CSC466 Project Task 3

Revealing tiles and mine functionality

Abstract

In this task I implemented functions for revealing tiles and for revealing all of the mines when one is selected. To reveal tiles I created a function which would take a tile as an input. If the tile is a mine then a different function would be called which reveals all of the mines. If the tile was not a mine, it would be revealed. If the value of the tile is zero, the adjacent tiles would recursively be revealed until all of the tiles that can be revealed in the area are revealed.

Demo

```
[1]> ( load "ms.l" )
;; Loading file ms.l ...
;; Loaded file ms.l
T
[2]> ( demo--reveal-tiles )
>>> Testing the revealing of tiles
       Board configuration
       1 0 0 0 1 1 1 0 0 0
       0 0 0 0 0 0 0 0 0 0
       0 0 0 0 0 0 0 0 0 0
       0 0 0 0 0 0 0 0 0 0
       1 1 1 1 1 1 1 1 1 1
       0 0 0 0 0 1 0 0 0 0
       0 0 0 0 0 1 0 0 0 0
       0 0 0 0 1 0 0 0 0 0
       0 0 0 1 0 0 0 0 0 0
       0 0 1 0 0 0 0 0 0 0
```

```
   A B C D E F G H I J
  +--------------------+
0 | # # # # # # # # # # 
1 | # # # # # # # # # # 
2 | # # # # # # # # # # 
3 | # # # # # # # # # # 
4 | # # # # # # # # # # 
5 | # # # # # # # # # # 
6 | # # # # # # # # # # 
7 | # # # # # # # # # # 
8 | # # # # # # # # # # 
9 | # # # # # # # # # # 

Revealing tile at ( a 2 )
   A B C D E F G H I J
  +--------------------+
0 | # 1 0 1 # # # 1 0 0 
1 | 1 1 0 1 2 3 2 1 0 0 
2 | 0 0 0 0 0 0 0 0 0 0 
3 | 2 3 3 3 3 3 3 3 3 2 
4 | # # # # # # # # # # 
5 | # # # # # # # # # # 
6 | # # # # # # # # # # 
7 | # # # # # # # # # # 
8 | # # # # # # # # # # 
9 | # # # # # # # # # # 
```

```
Revealing tile at ( b 6 )
      A B C D E F G H I J
    +--------------------+
  0 | # 1 0 1 # # # 1 0 0
  1 | 1 1 0 1 2 3 2 1 0 0
  2 | 0 0 0 0 0 0 0 0 0 0
  3 | 2 3 3 3 3 3 3 3 3 2
  4 | # # # # # # # # # #
  5 | 2 3 3 3 # # # # # #
  6 | 0 0 0 1 # # # # # #
  7 | 0 0 1 2 # # # # # #
  8 | 0 1 2 # # # # # # #
  9 | 0 1 # # # # # # # #
Revealing tile at ( g 8 )
      A B C D E F G H I J
    +--------------------+
  0 | # 1 0 1 # # # 1 0 0
  1 | 1 1 0 1 2 3 2 1 0 0
  2 | 0 0 0 0 0 0 0 0 0 0
  3 | 2 3 3 3 3 3 3 3 3 2
  4 | # # # # # # # # # #
  5 | 2 3 3 3 # # 5 3 3 2
  6 | 0 0 0 1 # # 2 0 0 0
  7 | 0 0 1 2 # 2 1 0 0 0
  8 | 0 1 2 # 2 1 0 0 0 0
  9 | 0 1 # # 1 0 0 0 0 0
```

```
[3]> ( demo--reveal-mine )
>>> Testing revealing a mine
       Board configuration
       1 0 0 0 0 0 0 0 0 0
       0 1 0 0 0 0 0 0 0 0
       0 0 1 0 0 0 0 0 0 0
       0 0 0 1 0 0 0 0 0 0
       0 0 0 0 1 0 0 0 0 0
       0 0 0 0 0 1 0 0 0 0
       0 0 0 0 0 0 1 0 0 0
       0 0 0 0 0 0 0 1 0 0
       0 0 0 0 0 0 0 0 1 0
       0 0 0 0 0 0 0 0 0 1

       A B C D E F G H I J
     +----------------------+
   0 | # # # # # # # # # # #
   1 | # # # # # # # # # # #
   2 | # # # # # # # # # # #
   3 | # # # # # # # # # # #
   4 | # # # # # # # # # # #
   5 | # # # # # # # # # # #
   6 | # # # # # # # # # # #
   7 | # # # # # # # # # # #
   8 | # # # # # # # # # # #
   9 | # # # # # # # # # # #
```

```
Revealing ( i 3 )
       A B C D E F G H I J
     +--------------------+
   0 | # # 1 0 0 0 0 0 0 0
   1 | # # 2 1 0 0 0 0 0 0
   2 | # # # 2 1 0 0 0 0 0
   3 | # # # # 2 1 0 0 0 0
   4 | # # # # # 2 1 0 0 0
   5 | # # # # # # 2 1 0 0
   6 | # # # # # # # 2 1 0
   7 | # # # # # # # # 2 1
   8 | # # # # # # # # # #
   9 | # # # # # # # # # #
Revealing ( b 8 )
       A B C D E F G H I J
     +--------------------+
   0 | # # 1 0 0 0 0 0 0 0
   1 | # # 2 1 0 0 0 0 0 0
   2 | 1 2 # 2 1 0 0 0 0 0
   3 | 0 1 2 # 2 1 0 0 0 0
   4 | 0 0 1 2 # 2 1 0 0 0
   5 | 0 0 0 1 2 # 2 1 0 0
   6 | 0 0 0 0 1 2 # 2 1 0
   7 | 0 0 0 0 0 1 2 # 2 1
   8 | 0 0 0 0 0 0 1 2 # #
   9 | 0 0 0 0 0 0 0 1 # #
```

```
Revealing mine at ( e 4 )
       A B C D E F G H I J
     +--------------------+
   0 | X # 1 0 0 0 0 0 0 0
   1 | # X 2 1 0 0 0 0 0 0
   2 | 1 2 X 2 1 0 0 0 0 0
   3 | 0 1 2 X 2 1 0 0 0 0
   4 | 0 0 1 2 X 2 1 0 0 0
   5 | 0 0 0 1 2 X 2 1 0 0
   6 | 0 0 0 0 1 2 X 2 1 0
   7 | 0 0 0 0 0 1 2 X 2 1
   8 | 0 0 0 0 0 0 1 2 X #
   9 | 0 0 0 0 0 0 0 1 # X
NIL
```

Code

```
( defun reveal-tiles ( unexplored explored )
    ( if ( null unexplored ) ( return-from reveal-tiles ) )
    ( push ( pop unexplored ) explored )
    ( cond
        ( ( tile-mine ( car explored ) ) ( hit-mine ) )
        ( t
            ( setf ( tile-revealed ( car explored ) ) T )
            ( if ( = ( tile-value ( car explored ) ) 0 )
                ( reveal-tiles ( append unexplored ( adjacent-tiles ( car
explored ) explored unexplored ) ) explored )
                ( reveal-tiles unexplored explored ) )
        )
    )
)
( defmethod adjacent-tiles ( ( ti tile ) ( explored list ) ( unexplored list )
&aux l )
    ( setf l '() )
    ( if ( not ( or ( null ( tile-nw ti ) ) ( member ( tile-nw ti ) explored ) (
member ( tile-nw ti ) unexplored ) ) )  ( push ( tile-nw ti ) l ) )
    ( if ( not ( or ( null ( tile-n ti ) ) ( member ( tile-n ti ) explored ) (
member ( tile-n ti ) unexplored ) ) ) ( push ( tile-n ti ) l ) )
    ( if ( not ( or ( null ( tile-ne ti ) ) ( member ( tile-ne ti ) explored ) (
member ( tile-ne ti ) unexplored ) ) ) ( push ( tile-ne ti ) l ) )
    ( if ( not ( or ( null ( tile-e ti ) ) ( member ( tile-e ti ) explored ) (
member ( tile-e ti ) unexplored ) ) ) ( push ( tile-e ti ) l ) )
    ( if ( not ( or ( null ( tile-se ti ) ) ( member ( tile-se ti ) explored ) (
member ( tile-se ti ) unexplored ) ) ) ( push ( tile-se ti ) l ) )
    ( if ( not ( or ( null ( tile-s ti ) ) ( member ( tile-s ti ) explored ) (
member ( tile-s ti ) unexplored ) ) ) ( push ( tile-s ti ) l ) )
    ( if ( not ( or ( null ( tile-sw ti ) ) ( member ( tile-sw ti ) explored ) (
member ( tile-sw ti ) unexplored ) ) ) ( push ( tile-sw ti ) l ) )
    ( if ( not ( or ( null ( tile-w ti ) ) ( member ( tile-w ti ) explored ) (
member ( tile-w ti ) unexplored ) ) ) ( push ( tile-w ti ) l ) )
    l
)
( defun hit-mine ( &aux ti )
    ( dotimes ( n ( length ( board-tiles *board* ) ) )
        ( setf ti ( nth n ( board-tiles *board* ) ) )
        ( if ( and ( not ( tile-flag ti ) ) ( tile-mine ti ) )
            ( setf ( tile-revealed ti ) T )
        )
    ) T
)
```