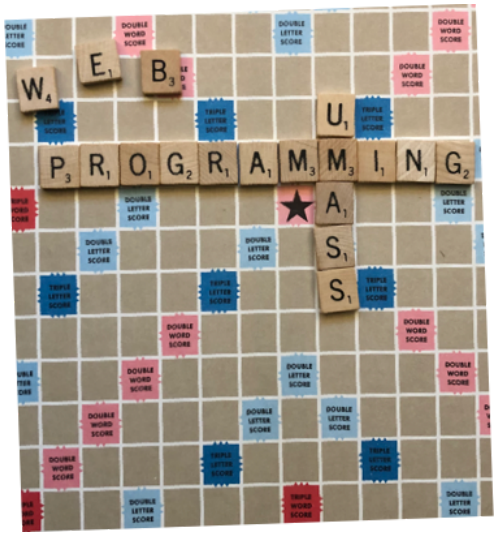
COMPSCI 326 - Web Programming
Homework 9 - Rules & Fetch - *individual assignment*
due November 15, 2021, 11pm EST
(GitHub classroom link: https://classroom.github.com/a/7P6ex0uB)



This is the ninth part of a series of assignments around the game of Scrabble. We hope that it will be a fun experience in progressively learning all pieces of modern web development, so as to engineer a fully functional game. In this assignment, we will add (most of) the Scrabble rules regarding placement and scoring, as well as use our previously written API.

Please submit this assignment on GitHub Classroom. It will be helpful to come up with test cases, and we encourage you to share them amongst each other; this will make everyone's code better and is actually how Quality Assurance (QA) can work in practice. However, this is an individual assignment and you **cannot share code**; submissions will be run against plagiarism detection tools. Additionally, we will be spot checking the code for good coding practices. It is expected your code **does not** contain (1) extraneous variables/code, (2) missing semicolons, (3) missing curly braces, (4) use of double equals, (5) use of `let` when a `const` would suffice, (6) use of `var`, (7) inconsistent return values. Furthermore, you should use whitespace consistently and to make the code legible. Now that you've learned how to use ESLint, it should be easy to satisfy these requirements.

You will find a template with support code when you create your repository. You should import the files you previously used in the `client` and `server` directories respectively (or of course use the solution posted on Slack). Note *both* directories will be used this time. **Please do not rename any of the existing files or change the directory structure.** You are free to create more files and import them. However, you **cannot** use any external modules beyond those provided without prior permission.

1. Rules

In this first part, we will add some of the rules of Scrabble that we didn't include in the past assignments. Doing this is relatively hard, and out of scope for this class. Therefore, we will provide a version that is updated to support (most of) the actual Scrabble rules. We will provide an updated `game.js` (which contains most of the changes) when this assignment is released, and a full version once everyone has submitted HW#8. If you've done extra styling / features for previous homeworks, please merge both versions instead of just keeping the provided one!
To be clear, for this part of the homework, you need to either:

- Incorporate the provided `game.js` code into your own version of the game.
- If you do not have a working version, use the provided version once it is released.

2. Serving our browser code with the server

We will again provide some code for you to serve your client files (`index.html`, `main.js`, ...) through the server. This is necessary to avoid CORS issues, where you load different parts of your website through different servers, leading to browser errors. We will provide a `index.js` file that will contain the previous code from HW#7 and be capable of serving files from the `client` directory. You will need to make small modifications to this file to write the correct MIME type in the header when serving a file. In our case, you can only worry about HTML, CSS, and JavaScript files. Not doing this can cause browsers to fail to load files (notably JavaScript files).

3. Using the API endpoints

Lastly, you will integrate the API endpoints we wrote for HW#7. Whenever a word is played, you should use `/wordScore`. You will also add a button to end the game, and call `/gameScore` when that button is clicked. You should record both players' scores when the game ends. You can optionally add logic to end the game following the rules (i.e. no tiles in the bag, no more possible moves, …), but you need to also have a button to make testing easier. Finally, you will display a table of the top word scores and game scores, using `/highestWordScores` and `/highestGameScores`, respectively.