University of Central Florida

CIS 4004 Web Based Information Technology

Assignment 3 JavaScript
Part 1 of 2

Due, Wednesday March 6, 2024 for 100% credit
Thursday, March 7, 2024 for 90% credit
Friday, March 8, 2024 for 80% credit
Saturday, March 9, 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

Files provided

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

Project description

MB
Milton Bradley
Connect Four
THE VERTICAL CHECKERS GAME
Ages 7 and Up
2 Players
Take turns dropping checkers into slots. First to get four in a row wins!
After the game a flick of the lever releases the checkers.
Color and design of parts may vary.

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	
connectfour.js	1. Rename **connectfour_template.js** source code file **connectfour.js**
	2. Declare constant **container** set equal to object **document**, method **querySelector**, passing as an argument class **".container"**
	3. Declare constant **playerTurn** set equal to object **document**, method **getElementById**, passing as an argument id **"playerTurn"**
	4. Declare constant **message** set equal to object **document**, method **getElementById**, passing as an argument id **"message"**
	5. Declare variable **initialMatrix** as a 2d array, 6 rows, 7 columns, initialized to all **0**s
	6. Declare variable **currentPlayer** to store the current player
function gameOverCheck	7. Write function **gameOverCheck** to do the following a. Empty parameter list b. Return **false**
function winCheck	8. Write function **winCheck** to do the following a. Parameter list i. **row** ii. **column** b. Return **false**
function setPiece	9. Write function **setPiece** to do the following a. Parameter list i. **startCount**

	ii. **colValue** b. Declare variable **rows** initialized to object **document**, method **querySelectorAll**, passing argument class **".grid-row"** c. **If** the element in array **initialMatrix** at indexes parameters **startCount** and **colValue** is NOT equal to **0** i. Decrement parameter **startCount** by **1** ii. Call function **setPiece**, passing as arguments parameters **startCount** and **colValue** d. **Else** i. Declare variable **currentRow** initialized to array **rows**, index **startCount**, method **querySelectorAll**, passing as an argument class **".grid-box"** ii. Modify **currentRow**, index **colValue**, object **classlist**, method **add**, passing as arguments **"filled"** and **`player${currentPlayer}`** iii. Update array **initialMatrix**, indexes **startCount** and **colValue**, set equal to **currentPlayer** iv. **If** function call **winCheck**, passing as arguments parameters **startCount** and **colValue,** is true 1. Set object **message**'s **innerHTML** equal to **`Player<span> ${currentPlayer}</span> wins`** 2. Return **false** e. Call function **gameOverCheck**
function fillBox	10. Write function **fillBox** to do the following a. Parameter list, **e** b. Declare variable **colValue** set equal to function **parseInt**() of parameter **e**, object **target**, function **getAttribute**, passing as argument **"data-value"** c. Call function **setPiece**, passing arguments **5** (because we have 6 rows, 0 - 5) and variable **colValue** d. Switch the **currentPlayer**, if currently 1 then 2, if currently 2, then 1 e. Set **playerTurn**'s **innerHTML** to **`Player <span>${currentPlayer}'s</span> turn`**
function createBoard	11. Write function **createBoard** to iterate through the 2d array **initialMatrix** and do the following a. Empty parameter list b. Write an outer **for in** loop to iterate through the rows, loop control variable **innerArray** i. Declare variable **outerDiv** set equal to object

	document, method **createElement**, passing **"div"** as an argument ii. Modify **outerDiv**, **classList**, to **add** class **"grid-row"** iii. Modify **outerDiv** to **setAttribute "data-value"** to loop control variable **innerArray** iv. Write an inner **for in** loop to iterate through the columns, loop control variable **j** 1. Set each element in array **initialMatrix** to the value of **0** 2. Declare variable **innerDiv** set equal to object **document**, method **createElement**, passing **"div"** as an argument 3. Modify **innerDiv**, **classList**, to **add** class **"grid-box"** 4. Modify **innerDiv** to **setAttribute "data-value"** to loop control variable **j** 5. Modify **innerDiv** to **addEventListener**, passing arguments **"click"** and **(e) => { fillBox(e); }** 6. Modify **outerDiv** to **appendChild**, passing argument **innerDiv** v. Modify **container** to **appendChild**, passing argument **outerDiv**
function startGame	12. Write function **startGame** to do the following a. Empty parameter list b. Set **currentPlayer** to 1, player 1 always goes first c. Set the **container**'s **innerHTML** to an empty string d. Call function **createBoard** e. Set **playerTurn**'s **innerHTML** to **`Player <span>${currentPlayer}'s</span> turn`**
connectfour.js	13. For the **window.onload** event, call function **startGame**
connectfour.css	14. Modify the following colors to any color of your choice a. body, background-color b. .container, background-color c. .player1:before, background d. .player2:before, background

Test Cases		
	Action	**Expected outcome**
Test Case 1	**Lauch index.html in web browser**	When index.html loads, the web browser should look similar to Figure 1
Test Case 2	**Player 1 clicks column**	The web browser updates the Connect Four

		game which should look similar to Figure 2
Test Case 3	**Player 2 clicks column**	The web browser updates the Connect Four game which should look similar to Figure 3
Test Case 4	**Refresh the web browser**	When index.html loads, the web browser should look similar to Figure 1
Test Case 5	**Web browser console**	The web browser console should have no errors, Figure 4

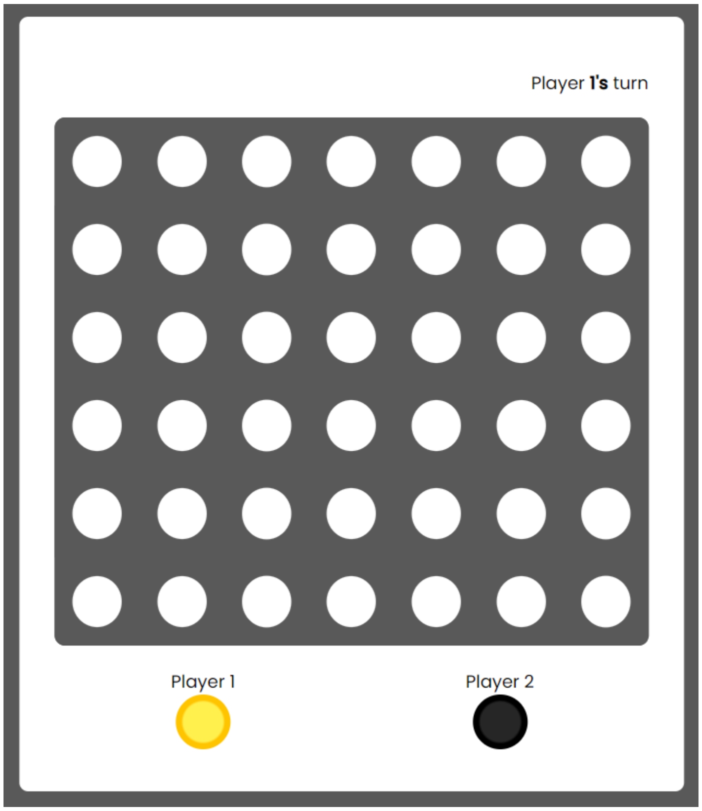

Figure 1 Connect Four

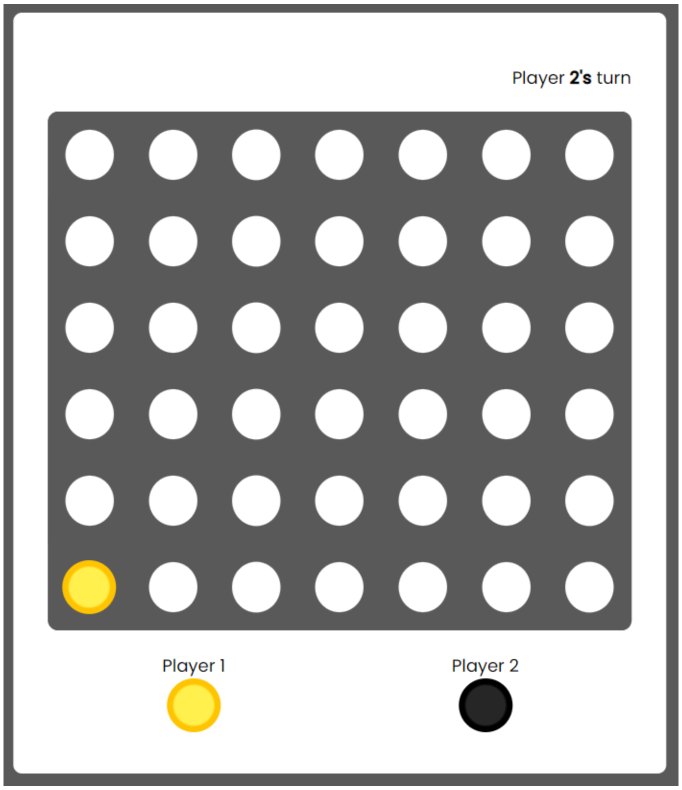
Player **2's** turn

Figure 2 Player 1 turn

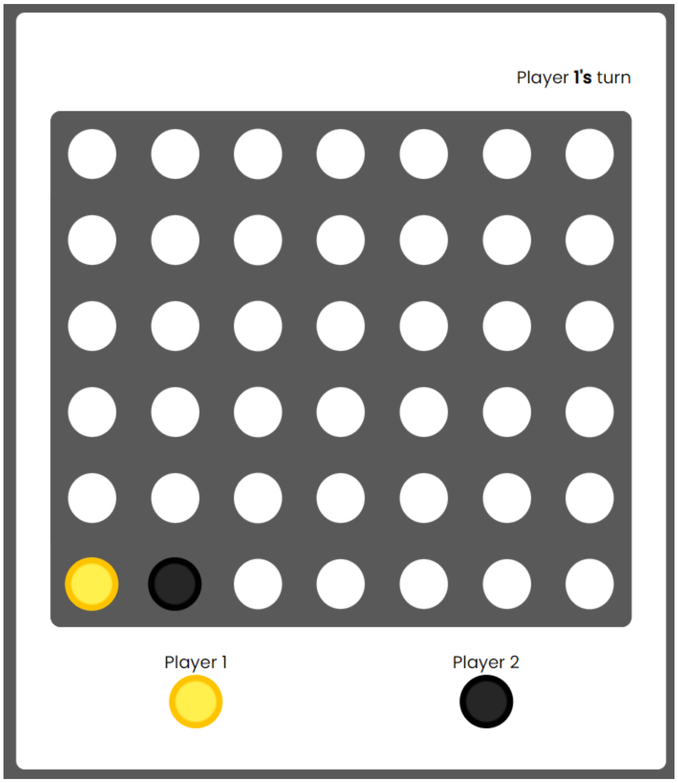
Player **1's** turn

Figure 3 Player 2 turn

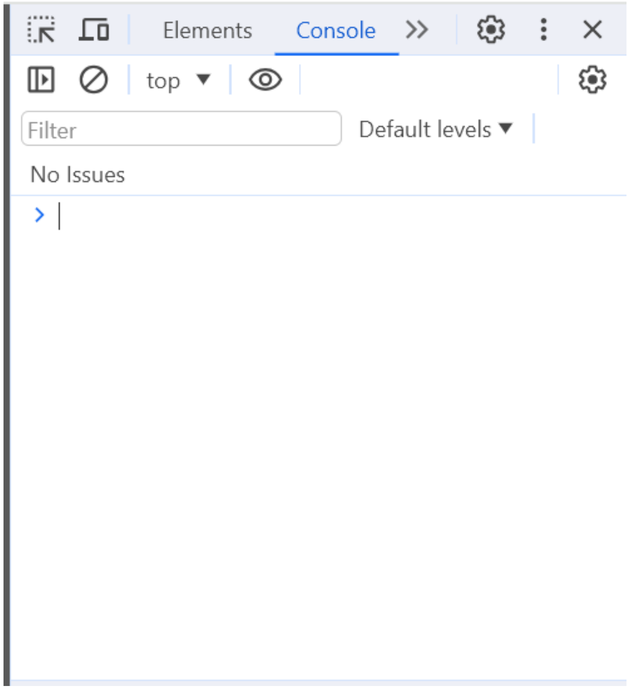

Figure 4 Web browser console