

TicTacToe

As stated in the specific the State of a cell is both a bound and constrained property. The click of a button (of a cell) raises a vetoable event to the controller that checks that the move is not a consecutive move from the same player that already moved.

If it is the case then the move is ignored, otherwise the State of the cell is changed and another event is raised to the board, which checks if the match has ended (either with a draw or with a winning player).

The rest of communication (reset of the board, a cell is part of a winning sequence, ...) is done with method invocation.

While this is far from the best possible solution since it makes highly coupled beans that are not easily usable in other contexts, it is the simpler and more readable one.

Doing all by event raising would be the best solution but it would require a rather intricate structure of event passing (every cell would raise an event whenever a button is clicked to notify the board to check for a winner, if a winner is found an event should be fired to specific cells to make the winning button green, ...).

A mixed version, where the communication is done by event with the exception of the complicated cases such as retrieving the state of a cell, where `getState()` would be used, does not decrease the coupling to a significant extent, hence we opted for a full method calling strategy.