Tic-tac-toe

J. Nelson Rushton
Texas Tech University

July 12, 2014

This is an LED program that defines a simple tic tac toe game to run in the Easel enviornment.

The game begins with an empty grid and ‘x to move. When an empty cell is clicked, the player whose turn it occupies that cell and it becomes the other player’s turn, until the game is over. When the game is over a message is displayed giving the result of the game. The player can press the *restart* button at any time to restart the game.

1 Tic tac toe data model

The types *player*, *cell*, *move*, and *state* are defined as follows:

- A *player* is the symbol ‘x or the symbol ‘o.
- A *cell* is an integer in $\{1, \ldots, 9\}$ .Cells represent squares on the tic tac toe board as pictured below:

1	2	3
4	5	6
7	8	9

- A *move* is a pair (c, p) where c is a cell and p is a player. The move (c, p) represents a move by player p in cell c
- A *state* is a set of moves, thought of as the set of moves made so far in the game. In this case, the state of the game is also the state of the program.

In this program, the variables p, c, and m will range over players, cells, and moves, respectively.

2 Game rules

This section defines the rules of tic-tac-toe in LED.

Since the game state is the set of moves made so far in the game, it begins as the empty set.

$$\mathit{startState} \;:=\; \emptyset$$

The *game board* is the set of all cells on the board:

$$\mathit{gameBoard} \;:=\; \{1, \ldots, 9\}$$

We say that player p *occupies* cell c if the pair (p, c) is a member of Γ, and that cell c is *occupied* if it is occupied by $`x$ or by $`o$.

$$\begin{aligned} \mathit{occupies}(p, c) \;&\equiv\; (p, c) \in \Gamma \\ \mathit{occupied}(c) \;&\equiv\; \mathit{occupies}(`x, c) \vee \mathit{occupies}(`o, c) \end{aligned}$$

A *row* is a set of cells that form three in a row either horizontally, vertically, or diagonally. Techincally, they should be called ‘lines’, but “three in a row” is an Anglo-American cultural idiom, while “three in a line” is not.

$$\mathit{rows} \;:=\; \mathit{hRows} \cup \mathit{vRows} \cup \mathit{dRows}$$

where

$$\begin{aligned} \mathit{hRows} \;&:=\; \{\{1,2,3\},\{4,5,6\},\{7,8,9\}\} \\ \mathit{vRows} \;&:=\; \{\{1,4,7\},\{2,5,8\},\{3,6,9\}\} \\ \mathit{dRows} \;&:=\; \{\{1,5,9\},\{3,5,7\}\} \end{aligned}$$

Player p has *three in a row* if he occupies all of the cells in some row.

$$\mathit{threeInRow}(p) \;\equiv\; \exists R \in \mathit{rows} : \forall c \in R : \mathit{occupies}(p, c)$$

We say that the *board is full* if every cell on the game board is occupied.

$$\mathit{boardFull} \;\equiv\; \forall c \in \mathit{gameBoard} : \mathit{occupied}(c)$$

The *game is over* if either the board is full, or one of the players has three in a row.

$$\mathit{gameOver} \;\equiv\; \mathit{boardFull} \vee \mathit{threeInRow}(`x) \vee \mathit{threeInRow}(`o)$$

The *player to move* is $`x$ if an even number of moves have been made, and $`o$ otherwise,

$$\mathit{playerToMove} \;:=\; \begin{cases} `x & \text{if } \mathit{even}(|\Gamma|), \\ `o & \text{otherwise} \end{cases}$$

where, as usual, an integer is *even* if it is divisible by 2.

$$\mathit{even}(n) \;\equiv\; n \bmod 2 = 0$$

The move (c, p) is *possible* in Γ if it is player p's turn, cell c is not occupied, and the game is not over.

$$possible(c, p) \;\equiv\; (playerToMove = p) \wedge \neg occupied(c) \wedge \neg gameOver$$

Finally, if move m is possible in the current state Γ, we write $result(m)$ for the state that results from making move m in Γ. Since the game state is just the set of moves that have been made, this is simple to define.

$$result(m) \;:=\; \Gamma \cup m$$