# EECS 1012: LAB 08 – Code Breaker –Part 1: client side (Nov 09-11, 2020)

## A. IMPORTANT REMINDERS

1) Note that the deadline of remaining labs is on Wednesday night.
2) Each lab including the pre-lab mini quiz is about 2.0 % of your overall grade.
3) You must attend your own lab session (the one you are enrolled in). If you need to change your lab enrollment, you should go to the department. Instructors or TAs cannot change your enrollment. TAs are available via Zoom to help you. The attendance is optional, but is highly recommended. You can also have your work verified and graded during the lab sessions. Feel free to signal a TA for help if you stuck on any of the steps below. Yet, note that TAs would need to help other students too. In case you run out of time, the submission you make over eClass will be marked by the TAs after the lab ends (possibly not by the same TAs who assisted you during the lab session).
4) You can submit your lab work anytime before the deadline. We do not accept late submissions.
5) You must complete the pre-lab quiz posted on eClass no later than the first 15 minutes of your lab time.

## B. IMPORTANT PRE-LAB WORKS YOU NEED TO DO BEFORE GOING TO THE LAB

1) Download this lab files and read them carefully to the end.
2) If you are not familiar with the Code Breaker board game, visit https://en.wikipedia.org/wiki/Mastermind_(board_game). Note that the one we make in this lab has 5 code pegs (not 4).
3) You should have a good understanding of
    - Document Object Model https://www.w3schools.com/js/js_htmldom.asp , in particular `createElement`, `append`, etc. https://www.w3schools.com/jsref/dom_obj_document.asp
    - jQuery https://www.w3schools.com/jquery/
    - Revisit Slide 09-16 to 09-18 and Sides 09-23 to 09-29 of the lecture notes and identify the jQuery commands and review their meaning. We highly encourage you—prior to go to labs this week—to do two sets of exercises on jQuery selectors and events available in w3schools, starting here: https://www.w3schools.com/jquery/exercise_jq.asp?filename=exercise_jq_selectors1
    - JSON https://www.w3schools.com/js/js_json_intro.asp , in particular `stringify`, and `parse` methods. https://www.w3schools.com/js/js_json_stringify.asp and https://www.w3schools.com/js/js_json_parse.asp
4) Understanding AJAX `request` and `response` is also an asset, even though we are not going to use it directly. https://www.w3schools.com/js/js_ajax_http_send.asp and https://www.w3schools.com/js/js_ajax_http_response.asp . This topic will be clearer by end of next lab when we address about the server side too.

## C. GOALS/OUTCOMES FOR LAB

1) To become familiar with JQuery and JSON
2) To work more with DOM

## D. TASKS

1) TASK 1: Client-Side of the Code Breaker Game.

## E. SUBMISSIONS

1) Manual verification by a TA (optional)

You may have one of the TAs verify your lab before submission. The TA will look at your various files in their progression. The TA may also ask you to make minor modifications to the lab to demonstrate your knowledge of the materials. The TA can then record your grade in the system.

2) eClass submission

Create a **folder** named "**Lab08**" and copy **all** of your HTML and JS files; Once you are done, compress the folder and upload the zip (or tar) file to eClass.

## F. FURTHER DETAILS

**Task 1:** Use html, CSS, and JS to design the client side of the code-breaker game. We have provided you with a starter code, but you can design with any style you wish. If you like to use our starter code, you should:

1) Open `code_breakerV0.html` and save it as `code_breaker.html`. You should read the comments in your `code_breaker.html`, and make changes such that your html file becomes similar to `code_breakerV1.html`, eventually. Note that your html file instead of having 150 lines is going to have 28.
2) Also, open `code_breaker_clientV0.js` and save it as `code_breaker_client.js`. In the `createGameBoard` function of `code_breaker.js`, you should read the comments and write the code for all 15 lines specified by "`//...`". These lines dynamically build the html tags that you just deleted from your html file: you are creating those tags via your js file at run time (dynamically).

If you make the above changes properly, your code_breaker should work fine, that means you can start playing the game by opening your `code_breaker.html` in Firefox. Note: we make sure that our `code_breaker_server` is up on `indigo.eecs.yorku.ca` from November 8 until November 11. In the next lab, you will complete your own `server_side` code such that you be able to run it on your computer any time.

**Show your code-breaker code to your TA (optional).**

Note that the CodeBreaker project is a great source of learning. Hence, read all comments carefully and make sure you have a clear understanding of it.

Next week, we will complete this project by adding the server component to it.

## G. AFTER-LAB TASKS (THIS PART WILL NOT BE GRADED)

In order to review what you have learned in this lab as well as expanding your skills further, we recommend the following questions and extra practices:

1) You may want to learn a bit more about AJAX `request` and `response`. We will use them indirectly in the next lab. https://www.w3schools.com/js/js_ajax_http_send.asp and https://www.w3schools.com/js/js_ajax_http_response.asp
2) You may also want to start learning about `express JS`, as we are going to use it for the server side of CodeBreaker in next lab.
3) It's good to start thinking to add a Run button as well as the Java equivalent of each of your solutions in your Learning Kit project. We highly recommend you pick `Java` as your next programming language. In the 3rd panel of your learning kit provide a screen shot of your Java code. Some students may prefer to use another language, such as C++, C, C#, etc. In next couple of weeks, we will announce some bonus points for students who have implemented the Run button as well as an alternative solution in the 3rd panel (for instance, in Java).

Please feel free to discuss any of these questions in the course forum or see the TAs and/or Instructors for help.