

Gameboard
entireGame: TicTacBoard*[9] # currentBoard: int # boardWinner: int #currentTurn: int
+GameBoard() : +displayGameBoard(boardNumbersDisplay: int) : void +getElementInOneSquare(square: TicTacBoard&, index: int) : char +setElementInOneSquare(boardIndex: int, squareIndex: int, newElement: char) : void +getOneTicTacBoard (index: int) : TicTacBoard +getCurrentBoardNumber () : int +setCurrentBoard (newBoard: int) : void +getBoardWinner():int +setBoardWinner (newWinner : int) : void +getTurn() : int +setTurn(newTurn : int) : void +checkWin() : bool +changePlayer(newPlayer:int) : void

TicTacBoard
-avaialableSquares: char[9] -winner: char
+TicTacBoard(): +getElement(index:int) : char +setElement(index:int,newElement:char) : void +displayBoard(currentCursorY:int,currentCursorX:int, displayAll:bool) : void +checkWin() : bool

Logic
-game:Gameboard -currentBoardNum:int
+logic() : +gameOpened(): void +playGame(game:GameBoard&) : void +ifBoardAlreadyWon(game : GameBoard&) : void +menuForPlayingGame(game:GameBoard&) : char +saveGame(game:GameBoard&) : void +loadGame(game:GameBoard&) : bool +menuGameOpened():int

