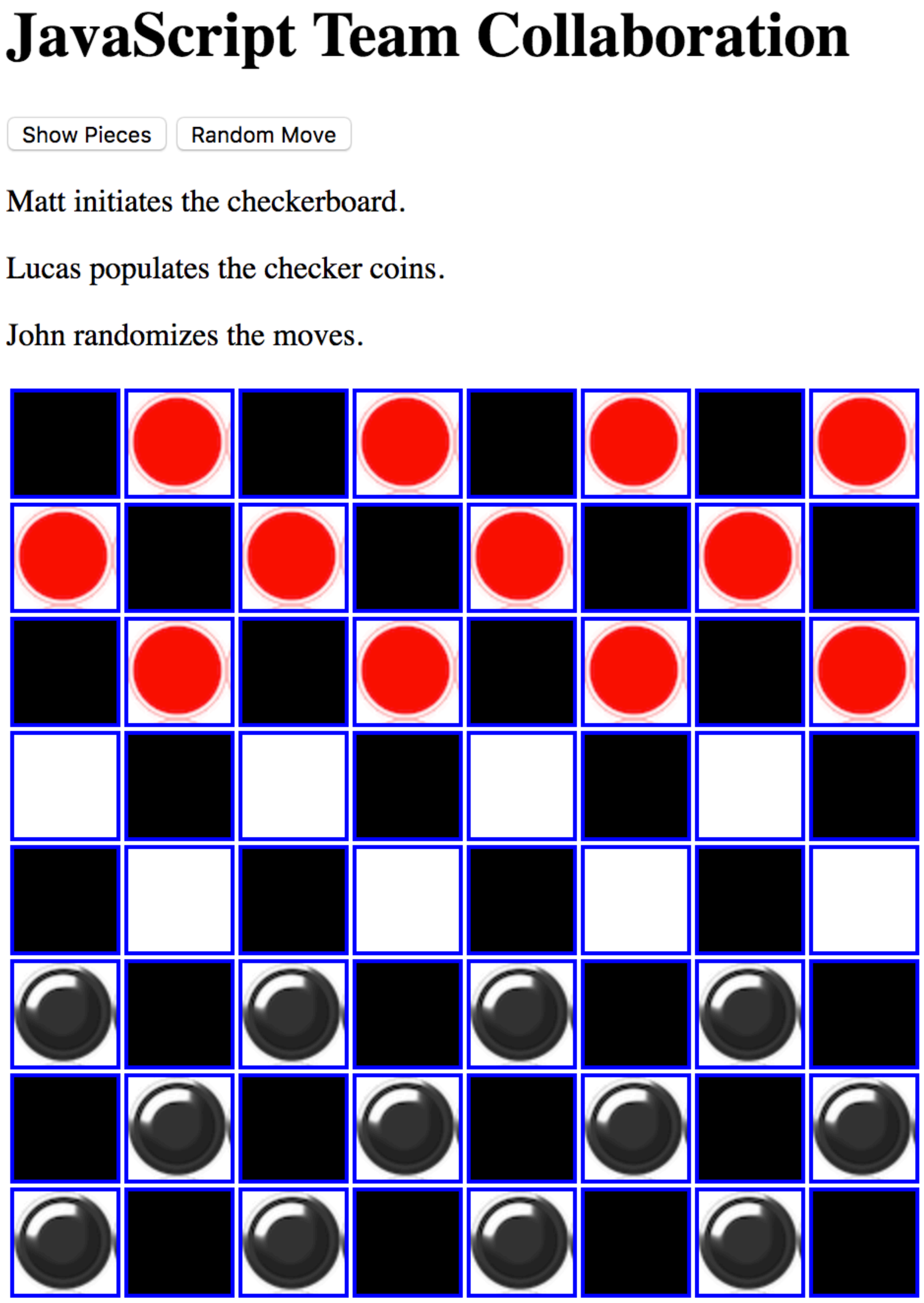
INFO 151: Team Project Assignment:

HTML and JavaScript Collaboration

Assignment specification

For this team assignment you are required to create a web page with a checkerboard as shown in the following figure:



JavaScript Team Collaboration

Matt initiates the checkerboard.

Lucas populates the checker coins.

John randomizes the moves.

The final output will be the 8 x 8 checkerboard showing the pieces correctly located. There will be two event driven buttons: (a) ***Show Pieces*** (to place the pieces on the initial blank checkerboard), and (b) ***Random Move*** (to move a single piece using a random move).

IMPORTANT: this is not a fully functioning checkers game which requires complex rules created using artificial intelligence techniques. Your objective is to create the checkerboard with correctly located pieces and random moves.

The objectives of the team project include:

- Creating of a web page using HTML5 with JavaScript scripting
- Using of iteration (loops), arrays, and functions in JavaScript
- Applying methods and properties in the DOM for dynamic content manipulation
- Develop team collaboration experience for system and application development

There will be three members of the team (in the figure: “Matt”, “Lucas”, and “John” – in your project you will use your own names).

Step 1: Select the team lead and project preparation:

- Appoint one member of the team as team leader (member #1). The team leader will lay the foundation for the assignment (the project);
- All the team (members #2 and #3) will assist the team leader (member #1) in implementing steps 1 to 4;
- Step #5 and step #6 are completed by team member #2 and team member #3.

Step 2: Create directories / folders and files:

- Create a directory **a3/** and the following files:
- **game.html** as the main page for the project
- **css/ directory** with a **style.css** file to hold the CSS rules
- **images/** directory to hold the images used in the project

Step 3: Write the HTML5 code:

Create the **game.html** web page and include the following HTML:

- Link to the CSS file at css/style.css. The HTML will be similar to:
```
<link rel=”stylesheet” type=”text.css” href=”css?main.css” media=”all” />
```
- Link to the JavaScript file at js/action.js:
```
<script type=”text/javascript” src=”js/action.js”  ></script>
```
- Register a function call to drawGameBoard(8,8) to the body’s onload event:
```
<body onload=”drawGamegoard(8,8)”;>
```
- Include the following elements in the body section:
```
<p><span id=”member1”Who</span> initiates the checkerboard. </p>
<p><span id=”member2”Who</span> populates the checker coins. </p>
<p><span id=”member3”Who</span> randomises the moves. </p>
<div id=”board”></div>
```

Step 4: Create the CSS code:

Add the CSS selectors and rules to **css.style.css.** Include the following CSS selector and rules:

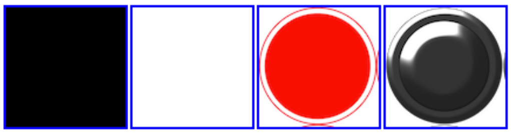
- Create a td *type* selector and set the size to: 100px width and 100px height with visible borders, including:
- Create a black class selector with a *darker* background colour (the checker board)
- Create a white class selector with a *lighter* background colour (the checker board)
- Create a redpiece *class* selector with a background-image :
  url(“../images/redpiece.png”); (Note: you must create the image and locate it in the images/ folder)
- Create a blackpiece *class* selector with a background-image :

url("../images/blackpiece.png"); (Note: you must create the image and locate it in the images/ folder)

- Test and validate the CSS code by adding the following HTML code to the game.html web page:

```
<table>
    <tr>
        <td class="black"></td>
        <td class="white"></td>
        <td class="redpiece"></td>
        <td class="blackpiece"></td>
   </tr>
</table>
```

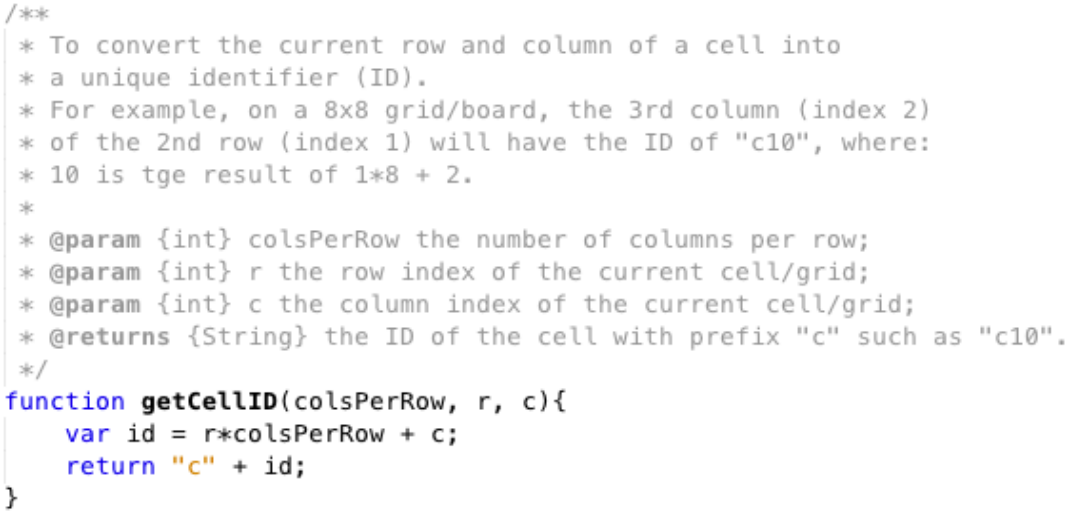
- The result (the output shown in a web browser) should be similar to the following figure. The output must be as shown before proceeding to step 5:



Step 5: Team Member #1: Draw the Game Board/Table:

Add a function to the **js/main.js** file:

1. Create a function called **getCellID()** with three parameters *colsPerRow*, *row*, and *col* in the parentheses the (). The JavaScript Code for this function with comments is provided here for your reading and use:



```
/**
 * To convert the current row and column of a cell into
 * a unique identifier (ID).
 * For example, on a 8x8 grid/board, the 3rd column (index 2)
 * of the 2nd row (index 1) will have the ID of "c10", where:
 * 10 is tge result of 1*8 + 2.
 *
 * @param {int} colsPerRow the number of columns per row;
 * @param {int} r the row index of the current cell/grid;
 * @param {int} c the column index of the current cell/grid;
 * @returns {String} the ID of the cell with prefix "c" such as "c10".
 */
function getCellID(colsPerRow, r, c){
    var id = r*colsPerRow + c;
    return "c" + id;
}
```

2. Create a second function called drawGameBoard() with two parameters for rows and columns in the parentheses (). This function will draw the checker board/grid. Add statements to do the following in the function block (between curly braces) to:

    a. Use document.getElementById("member1") and assign the name of the team member to it;
    b. Create a table element using document.createElement("table") and assign it to a variable;
    c. Reference the **div** element with id="board" by using document.getElementById("board") and assign it to another variable;

d. Add the table element above to the board div above using the appendChild() method;

e. Create a loop and a nested loop (an ‘**outer**’ loop with an ‘**inner**’ loop. In the ‘**outer**’ loop’ the number of repetitions is the number of the **rows** in the parameter.

f. Do the following in the ‘**outer**’ loop structure:

   i. Create a **tr** (for a table row) element using the same document.createElement() method above;

   ii. Assign the newly created element to a variable;

g. The ‘**inner**’ (nested) loop repeats the number of **columns**. In the ‘**inner**’ loop:

   i. Create a **td** (for a table data cell) element using the same document.createElement() method as shown above;

   ii. Assign the newly created **td** element to a variable;

   iii. Assign “**white**” to the **className** property of the above **td** element/variable;

   iv. Call getCellID() function and pass 8, i, j (assume i and j are loop indices (variables) for **rows** and **columns**) as parameter values to convert the current row and column to an unique ID;

   v. Assign the above ID to the id property/attribute of the above td element/variable;

   vi. Use appendChild() method to add the td element/variable into the **tr** element/variable;

h. The 8x8 table/grid should be displayed following step 5

Step 6: (Team Member #2: Show the Game Pieces)

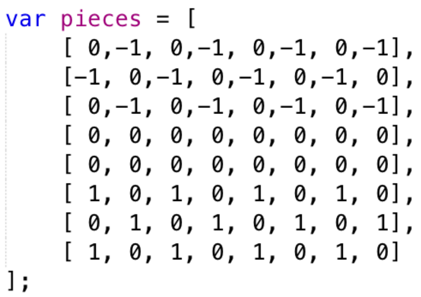
First: team member #1 to add the following code in the **game.html** file:

- An additional **script** tag with: src=”/~2ndMemberDrexelID/a3/js/action2.js ”;
- A new button and event registration:

  <button onclick=”showGamePieces(8,8);”>Show Pieces</button>
  <button onclick=”randomMove();”>Random Move</button>

Team member #2 to do the following:

- Create a new action2.js under /a3/js/ subdirectory;
- Add the following array variable with data in your code (before function):



```
var pieces = [
    [ 0,-1, 0,-1, 0,-1, 0,-1],
    [-1, 0,-1, 0,-1, 0,-1, 0],
    [ 0,-1, 0,-1, 0,-1, 0,-1],
    [ 0, 0, 0, 0, 0, 0, 0, 0],
    [ 0, 0, 0, 0, 0, 0, 0, 0],
    [ 1, 0, 1, 0, 1, 0, 1, 0],
    [ 0, 1, 0, 1, 0, 1, 0, 1],
    [ 1, 0, 1, 0, 1, 0, 1, 0]
];
```

- Create a function named showGamePieces() with rows and columns parameters. In the function:
  - Use document.getElementById(“member2”) and assign the name of team member #2 to it;
  - Create a ‘**inner**’ (nested) loop. Within the ‘**inner**’ loop:

- Assume the variables i and j are indices for the loops (for row and column respectively.
- Call getCellID(8, i, j) to convert row and column to a unique ID;
- Use document.getElementById() with the above ID to reference the corresponding td element and assign it to a variable;
- Create a selection structure (if… else…) to assign the proper CSS class name to the td element/variable.
    - If pieces[i][j] is lesser than zero, assign "redpiece" to the className property/attribute of the td variable;
    - If pieces[i][j] is greater than zero, assign "blackpiece" to the className property/attribute of the td variable;

Step 7: Team Member #3: Random Move

Ask team member #1 to add the following in the **game.html** file:

- An additional script tag with: src="/~3rdMemberDrexelID/a3/js/action3.js";

Team member #3: create a function randomMove() without parameters. In the function:

- Combine Math.random() and Math.floor() to generate a random integer
- between 0 and 8.
- Repeat the above step 4 times to generate four random integers and store them in variables r1, c1, r2, c2 (We will use the first two integers as row and column indices (to move from) and the last two integers as indices (where to move to).
- Use the first two integers **r1** and **c1** as indices to look up the value in the **pieces[r1][c1]** array;
- Assign the above value to replace **pieces[r2][c2];**
- Convert **r1** and **c1** into a unique ID by calling **getCellID(8, r1, c1);**
- Use document.getElementById() with the above ID to reference the corresponding td element and assign its className property to "white";
- Convert **r2** and **c2** into a unique ID by calling **getCellID(8, r2, c2);**
- Reference the corresponding td element with the above ID and change its className according to "redpiece" or "blackpiece", depending on the value of **pieces[r1][c1]** (This changes the style of the target cell to the style of where the piece moves from.)
- Assign 0 to pieces[r1][c1] at the end.

You will submit a written report **not exceeding 2000 words** which will include (**see also the report specification document**):

- The selection of the team leader (member #1)
- The planning and coordinating of the tasks
- The sharing and integration of the allocated software into the final program
- An analysis of the project creation with problems encountered
- The HTML and JavaScript program code (all files) in a text document.
- The complete final project files including all images.
- Screen shots showing the NetBeans IDE project, the TML project code, and the JavaScript program code.
- Screen shots showing the *index.html* file in your XAMPP 'htdocs' folder.
- Screen shots showing the output for your web page in a web browser (all stages including the functions in the program.
- Screen shots showing the output for your web page in the web browser embedded in the NetBeans IDE.
- All screen shots must show the full computer screen including the date and time.

- The report will be submitted in a Word document which will include the screen shots and descriptive text.
- The report must be correctly formatted and presented as shown in the report specification document; this will form part of the grading criteria.

In this assignment all work must be carried out using the NetBeans IDE and the XAMPP server. Tools that automatically generate HTML and XHTML must NOT BE USED.

Learning outcomes

For this assignment the learning outcomes are:

- To understand the creation of a web application using HTML5 with JavaScript scripting
- To practice and understand the use of iteration (loops), arrays, and functions in JavaScript
- To practice and understand the application of methods and properties in the DOM for dynamic content manipulation
- **To understand the dynamics and develop team collaboration experience for system and application development**
- To understand how to format a Word document in the correct report style and save the Word document as a pdf file for printing

Assessment criteria

For this assignment marks will be awarded based on the following criteria:

Task A – **20%**:	The written report
Task B – **50%**	The working program
Task C – **10%**	The Collaboration and project management
Task C – **20%**	The working program files and code with the programming style and documentation

Assignment Weighting

This assignment will contribute **25%** to your overall grade.

Assignment management advice

You should commence work on the assignment immediately as it will take time to complete the design and implementation of you web site, test it for submission, and prepare the written report in English using correct English grammar.

Anticipated work effort

This course assumes no prior knowledge of the topics covered in the course. The average student will be expected to spend about 5 hours per week (for each team member) to complete and submit this assignment

Assignment Submission

The assignment will be submitted as set out in the INFO 151 *Moodle* course web site.

Important Notice

You are reminded of the Lanzhou University Regulations and Disciplinary procedures relating to cheating details of which are set out in the Lanzhou University Regulations and Policies.

Except where the assignment is a team-based, all individual assignments, preparation and submission of the final piece of work must be all your own work. Close similarity between assignments will result in an investigation into cheating. It is not advisable to show your work to your colleagues or to share information relating to your assignments.

If you have any doubts about the extent to which you are allowed to collaborate with your colleagues, or the conventions or the conventions for acknowledging the sources you have used consult the course documentation and, if you are still unclear, your course tutor.

You must ensure that you acknowledge all sources of information used in the assignments. Work discovered to be the result of collusion or plagiarism will be dealt with under Lanzhou University Disciplinary procedures with penalties as set out in the Lanzhou University Regulations and Disciplinary procedures

[September 2019]