellLength)
		(format t "+---")
	)
	(format t "+ ~%")
)


(defmethod display((r row) &aux cells cellLength)
	(setf cells (row-cells r))
	(setf cellLength (length cells))
	; +---+---+---+
	(newLine cellLength)
	; A
	(format t " ~A " (rowToLetter (row-number r)))
	; |   |   |
	(dotimes (n cellLength)
		(display (nth n cells))
	)
	(format t "| ~%")
)
```

Cell.l

```
; File: Cell.l
(defclass cell()
	(
		(resident :accessor cell-resident :initarg :resident :initform
nil)
		(explored :accessor cell-explored :initarg :explored :initform
nil)
		(cellRow :accessor cell-row :initarg :cellRow :initform 0)
		(cellNum :accessor cell-num :initarg :cellNum :initform 0)
	)
)

(defmethod newCell(rowNumber cellNumber)
	(make-instance 'cell
		:resident nil
		:explored nil
		:cellRow rowNumber
		:cellNum cellNumber
	)
)

; This method display the player's board
(defmethod display((c cell) &aux resident number)
	(setf resident (cell-resident c))
	(setf number (cell-num c))
	(if (equal resident nil)
		(if (>= number 10)
			(format t "| ~A" number)
			(format t "| ~A " number)
```

```
            )
            (format t "| ~A " resident)
      )
)

; This method is for development purposes
(defmethod getInfo((c cell))
      (format t "Cell row: ~A~%" (cell-row c))
      (format t "Cell number: ~A~%" (cell-num c))
      (format t "Cell resident: ~A~%" (cell-resident c))
      (format t "Cell explored: ~A~%" (cell-explored c))
)
```