CS230 - Web Information Processing

Assignment 2

Assignment Release Date:	24-02-2020
Submission Due Date:	06-03-2020
Feedback Due Date (estimated):	13-03-2020 (for assignments that make Due Date)
Support Laboratories	Lab 03, Lab 04 (Two Weeks)
Total Mark:	10%

This Assignment is worth 10% of the Web Information Processing CA Component.

This is an open-book, graded assignment. You may use online resources for reference purposes only to help with the assignment. Please cite all references as comments in your submissions. You cannot directly reuse HTML/CSS/JS **solution code** from online sources. **You must not engage with another student, in person or electronically (phone, social media, etc.) to secure assistance with this assignment. If you do so you will receive an automatic fail (0%).** We will perform similarity checks on submitted assignments to check for collaborative efforts. A reasonable attempt at this assignment will gain you 5% of your continual assignment marks.

Assignment 02 - The Simon Memory Game

The Simon electronic memory skill game, invented in 1978 by Ralph H. Baer and Howard J. Morrison, creates a series of tones and lights and requires a user to repeat the sequence. If the user succeeds, the series becomes progressively longer and more complex. Once the user fails or the time limit runs out, the game is over. Simon is named after the simple children's game of Simon Says, but the gameplay is based on Atari's unpopular Touch Me arcade game from 1974.

The device has four coloured buttons, each producing a particular tone when it is pressed or activated by the device. A round in the game consists of the device lighting up one or more buttons in a random order, after which the player must reproduce that order by pressing the buttons. As the game progresses, the number of buttons to be pressed increases.

Assignment 02 - Requirements

You are required to re-create a version of the Simon game as a (HTML/CSS/JS) web app. You are only required to flash the coloured buttons in each round (no audio signal is required). Your app must implement the interface shown in Figures 1, and only requires implementation of the following single-player game play.

The Single-Player Simon Game is as follows:

1. Click the START button to begin, the game status indicator (the red/green light below the START button) switches from RED to GREEN. The game will begin 3s after the light turns GREEN.
2. Simon will give the first signal (randomly flash a coloured button). Repeat the signal by clicking the same colour button.
3. Simon will duplicate the first signal and add one. Repeat these two signals by clicking the same colour buttons, in order.
4. Simon will duplicate these first two signals and add one.
5. Continue playing as long as you can repeat each sequence of signals correctly. After the 5th, 9th and 13th signals in a sequence, Simon automatically speeds up the interval between signals.
6. If you fail to repeat a sequence exactly, or if you take more than 5 seconds to repeat a signal, Simon responds by flashing all four buttons simultaneously five times. This means you have lost, and the sequence of signals ends. The game status indicator switches from GREEN to RED and you will have to click START button to begin a new game.
7. Your progress (the number of correctly repeated signals) for the game just completed (last game) is shown in the display to the right of the START button. The all-time highest score is shown in the display to the left of the START button.

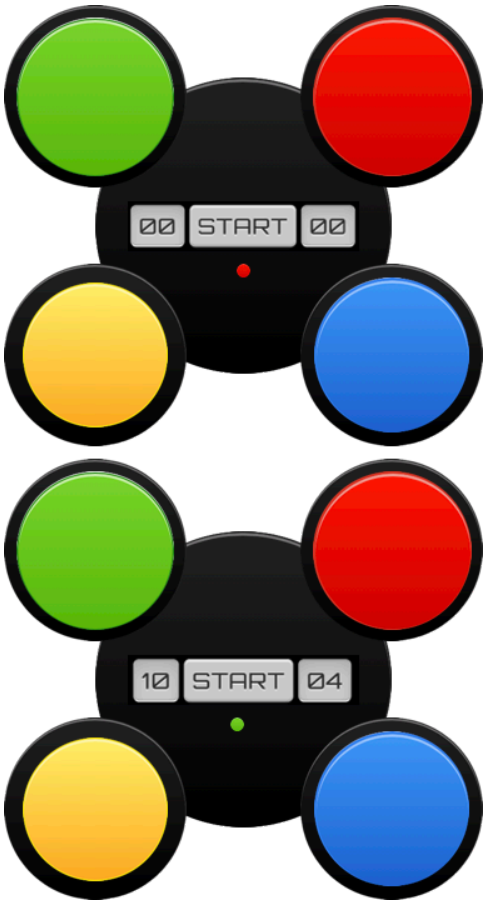

Figure 1

Required Interface for CS230 (HTML/CSS/JS) Simon Application

Assignment 01 - Development Notes

Please adhere to the following development requirements:

1. All game interface elements must be generated using HTML elements together with appropriate CSS styling. Generally, your colour schemes, must follow a similar colour scheme to that shown in Figure 1. You may not use pre-constructed images for any of the game elements. All of the elements shown were created using HTML and CSS in codepen.io, and (for convenience of display in this document) were combined using Keynote. The individual elements used are shown in a screenshot of the partial CodePen Interface in Figure 2.
2. The position of the interactive and display elements must, in general, be structured as shown in Figure 1. You cannot use another layer or design for your assignment.
3. The font family used is “Orbitron” and you may load it from https://fonts.googleapis.com/. You may not use any other font as part of the display.
4. You may not use a CSS framework, such as Bootstrap, for this assignment. You may, if you wish, use the jQuery Javascript framework. If you use TypeScript, or similar, and transpire to JavaScript, you need to provide all sources. Your app only needs to run on a desktop browser such as Chrome - you do not need to ensure it works on every browser (mobile browsers, for example).
5. You must comment your code, clearly indicating, how your code implements the Single-Player Simon game described above in the “Assignment 02 - Requirements” section.

Figure 2

The HTML/CSS Elements generated using CodePen

Please note that there are many sample (HTML/CSS/JS) solutions for Simon-like games available online. While it is fine to consult these, and accompanying articles, for references, you may not re-use code from these projects. Please cite your reference sources in your codebase. We will search and identify online Simon game code for the purposes of checking against submitted solutions in instances where we have concerns about code originality.

IMPORTANT SUBMISSION DETAILS

Before submitting your assignment students should check that their solution works in Chrome and/or Firefox. Please indicate the Browser, Lab Operating System (Linux/Windows) and Browser version used for testing (as a comment in your submitted code).

All work must be submitted via Moodle (see "Assignments" section for submission). Work submitted via other means will not be accepted unless you have prior arrangements with the Head Demonstrator (Behnam Faghih). All work MUST be submitted by the due-date deadline. Late submissions will not be accepted.

*The assignment submission is a zip file named "**assignment-02-xxxxxxxxx.zip**" (where "xxxxxxxxx" is your student id) containing a solution file named "**assignment-02.html**" together with any other resources used in the assignment solution. External CSS and Javascript files should be named "**assignment-02.css**" and "**assignment-02.js**", respectively. Please ensure that all external files use relative directory referencing, rather than hard-coding the files' location.*