CSC466: Artificial Intelligence 2
Tic Tac Toe: Visualization, Analysis, and Statistics
Ace DeSiena - Feb 2015

Code:

```
;;;Tic Tac Toe
;;;Ace DeSiena
;;;January-Febuary 2015
;;;CSC466

(defmethod select ((list list))
  (nth (random (length list)) list))

(defmethod snoc ((item symbol)( list list ))
  (if (null list)
      (list item)
    (cons (car list ) (snoc item (cdr list)))))

(defmethod visualize ((game list) &aux grid turnlist)
    (setf grid '(nw n ne w c e sw s se))
    (setf turnlist '(X1 O2 X3 O4 X5 O6 X7 O8 X9))
    (dotimes (x (length game))
      (setf grid (replacenth (position (nth x game) grid) grid (nth x turnlist))))
    (format t "~a ~a ~a ~%~a ~a ~a~%~a ~a ~a~%"
    (nth 0 grid)
    (nth 1 grid)
    (nth 2 grid)
    (nth 3 grid)
    (nth 4 grid)
    (nth 5 grid)
    (nth 6 grid)
    (nth 7 grid)
    (nth 8 grid)))

(defmethod replacenth ((index integer)(old list)(item symbol))
  (if (= index 0)
      (cons item (cdr old))
    (cons (car old) (replacenth (- index 1) (cdr old) item))))

(defmethod play (&aux play avail move)
  (setf play ())
  (setf avail '(nw n ne w c e sw s se))
  (dolist (player '(x o x o x o x o x))
    (cond
     ((eq player 'x)
      (setf move (select avail))
      (setf avail (remove move avail))
      (setf play (snoc move play)))
     ((eq player 'o)
      (setf move (select avail))
      (setf avail (remove move avail))
      (setf play (snoc move play)))))
  play)

(defmethod analyze ((game list) &aux outcome)
  (setf outcome ’d)
  (dotimes (x 7)
    (let ((temp (analyzePart game)))
      (cond
       ((eq temp ’w) (setf outcome ’w))
       ((eq temp ’L) (setf outcome ’l))))
    (setf game (butlast game)))
  outcome)


;;analyzepart takes a game of any length and determines if a win or loss for x has occured.
;;precondition: all previous subgames have been analyzed.
(defmethod analyzePart((game list)&aux xs os)
  (setf xs ())
  (setf os ())
  (dotimes (x (length game))
    (if (eq (mod x 2) 0)
(setf xs (snoc (nth x game) xs))
      (setf os (snoc (nth x game) os))))
  (cond
   ((win xs)
    ’w)
   ((win os)
    ’l)
   (t ())))

(defmethod win((moves list))
  (cond
   ;;horizontal
   ((and (member ’n moves) (member ’ne moves) (member ’nw moves)) t)
   ((and (member ’c moves) (member ’e moves) (member ’w moves)) t)
   ((and (member ’s moves) (member ’se moves) (member ’sw moves)) t)

   ;;vertical
   ((and (member ’sw moves) (member ’w moves) (member ’nw moves)) t)
   ((and (member ’c moves) (member ’s moves) (member ’n moves)) t)
   ((and (member ’e moves) (member ’se moves) (member ’ne moves)) t)

   ;;diagonal
   ((and (member ’se moves) (member ’c moves) (member ’nw moves)) t)
   ((and (member ’sw moves) (member ’c moves) (member ’ne moves)) t)
   (t ())))

;;this method simulates one play, visualizes it, and analyzes it
(defmethod demo-va (&aux p)
  (setf p (play))
  (format t "~a~%" p)
  (visualize p)
  (format t "~a~%" (analyze p))())

(defmethod stats((n number) (demo t) &aux w l d)
  (if demo (format t "Begin gathering statistics...~%"))
  (setf w 0 l 0 d 0)
  (dotimes (i n)
    (setf p (play))
    (if demo (format t "~A~%" p))
    (if demo (visualize p))
    (setf result (analyze p))
    (if demo (format t "~A~%" result))
    (cond
     ((eq result ’w) (setf w (+ w 1)))
     ((eq result ’l) (setf l (+ l 1)))
     ((eq result ’d) (setf d (+ d 1)))))
  (setf results (mapcar #’probability (list w l d) (list n n n)))
  (if demo (format t "End gathering statistics~%"))
  (mapcar #’list ’(w l d) results))

(defmethod probability ((special integer)(total integer))
  (/ (float special) (float total)))
```

Demo:

```
* (stats 5 t)
Begin gathering statistics...
(NW C E SE N S W SW NE)
X1 X5 X9
X7 O2 X3
O8 O6 O4
L
(NE SE E SW S W C N NW)
X9 O8 X1
O6 X7 X3
O4 X5 O2
D
(W NW SW C N NE SE S E)
O2 X5 O6
X1 O4 X9
X3 O8 X7
D
(SW NE C W E S N NW SE)
O8 X7 O2
O4 X3 X5
X1 O6 X9
D
```

```
(SE N C S SW E W NE NW)
X9 O2 O8
X7 X3 O6
X5 O4 X1
W
End gathering statistics
((W 0.2) (L 0.2) (D 0.6))
* (stats 1000000 ())

((W 0.585387) (L 0.287662) (D 0.126951))
*
```

Statistical Results

With an N of 1,000,000 the results of this test are fairly accurate. Given the conditions and strategies of the game, the X player has a 0.5853 chance of winning. The O player has a 0.2876 chance of winning. This is considerably less likely. There is also a distinct chance that there will be a draw with a likelihood of 0.1269.