CIS*1000: JavaScript Tic Tac Toe Guide

This guide will walk you through how to make a JavaScript calculator, starting from the template provided on the Assignment 4 page. This is not a guide to JavaScript, HTML, or CSS. It is assumed that you have completed the previous assignment and have some understanding of HTML and CSS. As for JavaScript, it is understood that you may have little or no programming knowledge, however, you are expected to familiarize yourself with JavaScript before starting this guide.

It is highly recommended that you look at the JavaScript tutorial from W3Schools (http://www.w3schools.com/js/). It covers a lot of basic and advanced concepts and provides interactive examples, so that you can you can experiment and see exactly how JavaScript works. In addition, there is also a JavaScript tutorial based off of the material from W3Schools with this course in mind (http://www.prezi.com/j7gnxedptxaw/introduction-to-javascript). When you're ready, download the template from the Assignment 5 page and open it up in both a web browser and your favourite plain text editor.

It's important that you regularly check to see what kind of effects your changes are having. If you break your calculator, then it will be easier to fix it sooner rather than later. Most web browsers have built-in web development tools that will help you determine why your JavaScript isn't working. Also here is a wiki for Tic Tac Toe rules (https://en.wikipedia.org/wiki/Tic-tac-toe)

Firefox

On Windows: Firefox menu -> Web Developer -> Web Console
On Mac: Tools menu -> Web Developer -> Web Console

Chrome

On both Windows and Mac: Menu -> Tools -> JavaScript Console

Safari

On Mac: Develop menu -> Show Error Console
Can't see the Develop menu? Go to the Safari menu, then Preferences…, and pick the Advanced tab. At the very bottom of this window should be the option Show Develop menu in menu bar, check this. You may need to restart Safari.

Internet Explorer

On Windows: Press F12 -> click the second symbol from the top on the left side of the little window
Finally, before the guide begins, please note that all of the code shown in this tutorial is colour coded. Code with a blue font should exist in your template already, or it should have been added in a previous step. Code with green font and a border is new.

Step 1: Getting Buttons to Work

First thing you should do is open the TicTacToe_Template.html in notepad, or text edit, or text wrangler, or notepad++, etc. And open TicTacToe_Template.html in your browser like Internet Explorer, Firefox, or Chrome. You should have 2 files open at once for testing and completing the assignment.

Starting from the template, you'll notice that none of the buttons seem to work. Our first goal will be to get these buttons to work when clicking on them. To start, let's look at the buttons themselves. These are buttons which begin at line 167 and go to line 181:

```
<td><input id="1" class="button" type="button" value="" onclick="placeMark(this.id, 1)"/></td>
<td><input id="2" class="button" type="button" value="" onclick="placeMark(this.id, 2)"/></td>
<td><input id="3" class="button" type="button" value="" onclick="placeMark(this.id, 3)"/></td>
<td><input id="4" class="button" type="button" value="" onclick="placeMark(this.id, 4)"/></td>
<td><input id="5" class="button" type="button" value="" onclick="placeMark(this.id, 5)"/></td>
<td><input id="6" class="button" type="button" value="" onclick="placeMark(this.id, 6)"/></td>
<td><input id="7" class="button" type="button" value="" onclick="placeMark(this.id, 7)"/></td>
<td><input id="8" class="button" type="button" value="" onclick="placeMark(this.id, 8)"/></td>
<td><input id="9" class="button" type="button" value="" onclick="placeMark(this.id, 9)"/></td>
```

You will see the new changes in THIS COLOUR that you should apply to your Tic_tac_toe file like above. We gave each button an ID which corresponds to each "slot" in a tic tac toe board, with 1 being the top left slot and 9 being the bottom right slot. Now you see placeMark() changed to placeMark(this.id, 1) will allow you to call the placeMark function that accepts the id of the button being clicked on and the position of the mark being used on the board. Once you complete the above step, you should save your file and move to next step. Your calculator will still not work properly until you complete a few more steps!

Go back to near the top of your code at line 66 you will see the below code, please apply the new changes below:

```
var playerOneSymbol = "X";

var playerTwoSymbol = "O";

var playerTurn = "One";

var gameWon = false;

var numOfMarks = 0;
```

From reading the above code and the new changes, what the variables do is exactly what you think they would from reading each one. For example the playerOneSymbol means player one has a symbol of “X”, playerTurn is set to the first player to go first and numOfMarks means there are currently 0 marks displayed on the tic tac toe board (a mark is either an “X” or “O”). Make sure to save your file, and now you are ready to start writing code in a function called placeMark to display a mark on the screen! You will see the below code at line 84.

```
function placeMark (id, position){
        if (gameWon == false){
                if (document.getElementById(id).value == ""){
                        /* Check which player is placing mark */
                        if (playerTurn == "One"){
                                document.getElementById(id).value = playerOneSymbol
                                /* Change Player Turn */
                                playerTurn = "Two"
                        }else {
                                document.getElementById(id).value = playerTwoSymbol
                                /* Change Player Turn */
                                playerTurn = "One"
                        }
                        /* Increment num of marks to keep track of how many are on the board */
                        numOfMarks++;
                        /* Change Board State */
                        checkBoard();
                        /* Update Player Turn Text */
                        document.getElementById('playerTurn').innerHTML = playerTurn;
                }
        }
}
```

There is a lot going on in this function, since this tutorial expects you to have some knowledge of basic JavaScript, I will only touch on each part of this function. First we check with an IF statement if the button chosen is BLANK (“”) we will input a new symbol into that button. Since we have a variable that keeps track which player is making the move, we will know which symbol to set to the button! Next we will keep track of the number of marks on the board and check if the placed mark is a winning spot with checkboard(). Since checkboard() function is empty, it does nothing for now. Once you complete all the above steps you should save your file and REFRESH your Tic Tac Toe game in your favourite internet browser and click a button to see new symbols working now! If nothing works, redo all of the above steps as you have an error somewhere in your code.

Step 2: Reset the Game

After testing the calculator a bit, I bet it’s getting annoying to not be able to RESET the game, so let’s do that next! We will begin at line 152:

```
function resetGame () {
        playerTurn = "One"
        document.getElementById('playerOneSymbolSpan').innerHTML = playerOneSymbol;
        document.getElementById('playerTwoSymbolSpan').innerHTML = playerTwoSymbol;
        document.getElementById('playerTurn').innerHTML = playerTurn;
        document.getElementById('playerWon').innerHTML = "";
        gameWon = false;
        numOfMarks = 0;
        document.getElementById('1').value = "";
        document.getElementById('2').value = "";
        document.getElementById('3').value = "";
        document.getElementById('4').value = "";
        document.getElementById('5').value = "";
        document.getElementById('6').value = "";
        document.getElementById('7').value = "";
        document.getElementById('8').value = "";
        document.getElementById('9').value = "";
    }
```

The above code is pretty easy to follow, it really just resets all our variables back to the same they were when you refresh your browser and it resets all your text / buttons displayed on your screen.

Step 3: Check marks on the board

Let's go back to the checkBoard() function as this is the function that allows us to logically solve the game depending on current marks on the board. Begin at line 135:

```
function checkBoard(){
        /* Check Top Row */
        checkRowOrColumn(1,3,'row');
        /* Check Middle Row */
        checkRowOrColumn(4,6,'row');
        /* Check Bottom Row */
        checkRowOrColumn(7,9,'row');
        /* Check Top Column */
        checkRowOrColumn(1,7,'column');
        /* Check Middle Column */
        checkRowOrColumn(2,8,'column');
        /* Check Bottom Column */
        checkRowOrColumn(3,9,'column');
        /* Check Diagonal One */
        checkRowOrColumn(1,9,'diagonalOne');
        /* Check Diagonal Two */
        checkRowOrColumn(3,7,'diagonalTwo');

        /* Reset Game if  board is full */
        if (numOfMarks >= 9 && gameWon == false){
                document.getElementById('playerWon').innerHTML = "TIE GAME -- Press RESET to play again!";
        }
}
```

You will notice I left out some of the green commented code that was in your file, you can keep it in there as it will assist with your understanding. Most of the explanation is done in the comments, but to further explain where you see the /* Check Top Row */ the code below that takes in the beginning mark to check which is 1 and the last mark which is 3 so it counts the top 1-3 marks which is the "top row" of a tic tac toe board. Also you will see a check for numofMarks >= 9 and this means we only show a TIE if the board is full! Now on line 113 add:

***** Do not test the game yet, as it will CRASH! We need to complete the next part first!! *****

```
function checkRowOrColumn(min, max, rowOrColumnOrDiagonal){
        var playerOneCount = 0;
        var playerTwoCount = 0;
        var valueChar = '';
        var number = 0;
        /* Check how we should loop through the slots of the board
        /* Very important to determine which logic to be used 'row', 'column', 'diagonalOne',
'diagonalTwo' --> would cover all possibilities of logic checks */
        if (rowOrColumnOrDiagonal == 'row'){
                number = 1;
        }else if (rowOrColumnOrDiagonal == 'column'){
                number = 3;
        }else if (rowOrColumnOrDiagonal == 'diagonalOne'){
                number = 4;
        }else if (rowOrColumnOrDiagonal == 'diagonalTwo'){
                number = 2;
        }

        /* Loop through the board to see which marks are displayed, make use of the number value to
determine how to do the logic check */
        for (i = min; i <= max; i = i + number) {
                valueChar = i.toString();
                if (document.getElementById(valueChar).value == playerOneSymbol){
                        playerOneCount++;
```

```
            }else if (document.getElementById(valueChar).value == playerTwoSymbol){
                playerTwoCount++;
            }
        }

        /* Show if someone won! It would count 3 marks for a player, and if someone got 3 it should show the winning player */
        if (playerOneCount == 3){
            document.getElementById('playerWon').innerHTML = "Player One WON!!!!";
            gameWon = true;
        }else if (playerTwoCount == 3){
            document.getElementById('playerWon').innerHTML = "Player Two WON!!!!";
            gameWon = true;
        }
}
```

The above code implements the logic to solve the Tic Tac Toe game with comments to support each part. Depending on which part the function is solving is based on the given input like 'row' which sets the number as the way to iterate over the game board marks from the for loop. The for loop keeps track of the amount of marks from each player and when there is 3 counted it will stop the game and tell the user that a player has won. Once you complete this code, you should save the code file and REFRESH your internet browser and test the Tic Tac Toe game!

Step 4: Set new symbols

On line 278 change the following code to:

```
<td><input id="setUpEverything" class="buttonSetup" type="button" value="Set Symbols" onclick="setSymbols()"/></td><br>
```

You will now see on line 210:

```
function func1(){

}
```

This function needs to be rewritten and changed so we can set new symbols for the player one or two while playing the game and support the new symbols. This function will also have "error checking" and will check if the input given was incorrect or not.

```
function setSymbols(){
	var error = false;
	playerOneSymbol = document.getElementById('playeOneSymbolText').value;
	playerTwoSymbol = document.getElementById('playeTwoSymbolText').value;

	/* Make sure to call the resetGame() function so the board is cleared for the new symbols! */
	resetGame ()

	/* You input checks after you reset ! */
	if (playerOneSymbol == "" || playerTwoSymbol == ""){
		document.getElementById('playerWon').innerHTML = "Input incorrect, using default symbols!";
		error = true;
	}else if (playerOneSymbol == playerTwoSymbol){
		document.getElementById('playerWon').innerHTML = "Both symbols are the same, using default symbols!";
		error = true;
	}else if (playerOneSymbol.length > 1){
		document.getElementById('playerWon').innerHTML = "Symbol length too long, using default symbols!";
		error = true;
	}

	if (error == true){
		playerOneSymbol = 'X';
		playerTwoSymbol = 'O';
		document.getElementById('playerOneSymbolSpan').innerHTML = playerOneSymbol;
		document.getElementById('playerTwoSymbolSpan').innerHTML = playerTwoSymbol;
	}
```

```
        //Reset Input
        document.getElementById('playeOneSymbolText').value = '';
        document.getElementById('playeTwoSymbolText').value = '';

}
```

The above code first resets the game because once you change a symbol the game should be reset. Next it checks if any of the inputs are empty, or longer than one character, or if two symbols are exact. If it finds any errors it will set the symbols to default and resume the game as if it never happened.

Once you complete the above code, save your code and refresh your internet browser of your Tic Tac Toe game and check if you can now set new symbols! If at any time your game stops working, you should go back to a previous step and retry the step and erase the code you did up to the point you know that you had working code.

Congratulations on completing the tutorial, hope you enjoy your new Tic Tac Toe JavaScript program!