```
public class Board {
	//receives how many rows and columns are in the 2D array
	//creates a 2D array of chars - this will be the TicTacToe board
	protected int rows;
	protected int columns;
	protected char[][] board;

	public Board() {
		rows = 3;
		columns = 3;
		board = new char[rows][columns];
		int i = 1;
		for (int r = 0; r < rows; r++) {
			for (int c = 0; c < columns; c++) {
				board[r][c] = Character.forDigit(i, 10);
				i++;
			}
		}
	}

	public int getRows() {
		return rows;
	}

	public int getColumns() {
		return columns;
	}

	public char[][] getBoard() {
		return board;
	}

	public char getValue(int r, int c) {
		return board[r][c];
	}

	public boolean isValidMove(Move m) {
		return m.getRow() < 3 && m.getColumn() < 3 && m.getRow() > -1 &&
m.getColumn() > -1
				&& Character.isDigit(board[m.getRow()][m.getColumn()]);
	}
```

```
	public void resetSquare(int row, int col) {
		String ans = "";
		ans += 3 * row + col + 1;
		board[row][col] = ans.charAt(0);
	}

	public void makeMove(Move m) {
		board[m.getRow()][m.getColumn()] = m.getSigil();
	}

	public void setSquare(char marker, Move m) {
		board[m.getRow()][m.getColumn()] = marker;
	}

	public boolean hasWonHorizontal(char marker) {
		boolean hasWon = false;
		for (int r = 0; r < board.length; r++) {
			if (board[r][0] == marker && board[r][1] == marker && board[r][2] ==
marker) {
				hasWon = true;
			}
		}
		return hasWon;
	}

	public boolean hasWonVertical(char marker) {
		boolean hasWon = false;
		for (int c = 0; c < board[0].length; c++) {
			if (board[0][c] == marker && board[1][c] == marker && board[2][c] ==
marker) {
				hasWon = true;
			}
		}
		return hasWon;
	}

	public boolean hasWonDiag1(char marker) {
		return (board[0][0] == marker && board[1][1] == marker && board[2][2] ==
marker);
	}

	public boolean hasWonDiag2(char marker) {
```

```
		return (board[0][2] == marker && board[1][1] == marker && board[2][0] ==
marker);
	}

	public boolean hasWon(char marker) {
		return hasWonHorizontal(marker) || hasWonVertical(marker) ||
hasWonDiag1(marker) || hasWonDiag2(marker);
	}

	public boolean isDraw() {
		if (hasWon('X') || hasWon('O')) {
			return false;
		}
		for (int r = 0; r < board.length; r++) {
			for (int c = 0; c < board[r].length; c++) {
				if (Character.isDigit(board[r][c])) {
					return false;
				}
			}
		}
		return true;
	}

	public boolean gameOver() {
		return (isDraw() || hasWon('X') || hasWon('O'));
	}

	public String toString() {
		String str = "";
		for (int r = 0; r < rows; r++) {
			for (int c = 0; c < columns - 1; c++) {
				str += (board[r][c] + " | ");
			}
			str += (board[r][2] + "\n");
		}
		return str;
	}
}
```