

# University of Central Florida

## CIS 4004 Web Based Information Technology

### Assignment 3 JavaScript Part 2 of 2

**Due, Thursday March 14, 2024 for 100% credit**

**Friday, March 15, 2024 for 90% credit**

**Saturday, March 16, 2024 for 80% credit**

**Sunday, March 17, 2024 for 70% credit**

#### **Deliverables**

1. To receive credit for the assignment upload to Webcourses as a compressed file (i.e. .zip, .rar, etc...) the following files:
  - a. index.html
  - b. connectfour.js
  - c. connectfour.css

#### **Update files from Assignment 3 Part 1**

1. index.html
2. connectfour.js
3. connectfour.css

#### **File provided**

1. connectfour\_template.js

#### **Project description**



This project will require students to generate a Connect Four board and replicate the board game based on game components, game setup, object of the game, game play, valid moves and end of game.

### Game components

The Connect Four game is a classic strategy game in which two players go head-to-head in a battle to own the grid!

- Players choose their disc colors.
- Empty board in a grid sized six row by seven columns.

### Object of the game

Players stack their colored discs upwards, horizontally, or diagonally to get four in a row to win.

### Game play

- “Yellow” goes first.
- Players take turns dropping the discs into the grid, starting in the middle or at the edge to stack their colored discs upwards, horizontally, or diagonally.
- Use strategy to block opponents while aiming to be the first player to get four in a row to win.

### End of game

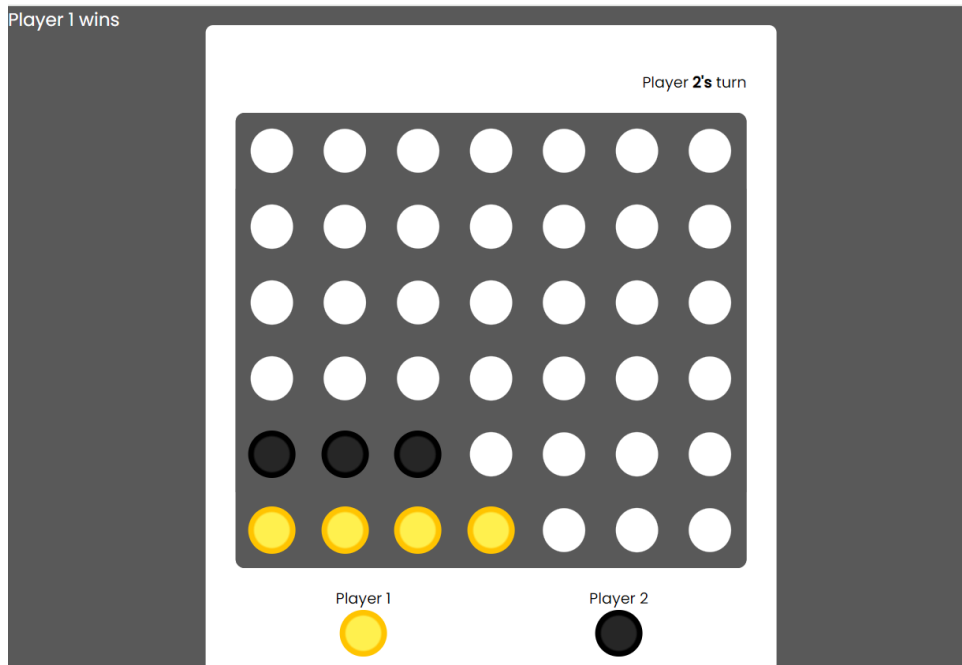
One player gets four discs in a row upwards, horizontally, or diagonally.

Tasks	
<b>function gameOverCheck</b>	<ol style="list-style-type: none"> <li>Update function <b>gameOverCheck</b> to do the following               <ol style="list-style-type: none"> <li>Declare variable <b>count</b>, initialized to <b>0</b></li> <li>Write a <b>for/of</b> loop to iterate through the rows, loop control variable <b>innerArray</b>, in 2d array <b>initialMatrix</b> <ol style="list-style-type: none"> <li>If object <b>innerArray</b>, function <b>every(val =&gt; (val) != 0)</b> <ol style="list-style-type: none"> <li>increment variable <b>count</b> by 1</li> </ol> </li> <li>Else                   <ol style="list-style-type: none"> <li>return <b>false</b></li> </ol> </li> </ol> </li> <li>If variable count is equal to 6               <ol style="list-style-type: none"> <li>Set constant <b>message</b>, property <b>innerText</b>, equal to "<b>Game Over</b>"</li> <li>return <b>false</b></li> </ol> </li> </ol> </li> </ol>
<b>function winCheck</b>	<ol style="list-style-type: none"> <li>Update function <b>winCheck</b> to do the following               <ol style="list-style-type: none"> <li>Write decision making logic, if function call <b>checkHorizontal</b>, <b>checkVertical</b>, <b>checkPositiveDiagonal</b>, or <b>checkNegativeDiagonal</b> is true                   <ol style="list-style-type: none"> <li>return <b>true</b></li> </ol> </li> <li>else                   <ol style="list-style-type: none"> <li>return <b>false</b></li> </ol> </li> </ol> </li> </ol>
<b>function setPiece</b>	<ol style="list-style-type: none"> <li>Update function <b>setPiece</b> to do the following               <ol style="list-style-type: none"> <li>Current <b>if/else</b> should be wrapped inside a <b>try{}</b> block</li> <li>The <b>catch(e){}</b> block should display and <b>alert</b> dialog</li> </ol> </li> </ol>

	box informing the player " <b>Column full, select again</b> "
<b>function checkHorizontal</b>	4. Write function <b>checkHorizontal</b> to do the following <ul style="list-style-type: none"> <li>a. Write a nest for loop to iterate through the rows and columns <ul style="list-style-type: none"> <li>1. if the <b>currentPlayer</b> has four discs in a row horizontally <ul style="list-style-type: none"> <li>a. return <b>true</b></li> </ul> </li> </ul> </li> <li>b. return <b>false</b></li> </ul>
<b>function checkVertical</b>	5. Write function <b>checkVertical</b> to do the following <ul style="list-style-type: none"> <li>a. Write a nest for loop to iterate through the columns and rows <ul style="list-style-type: none"> <li>1. if the <b>currentPlayer</b> has four discs in a row vertically <ul style="list-style-type: none"> <li>a. return <b>true</b></li> </ul> </li> </ul> </li> <li>b. return <b>false</b></li> </ul>
<b>function checkPositiveDiagonal</b>	6. Write function <b>checkPositiveDiagonal</b> to do the following <ul style="list-style-type: none"> <li>a. Write a nest for loop to iterate through the rows and columns <ul style="list-style-type: none"> <li>1. if the <b>currentPlayer</b> has four discs in a row diagonally, bottom right to top left <ul style="list-style-type: none"> <li>a. return <b>true</b></li> </ul> </li> </ul> </li> <li>b. return <b>false</b></li> </ul>
<b>function checkNegativeDiagonal</b>	7. Write function <b>checkNegativeDiagonal</b> to do the following <ul style="list-style-type: none"> <li>a. Write a nest for loop to iterate through the rows and columns <ul style="list-style-type: none"> <li>1. if the <b>currentPlayer</b> has four discs in a row diagonally, bottom left to top right <ul style="list-style-type: none"> <li>a. return <b>true</b></li> </ul> </li> </ul> </li> <li>b. return <b>false</b></li> </ul>
<b>connectfour.js</b>	8. The <b>window.onload = startGame</b> should be the last statement in the .js source code

Test Cases		
	Action	Expected outcome
Test Case 1	A player has four discs in a row horizontally	The web browser updates the Connect Four game to state the players wins, similar to Figure 1
Test Case 2	A player has four discs in a row vertically	The web browser updates the Connect Four game to state the players wins, similar to Figure 2
Test Case 3	A player has four discs in a row diagonally, bottom left to top right	The web browser updates the Connect Four game to state the players wins, similar to Figure 3
Test Case 4	A player has four discs in a row diagonally, bottom right to top left	The web browser updates the Connect Four game to state the players wins, similar to Figure 4
Test Case 5	The board is full with no	The web browser updates the Connect Four

	<b>winner, game is over</b>	game to state the game is over, similar to Figure 5
<b>Test Case 6</b>	<b>A column is full and a player clicks on the column</b>	The web browser updates to display an alert dialog box, similar to Figure 6
<b>Test Case 7</b>	<b>Web browser console</b>	The web browser console should have no errors, Figure 7



**Figure 1 Horizontal win**

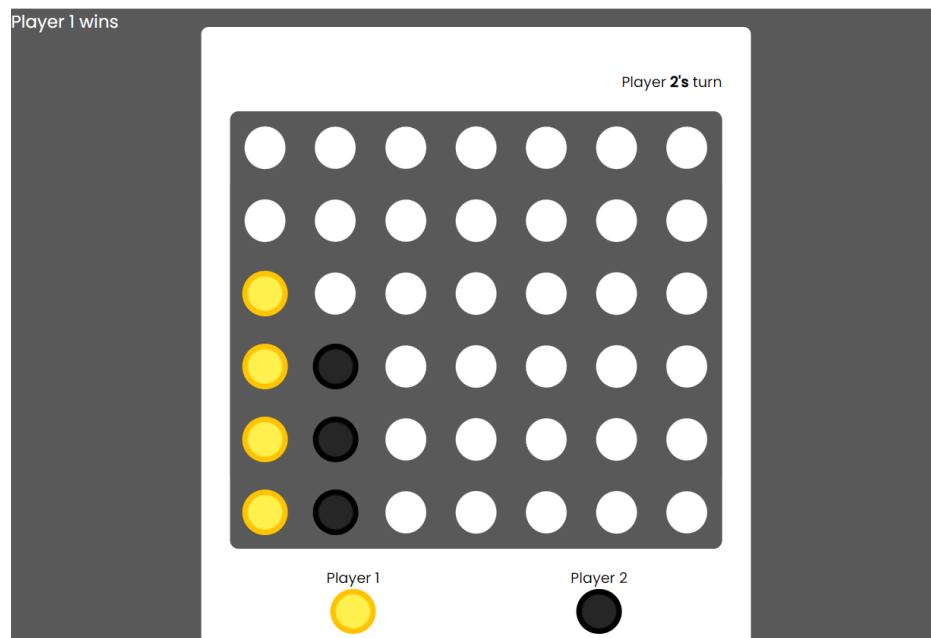


Figure 2 Vertical win

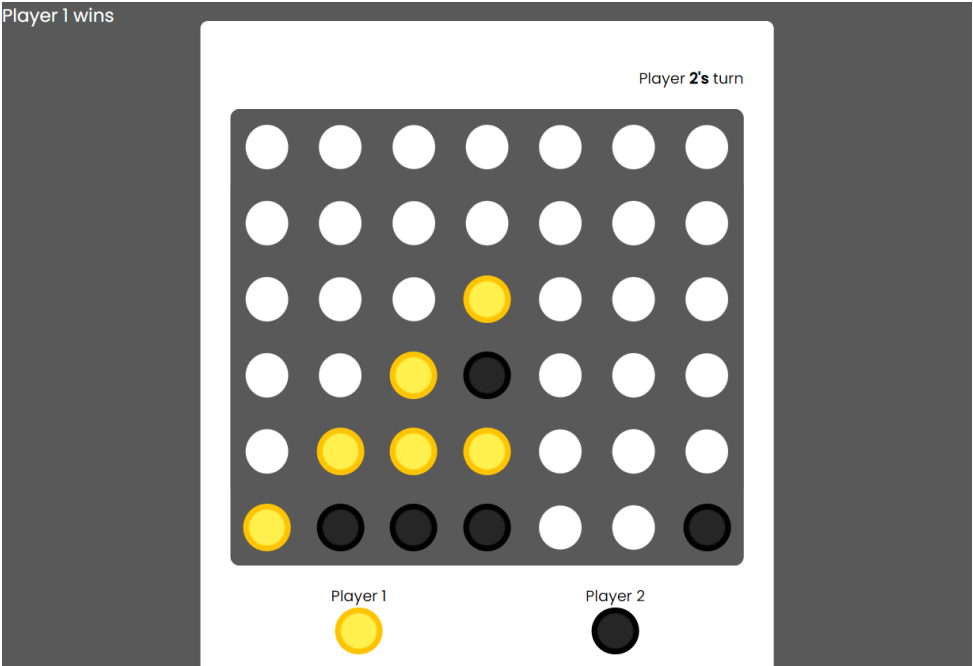


Figure 3 Diagonal win

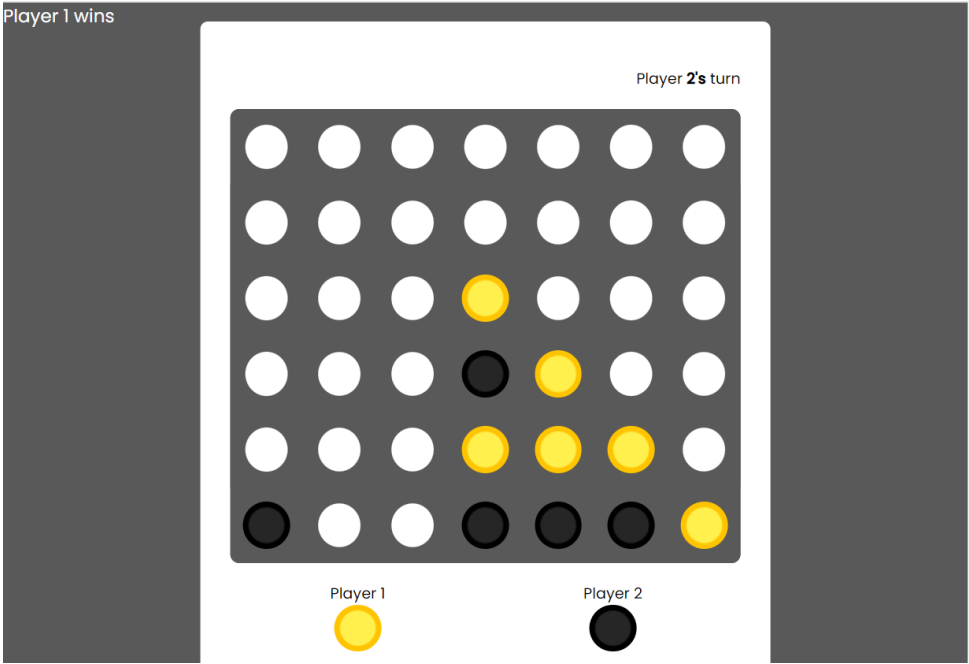
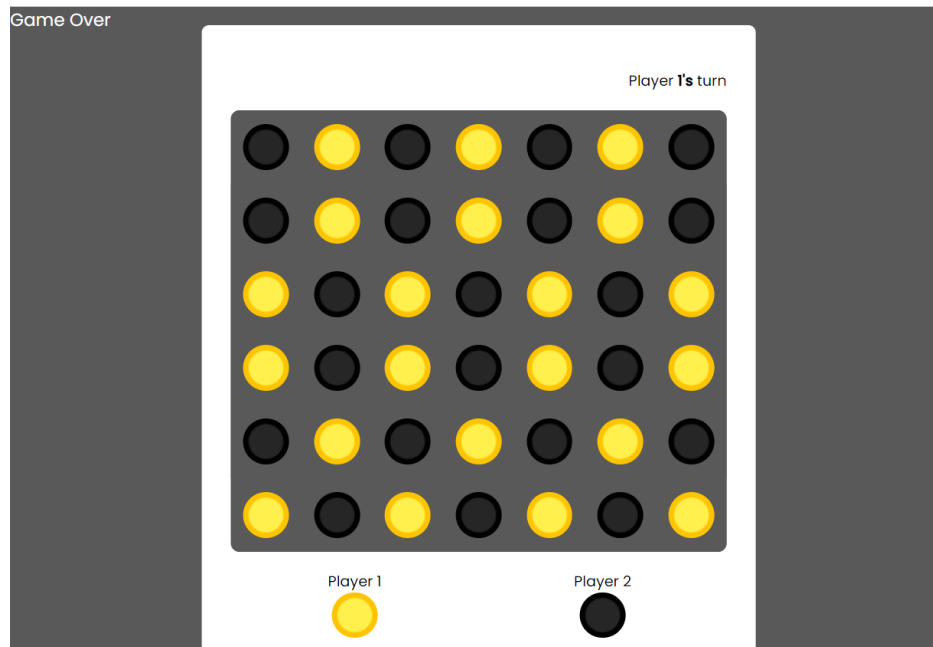
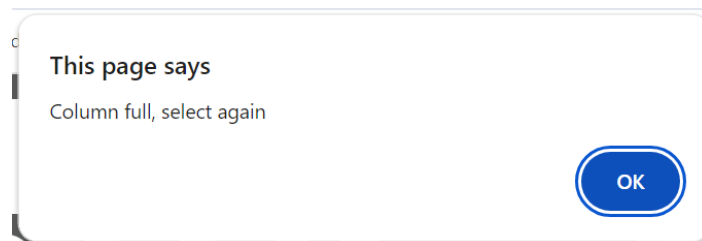


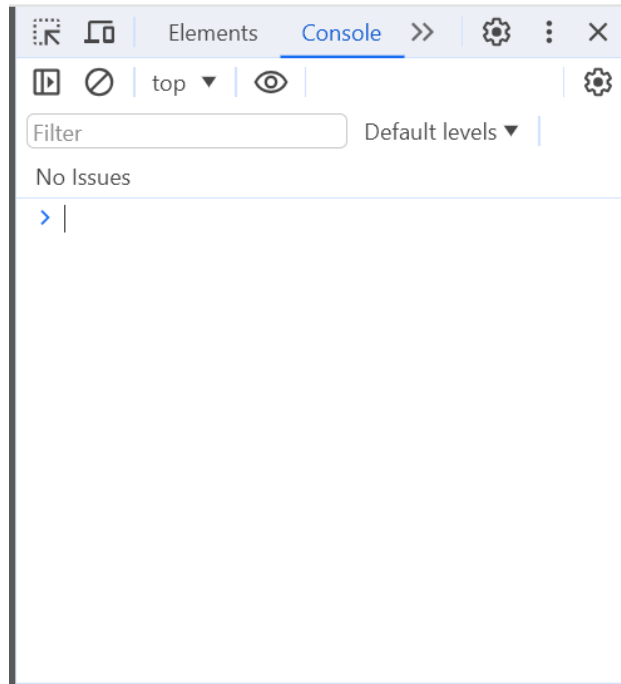
Figure 4 Diagonal win



**Figure 5 Game over, no winner**



**Figure 6 Column full**



**Figure 7 Web browser console**