University of Central Florida

CIS 4004 Web Based Information Technology

Assignment 2 JavaScript

Due, Wednesday July 5, 2023 for 100% credit
Thursday, July 6, 2023 for 90% credit
Friday, July 7 2023 for 80% credit
Saturday, July 8, 2023 for 70% credit

Assignment scope
This assignment is focused on writing JavaScript and will create an electronic version of a Checkers game. This assignment will require students to generate a Checkers board.

Deliverables
To complete this assignment you must execute the following tasks

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. script.js
    c. style.css
    d. function.js
    e. img folder with
        i. board.png
        ii. checkers.gif

Files provided
1. index_template.html
2. script_template.js
3. style.css
4. function.js
5. img folder with
    a. board.png
    b. checkers.gif

Tasks

Activity

index.html

<head>

1. Set <title> to Checkers
2. Use the <link> element to reference the external cascading style sheet style.css

<body>

1. Create div with class container
    a. Create div with classes next-player and counter
        i. Create div with classes occupied and whitePiece; id next-player
    b. Create div with class game; id game
    c. Create div with class counter
        i. Create div
            1. Create div with classes occupied and whitePiece
            2. Create span with id white-player-count-pieces, initial value 10
        ii. Create div
            1. Create div with classes occupied and blackPiece
            2. Create span with id black-player-count-pieces, initial value 10
2. Create div with class modal; id easyModal
    a. Create div with class modal-content
        i. Create div with class modal-body
            1. Create paragraph; The player won the game !!
            2. Create paragraph; , would you like to take your revenge ???
            3. Create div with class btn-container
                a. Create button with class btn; onclick calls function location.reload(); text Yes
                b. Create button with class btn; onclick calls function modalClose(); text No
3. Use <script> to include external JavaScript files function.js and script.js

script.js

1. Create class **Piece**
    a. Create method **constructor**, parameter list includes **row** and **column**
        i. Add attribute **row** to this class, set equal to the **row** parameter
        ii. Add attribute **column** to this class, set equal to the **column** parameter

b. Create method **compare**, parameter list includes **piece**
    i. Return the result of condition
        1. **row** attribute of **piece** object is equal to **this** class's **row** attribute

        AND

        2. **column** attribute of **piece** object is equal to **this** class's **column** attribute

2. Declare a constant **modal**, set equal to **document**, **getElementById**, passing **"easyModal"** as an argument
3. Declare a variable **game**, set equal to **document**, **getElementById**, passing **"game"**
4. Declare a variable **currentPlayer**, set equal to the value **1**
5. Declare an empty array **posNewPosition**
6. Declare an empty array **capturedPosition**
7. Declare a 2-d array **board**, set equal to data set
    a. row 0: 0, -1, 0, -1, 0, -1, 0, -1, 0, -1
    b. row 1: -1, 0, -1, 0, -1, 0, -1, 0, -1, 0
    c. row 2: 0, -1, 0, -1, 0, -1, 0, -1, 0, -1
    d. row 3: -1, 0, -1, 0, -1, 0, -1, 0, -1, 0
    e. row 4: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    f. row 5: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    g. row 6: 0, 1, 0, 1, 0, 1, 0, 1, 0, 1
    h. row 7: 1, 0, 1, 0, 1, 0, 1, 0, 1, 0
    i. row 8: 0, 1, 0, 1, 0, 1, 0, 1, 0, 1
    j. row 9: 1, 0, 1, 0, 1, 0, 1, 0, 1, 0
8. Call function **buildBoard**

Test Cases

Test Case 1	Rendered HTML page looks similar to Figure 1
Test Case 2	White checkers can only make valid moves, Figure 2
Test Case 3	Black checkers can only make valid moves, Figure 3
Test Case 4	Player scores update after checkers are captured, Figure 4

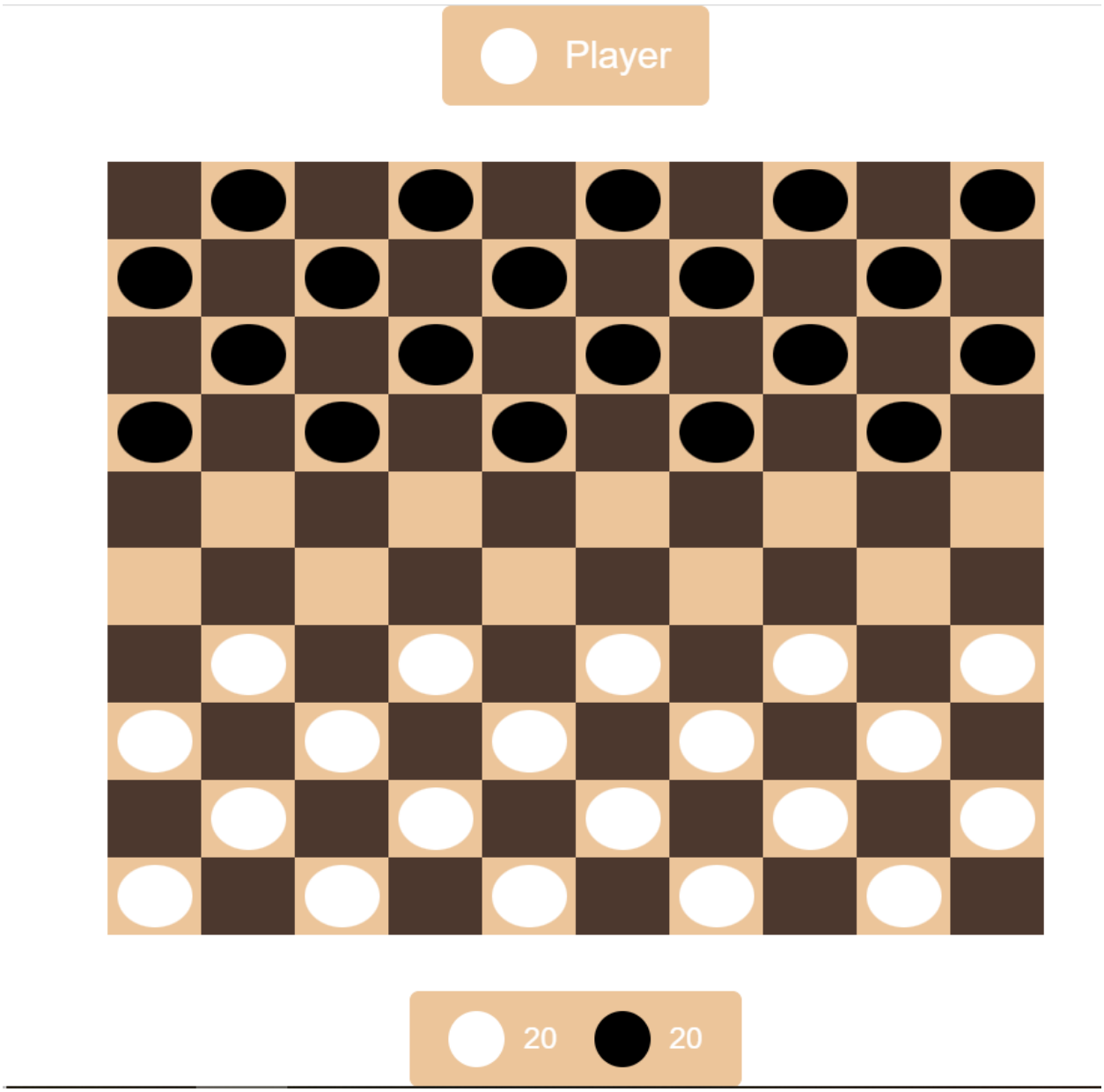

Figure 1 Checkers

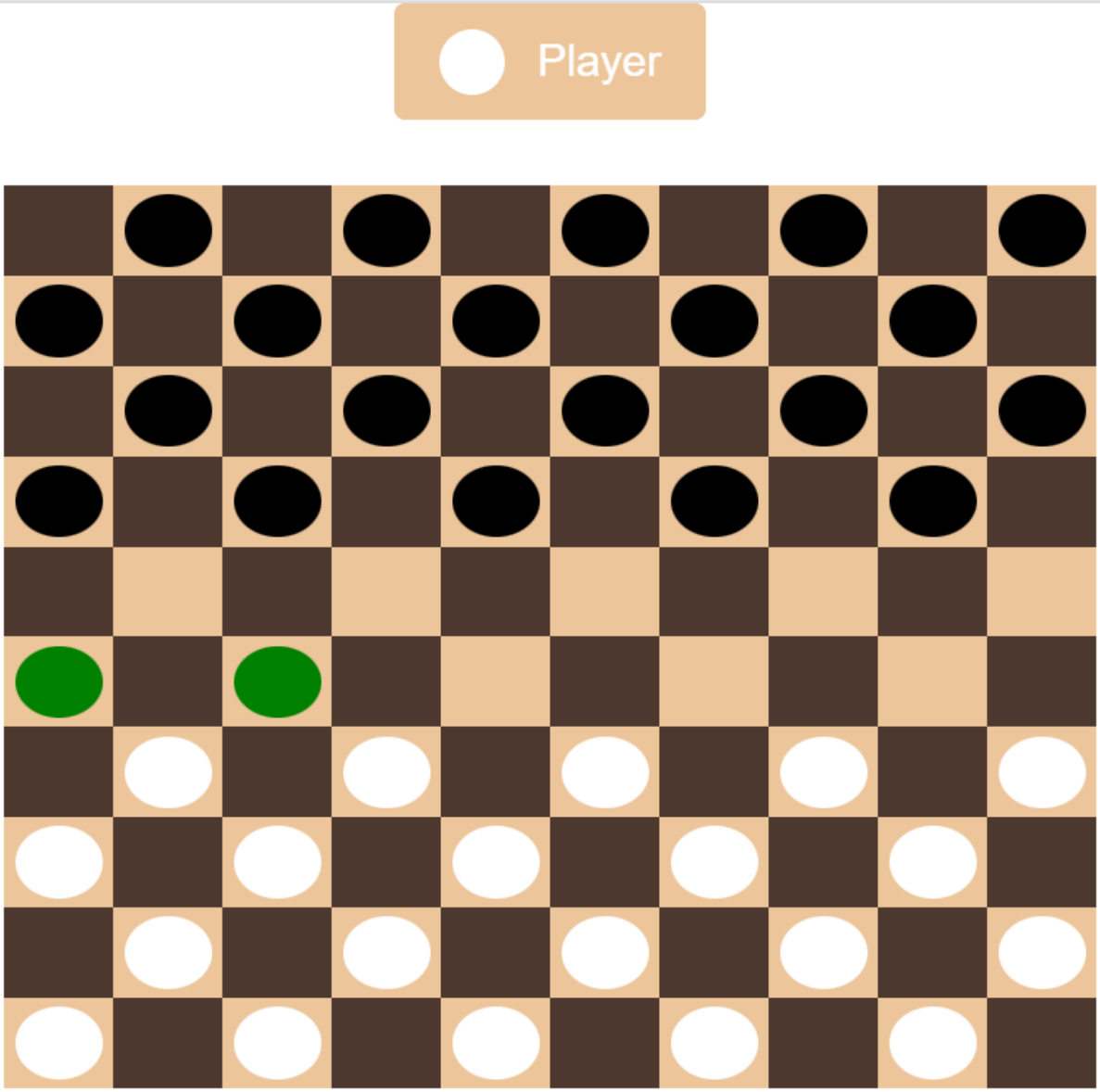

Figure 2 White checker turn

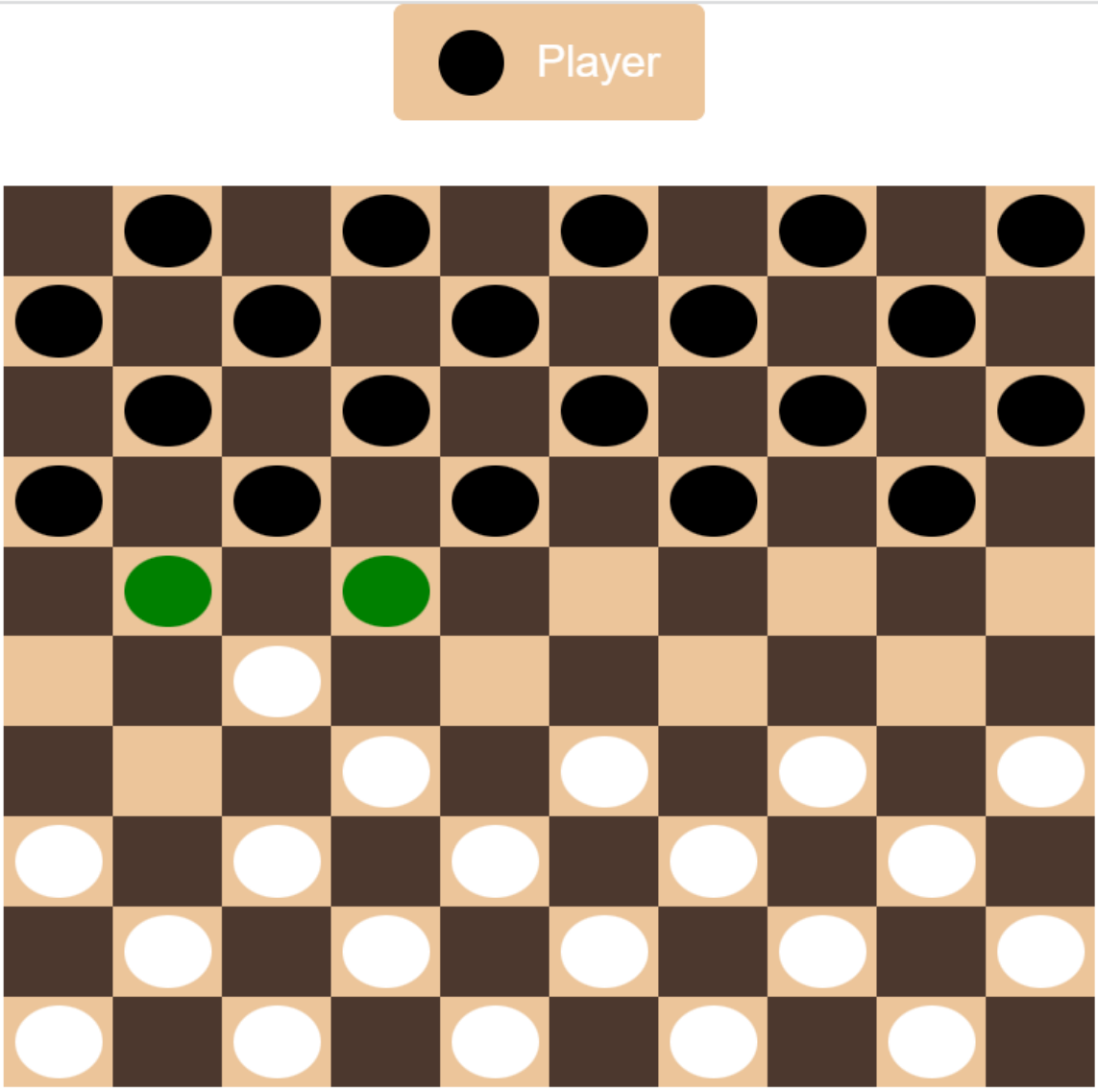

Figure 3 Black checker turn

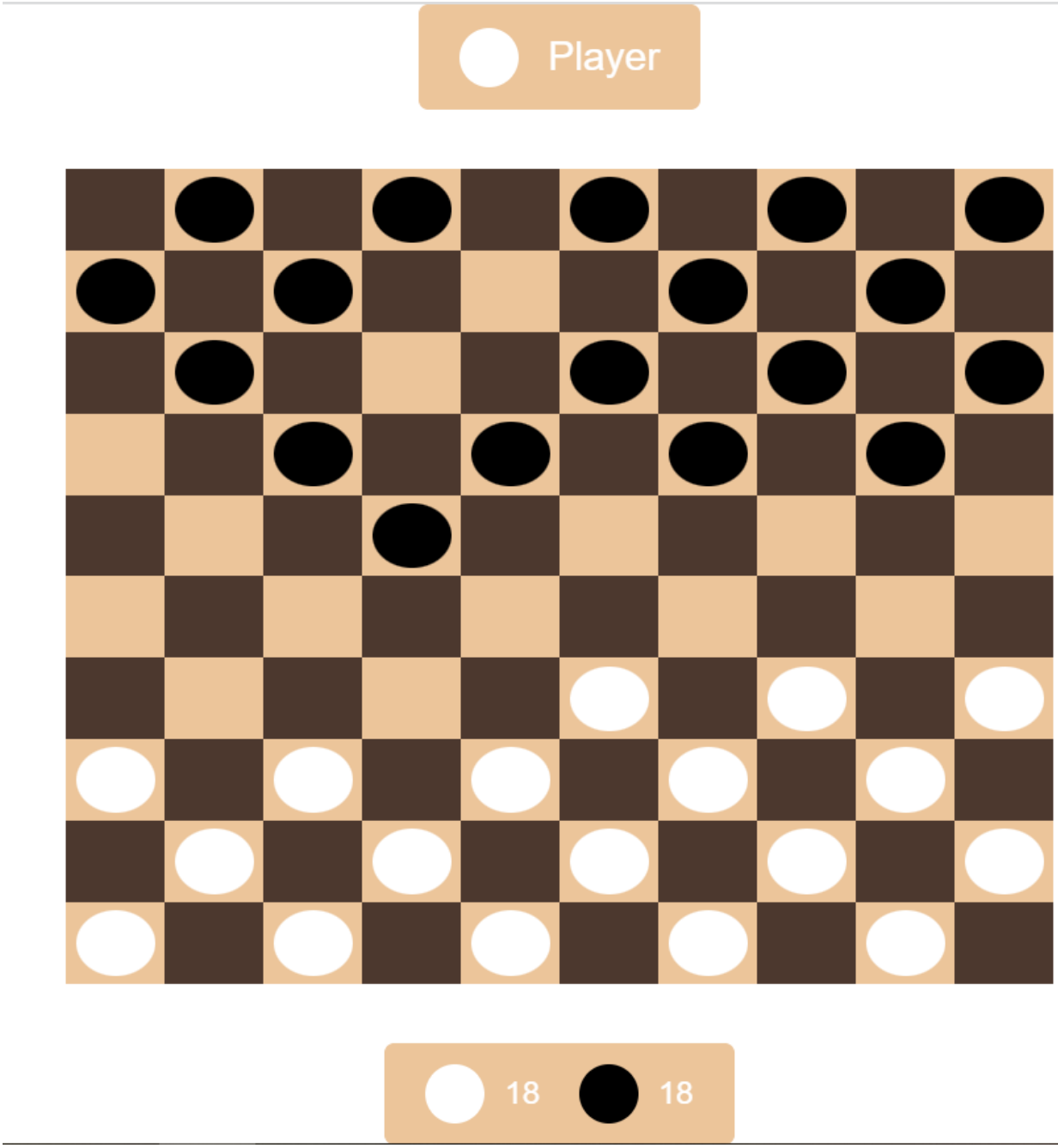

Figure 4 Scores update