

# JAVA – TicTacToe

[hobzamar@feld.cvut.cz](mailto:hobzamar@feld.cvut.cz) CVUT FEL: KYR – 2012/2013

2 Player game with standard TicTacToe rules

Gamefield takes the form of ButtonField. Handler is common to all the buttons in the field. It changes the ButtonText accordingly to whose turn it currently is, then changes the value of 2 dimensional int field to proper value. After that it checks the 2 dimensional int field and search for 5 same values in a row, if successful, it locks the ButtonField and open new window telling the players who was victorious.

Players then have choice of new game or exiting program.

Once the program starts, a window with 3 choices pops up (Offline game, online join, online host)

After picking one, another window pops up with 3 choices of field size

After picking one the actual game starts with X player starting.

Button that has been clicked doesn't react, thus preventing the players from overwriting opponent's move or switching to opponent's turn.

Once one of the player emerges victorious, the field is locked and new window pops up.

By choosing new game, new field appears and old one closes.

By choosing Exit, the program closes the same way as by hitting X button in top right corner of any window that pops up during the program run

Closer examination of the classes:

TicTacToe – Contains Main. Starts new instance of PreStart

PreStart – Extends JFrame: It's the first Window that pops up with 3 buttons

PreStartOper – Button handler for PreStart Window. Closes PreStart and starts new instance of StartWin, passing on the information of which button was pressed in form of number.

StartWin – Second window to pop up. It gives you the choice of field size(8x8,12x12,20x20)

StartOper – Handler for buttons in StartWin. Starts new instance of Window with the size parameter. Closes StartWin.

Window – Main window in the program. The actual game takes place here. It is filled with chosen amount of buttons that represent the places where you can play.

Oper – Handler for buttons in Window. If Button has no text, then it increases the counter by 1.

Depending on the counter it changes the button text to X or O and changes the value of 2 dimensional int field to 1 or 2 (the int field was initialized in class Window and has the same size as the button field) Then Goes through the int field in 4 directions looking for a row of 5 same numbers. If search is successful, the ButtonField locks down and new instance of EndWin (counter and field size are used as parameters) pops up.

EndWin – The last window in the game informs the players about the game result. Tells the players whether X or O is victorious depending on the counter value (the same way as Oper finds out if it's O or X turn)

Contains 2 buttons: New game and Exit

EndOper – Button handler for EndWin. New game will start new instance of Window. Exit will close the program.

All classes ending with \*Net are duplicate classes with modifications to run online

WindowNet - Depending on parameter, sets up new socket, or connects to socket. It uses two Booleans to determine whose turn it is and whether it's in state of waiting for response, or awaiting user's action. For determining victory it uses same algorithm as non \*Net class.