CIS\*1000: JavaScript Connect Four 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](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 Connect Four rules (<https://en.wikipedia.org/wiki/Connect_Four>)

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 brown font is new code that you should add to your template.

Step 1: Getting Buttons to Work

First thing you should do is open the Connect\_Four\_Template.html in notepad, or text edit, or text wrangler, or notepad++, etc. And open Connect\_Four\_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 168 and go to line 221:

<td><input id="8" class="button" type="button" value="" onclick="placeDisc(this.id, 1)"/></td>

<td><input id="9" class="button" type="button" value="" onclick="placeDisc(this.id, 2)"/></td>

<td><input id="10" class="button" type="button" value="" onclick="placeDisc(this.id, 3)"/></td>

<td><input id="11" class="button" type="button" value="" onclick="placeDisc(this.id, 4)"/></td>

<td><input id="12" class="button" type="button" value="" onclick="placeDisc(this.id, 5)"/></td>

<td><input id="13" class="button" type="button" value="" onclick="placeDisc(this.id, 6)"/></td>

<td><input id="14" class="button" type="button" value="" onclick="placeDisc(this.id, 7)"/></td>

</tr>

<tr>

<td><input id="15" class="button" type="button" value="" onclick="placeDisc(this.id, 1)"/></td>

<td><input id="16" class="button" type="button" value="" onclick="placeDisc(this.id, 2)"/></td>

<td><input id="17" class="button" type="button" value="" onclick="placeDisc(this.id, 3)"/></td>

….Repeat pattern and fill in all ID up to 42 and placeDisc(this.id, \*) until you get to the last 3 as seen below

<td><input id="40" class="button" type="button" value="" onclick="placeDisc(this.id, 5)"/></td>

<td><input id="41" class="button" type="button" value="" onclick="placeDisc(this.id, 6)"/></td>

<td><input id="42" class="button" type="button" value="" onclick="placeDisc(this.id, 7)"/></td>

You will see the new changes in THIS COLOUR that you should apply to your Connect\_Four file like above. We gave each button an ID which corresponds to each “position” in a Connect Four board, with 1 being the top left slot and 42 being the bottom right position. Now you see placeDisc() changed to placeDisc(this.id, 1) will allow you to call the placeDisc function that accepts the id of the button being clicked on and the position of the disc being used on the board. Once you complete the above step, you should save your file and move to next step. Your Connect Four game will still not work properly until you complete a few more steps!

Go back to near the top of your code at line 76 you will see the below code, please apply the new changes below:

var gameWon = false;

var playerOneSymbol = "X";

var playerTwoSymbol = "O";

var playerTurn = "One";

var numOfDiscs = 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 numOfDiscs means there are currently 0 discs displayed on the Connect Four board (a disc 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 placeDisc to display a disc on the screen! You will see the below code at line 99.

function placeDisc (id, slot){

if (gameWon == false){

if (document.getElementById(id).value == ""){

/\* Check which player is placing disc \*/

if (playerTurn == "One"){

checkBelow(id, "X", slot);

/\* Change Player Turn \*/

playerTurn = "Two"

}else {

checkBelow(id, "O", slot);

/\* Change Player Turn \*/

playerTurn = "One"

}

/\* Increment num of discs to keep track of how many are on the board \*/

numOfDiscs++;

/\* Check Game 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! In this function we will also call checkBelow() function to see where to put the disc in that slot.

Next we will keep track of the number of discs on the board in the placeDisc function and check if the placed disc 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 Connect Four 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 Connect Four game a bit, we will need to RESET the game, so let’s do that next! We will begin at line 172:

function resetGame () {

/\* Put the game back to reset state \*/

playerTurn = "One"

document.getElementById('playerOneSymbolSpan').innerHTML = playerOneSymbol;

document.getElementById('playerTwoSymbolSpan').innerHTML = playerTwoSymbol;

document.getElementById('playerTurn').innerHTML = playerTurn;

document.getElementById('playerWon').innerHTML = "";

gameWon = false;

numOfDiscs = 0;

/\* Reset the original states of the button's - orange background colour and black text \*/

for (i = 1; i <= 42; i = i + 1) {

document.getElementById(i).value = "";

document.getElementById(i).style.backgroundColor = "orange";

document.getElementById(i).style.color = "black";

}

}

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 to their original state when you started the program.

Step 3: Check discs 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 discs on the board. Begin at line 149:

function checkBoard(){

/\* Check All Rows Here \*/

checkRowOrColumn(1, 7, 'row');

checkRowOrColumn(8, 14, 'row');

checkRowOrColumn(15, 21, 'row');

checkRowOrColumn(22, 28, 'row');

checkRowOrColumn(29, 35, 'row');

checkRowOrColumn(36, 42, 'row');

/\* Check All Columns \*/

checkRowOrColumn(1, 36, 'column');

checkRowOrColumn(2, 37, 'column');

checkRowOrColumn(3, 38, 'column');

checkRowOrColumn(4, 39, 'column');

checkRowOrColumn(5, 40, 'column');

checkRowOrColumn(6, 41, 'column');

checkRowOrColumn(7, 42, 'column');

/\* Check All for diagonals that go left to right (top to bottom) \*/

checkRowOrColumn(4, 28, 'diagonalOne');

checkRowOrColumn(3, 35, 'diagonalOne');

checkRowOrColumn(2, 42, 'diagonalOne');

checkRowOrColumn(1, 41, 'diagonalOne');

checkRowOrColumn(8, 40, 'diagonalOne');

checkRowOrColumn(15, 39, 'diagonalOne');

/\* Check All for diagonals that go right to left (top to bottom) \*/

checkRowOrColumn(4, 22, 'diagonalTwo');

checkRowOrColumn(5, 29, 'diagonalTwo');

checkRowOrColumn(6, 36, 'diagonalTwo');

checkRowOrColumn(7, 37, 'diagonalTwo');

checkRowOrColumn(14, 38, 'diagonalTwo');

checkRowOrColumn(21, 39, 'diagonalTwo');

/\* Reset Game if board is full \*/

if (numOfDiscs >= 42 && 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 Here \*/ the code below that takes in the beginning disc to check which is 1 and the last disc which is 7 so it counts the top 1-7 discs which is the “top row” of the Connect Four board. Also you will see a check for numofDiscs >= 42 and this means we only show a TIE if the board is full! Now on line 127 add:

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

function checkRowOrColumn(min, max, rowOrColumnOrDiagonal){

/\* Do not do anymore logic checks as we found the winning pieces \*/

if (gameWon != true){

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 = 7;

}else if (rowOrColumnOrDiagonal == 'diagonalOne'){

number = 8;

}else if (rowOrColumnOrDiagonal == 'diagonalTwo'){

number = 6;

}

/\* Loop through the board to see which discs are displayed, make use of the number value to determine how to do the logic check \*/

for (i = min; i <= max; i = i + number) {

if (playerOneCount != 4 && playerTwoCount != 4){

if (document.getElementById(i).value == playerOneSymbol){

playerOneCount++;

/\* Show the winning pieces by changing their font color on the board \*/

document.getElementById(i).style.color = "wheat";

/\* Reset colours of player two symbol \*/

for (n = min; n <= max; n = n + number) {

if (document.getElementById(n).value == playerTwoSymbol){

document.getElementById(n).style.color = "black";

}

}

playerTwoCount = 0;

}else if (document.getElementById(i).value == playerTwoSymbol){

playerTwoCount++;

/\* Show the winning pieces by changing their font color on the board \*/

document.getElementById(i).style.color = "wheat";

/\* Reset colours of player one symbol \*/

for (n = min; n <= max; n = n + number) {

if (document.getElementById(n).value == playerOneSymbol){

document.getElementById(n).style.color = "black";

}

}

playerOneCount = 0;

}else if (document.getElementById(i).value == ""){

playerOneCount = 0;

playerTwoCount = 0;

}

}

}

/\* Show if someone won! It would count 4 discs for a player, and show winning player \*/

if (playerOneCount == 4){

document.getElementById('playerWon').innerHTML = "Player One WON!!!!";

gameWon = true;

}else if (playerTwoCount == 4){

document.getElementById('playerWon').innerHTML = "Player Two WON!!!!";

gameWon = true;

}else{

/\* Reset all colours \*/

for (i = min; i <= max; i = i + number) {

/\* Show the winning pieces by changing their font color on the board row \*/

document.getElementById(i).style.color = "black";

}

}

}

}

The above code implements the logic to solve the Connect Four 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 discs from the for loop. The for loop keeps track of the amount of discs from each player and when there is 4 counted it will stop the game and tell the user that a player has won. Your game still won’t properly work till you finish the next step, but you should be able to click on a button and see the next player turn changing. When clicking on a button and you don’t see the player turn changing, you will have to restart the previous steps and delete your old code.

Step 4: Check discs below checked slot

You will now see on line 93:

/\* To be completed by you \*/

function checkBelow(){

}

Change the function to:

function checkBelow(id, value, slot){

var placedDisc = false;

var tempId = 35;

var position = 6;

/\* Find the last slot at the bottom and set value to that \*/

tempId = tempId + slot;

while (placedDisc == false){

if (document.getElementById(tempId).value == ""){

document.getElementById(tempId).value = value;

if (value == "X"){

document.getElementById(tempId).style.backgroundColor = "red";

}else{

document.getElementById(tempId).style.backgroundColor = "lightblue";

}

placedDisc = true;

}

/\* Start going up in columns in that slot \*/

tempId = tempId - 7;

}

}

This function will allow you to check the discs that are below your selected slot and try to place them at the lowest part of that slot at the bottom of the board.

Once you complete the above code, save your code and refresh your internet browser of your Connect Four game and check if your game finally works and plays properly! If the game is not 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 Connect Four JavaScript program!**