

CSC466 Project Task 1: Modeling Tiles

Abstract

In this task I created a model for representing a minesweeper tile in CLISP. I used the Common Lisp Object System to accomplish this task. I established a tile class which has fields for each adjacent tile along with fields for if it is a mine, if it is flagged, and how many mines are touching it.

Demo

```
[1]> ( load "ms.l" )  
;; Loading file ms.l ...  
;; Loaded file ms.l  
T  
[2]> ( demo--tile )  
>>> Testing tile class
```

Demo tile board

```
+-----+  
| > x # |  
| 1 4 # |  
| 0 1 > |  
+-----+
```

```
Tile: T1 nw: NIL n: NIL ne: NIL e: T2 se: T5 s: T4 sw: NIL w:  
NIL revealed: NIL mine: T flag: T value: 0  
Tile: T2 nw: NIL n: NIL ne: NIL e: T3 se: T6 s: T5 sw: T4 w: T1  
revealed: T mine: T flag: NIL value: 0  
Tile: T3 nw: NIL n: NIL ne: NIL e: NIL se: NIL s: T6 sw: T5 w:  
T2 revealed: NIL mine: T flag: NIL value: 0  
Tile: T4 nw: NIL n: T1 ne: T2 e: T5 se: T8 s: T7 sw: NIL w: NIL  
revealed: T mine: NIL flag: NIL value: 1  
Tile: T5 nw: T1 n: T2 ne: T3 e: T6 se: T9 s: T8 sw: T7 w: T4  
revealed: T mine: NIL flag: NIL value: 4
```

```
Tile: T6 nw: T2 n: T3 ne: NIL e: NIL se: NIL s: T9 sw: T8 w: T5
revealed: NIL mine: NIL flag: NIL value: 3
Tile: T7 nw: NIL n: T4 ne: T5 e: T8 se: NIL s: NIL sw: NIL w:
NIL revealed: T mine: NIL flag: NIL value: 0
Tile: T8 nw: T4 n: T5 ne: T6 e: T9 se: NIL s: NIL sw: NIL w: T7
revealed: T mine: NIL flag: NIL value: 1
Tile: T9 nw: T5 n: T6 ne: NIL e: NIL se: NIL s: NIL sw: NIL w:
T8 revealed: NIL mine: T flag: T value: 0
NIL
[3]>
```

Source Code

```
( defclass tile ()
  (
    ( name :accessor tile-name :initarg :name :initform nil )
    ( nw :accessor tile-nw :initarg :nw :initform nil )
    ( n :accessor tile-n :initarg :n :initform nil )
    ( ne :accessor tile-ne :initarg :ne :initform nil )
    ( e :accessor tile-e :initarg :e :initform nil )
    ( se :accessor tile-se :initarg :se :initform nil )
    ( s :accessor tile-s :initarg :s :initform nil )
    ( sw :accessor tile-sw :initarg :sw :initform nil )
    ( w :accessor tile-w :initarg :w :initform nil )
    ( revealed :accessor tile-revealed :initarg :revealed
:initform nil )
    ( mine :accessor tile-mine :initarg :mine :initform nil )
    ( flag :accessor tile-flag :initarg :flag :initform nil )
    ( value :accessor tile-value :initarg :value :initform 0 )
  )
)
```

```

( defmethod tile-info ( ( ti tile ) )
  ( format t "Tile: ~a nw: ~a n: ~a ne: ~a e: ~a se: ~a s: ~a
sw: ~a w: ~a revealed: ~a mine: ~a flag: ~a value: ~a~%"
    ( tile-name ti )
    ( tile-nw ti )
    ( tile-n ti )
    ( tile-ne ti )
    ( tile-e ti )
    ( tile-se ti )
    ( tile-s ti )
    ( tile-sw ti )
    ( tile-w ti )
    ( tile-revealed ti )
    ( tile-mine ti )
    ( tile-flag ti )
    ( tile-value ti )
  )
)

( defmethod demo--tile ()
  ( format t ">>> Testing tile class ~%~%" )
  ( format t " Demo tile board~%" )
  ( format t "      +-----+~%" )
  ( format t "      | > x # |~%" )
  ( format t "      | 1 4 # |~%" )
  ( format t "      | 0 1 > |~%" )
  ( format t "      +-----+~%~%" )
  ( let
    (
      t1 t2 t3 t4 t5 t6 t7 t8 t9
    )
    ( setf t1 ( make-instance 'tile :name 't1 :e 't2 :se 't5 :s
't4 :mine t :flag t ) )
    ( setf t2 ( make-instance 'tile :name 't2 :w 't1 :w 't1 :e
't3 :sw 't4 :s 't5 :se 't6 :mine t :revealed t ) )

```

```

    ( setf t3 ( make-instance 'tile :name 't3 :w 't2 :sw 't5 :s
't6 :mine t ) )
    ( setf t4 ( make-instance 'tile :name 't4 :n 't1 :ne 't2 :e
't5 :se 't8 :s 't7 :revealed t :value 1 ) )
    ( setf t5 ( make-instance 'tile :name 't5 :nw 't1 :n 't2 :ne
't3 :e 't6 :se 't9 :s 't8 :sw 't7 :w 't4 :revealed t :value 4 )
)
    ( setf t6 ( make-instance 'tile :name 't6 :nw 't2 :n 't3 :w
't5 :sw 't8 :s 't9 :value 3 ) )
    ( setf t7 ( make-instance 'tile :name 't7 :n 't4 :ne 't5 :e
't8 :revealed t ) )
    ( setf t8 ( make-instance 'tile :name 't8 :nw 't4 :n 't5 :ne
't6 :w 't7 :e 't9 :revealed t :value 1 ) )
    ( setf t9 ( make-instance 'tile :name 't9 :nw 't5 :n 't6 :w
't8 :mine t :flag t ) )

```

```

    ( tile-info t1 )
    ( tile-info t2 )
    ( tile-info t3 )
    ( tile-info t4 )
    ( tile-info t5 )
    ( tile-info t6 )
    ( tile-info t7 )
    ( tile-info t8 )
    ( tile-info t9 )
)
)

```