CO583 An Introduction to Programming and Web Technologies
Assessment 3 - Integrating JavaScript, HTML, and CSS

You may seek general help from wherever you like, but the work you submit must be your own. We will run checks on all submitted work in an effort to identify possible breaches of this rule, and take disciplinary action against anyone found to have committed plagiarism. Further advice on plagiarism and collaboration is available at this link: https://www.cs.kent.ac.uk/students/plagiarism-faq.html

Submitting your work

Create separate HTML files with .html extension, named “questionX.html” where X is the number of the question. Please submit a single ZIP file containing all your work.

You will gain marks for partial solutions, so please have a go at every question. Submit your answer files via the link on the Task 3 section of the course Moodle page. You may re-submit your work as often as you like, but only the last submission will be kept and marked.

Question 1 - Interactive Art [50 marks total]

The aim of this question to make some interactive art which combines your knowledge of forms, using JavaScript to adapt the styling, and keyboard input.

Part a). User setup [10 marks]

Create a webpage which provides a form with a drop down box to select a number between 1 and 10 and a radio button to select between ‘Red’, ‘Blue’ and ‘Green’, with a button “Go” to trigger some JavaScript given as your solution to the next part. Put some appropriate labels on the input fields.

Part b). Initial canvas [15 marks]

Using the input from the user generate a series of boxes added to the page where:

- The number of boxes is equal to that specified by the dropdown box.
- The background of each box is a random variant of the colour chosen by the user (you can use the “rgb” format of CSS colours);

For example, for 6 red boxes your code might produce something like:

The approach should be such that the user can repeatedly update he form and add more boxes to the page, e.g., after selecting 4 red boxes (then clicking “Go”) and selecting 10 green boxes (then clicking “Go”) you might get something like:

Part c). Select and grow [10 marks]

Adapt the code from before such that every box is clickable.

Every time the box is clicked it show grow in width by twice its current width.
Every time the box is right-clicked it should shrink in width to half its current width.

Part d). Keyboard [10 marks]

If the user presses key ’s’, swap the position of two randomly selected boxes.

Part e). BONUS [5 marks]

Add any other feature you like for a bonus 5 marks. Please make the new feature clear or document it with comments so I can understand what it is.

Question 2 - Tic-Tac-Toe [50 marks total]

The aim of this question is to build the Tic-Tac-Toe game (otherwise known as Noughts-and-Crosses). The following sub questions give some guidance:

1. Create the "game board" which comprises a 3x3 grid of boxes. Recall that in Class 4 Question 2, you created a ‘game board’ grid for Noughts-and-Crosses which you might like to use as a starting point.

   Consider that later you will need to check for "winning conditions”, e.g., a line of three Xs or Os. Think about how to structure the code so that this task is easier in the future. You will want an easy way to access the value of a box based on its row and column position. Therefore, you may want to generate the game board dynamically (using createElement etc.) and storing element objects in a two-dimensional array (see Class 7);

2. Provide the following behaviour: when a box is clicked it is filled with either an X or O. When another box is clicked it should then be filled with the alternate marker for the other player.

   (Hint: you'll need a piece of state recording the current player. Each click event updates a particular box and switches the player.). Make sure that a box can't be changed from an X or O once it has been

3. After each user update of a box, provide a function that checks whether the winning condition has been met and provides a message if so. Consider the various possible winning conditions and how to write a function that concisely checks these.

4. Add some CSS styling.

You can receive partial marks for a game with some of the right features.

Be creative.