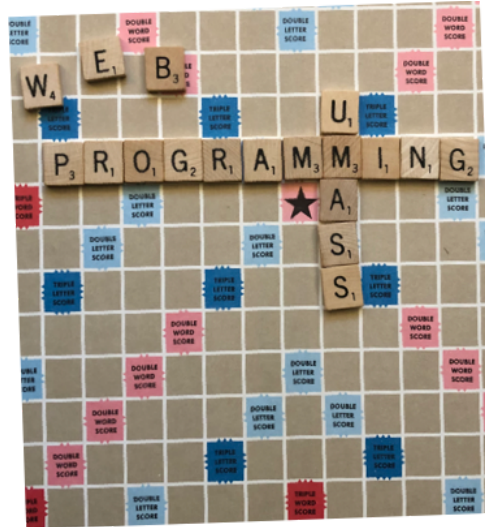


COMPSCI 326 - Web Programming

Homework 4 - Scrabble Renderer - *individual assignment*

due October 17, 2022, 11pm EDT

(GitHub classroom link: https://classroom.github.com/a/t_hhkkDC)



This is the fourth 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, you will start writing HTML to render the Scrabble board on a webpage.

Please submit this assignment on GitHub Classroom. There will be an automated grader that will check the functionality of your submission. 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`. 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 this homework's skeleton included in your repository when you create it. 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.

0. Using a Local Web Server

One issue you can face (discussed in class) is that your browser won't let you import files when working on your local machine. You might get errors like the one below.

Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at <filename>

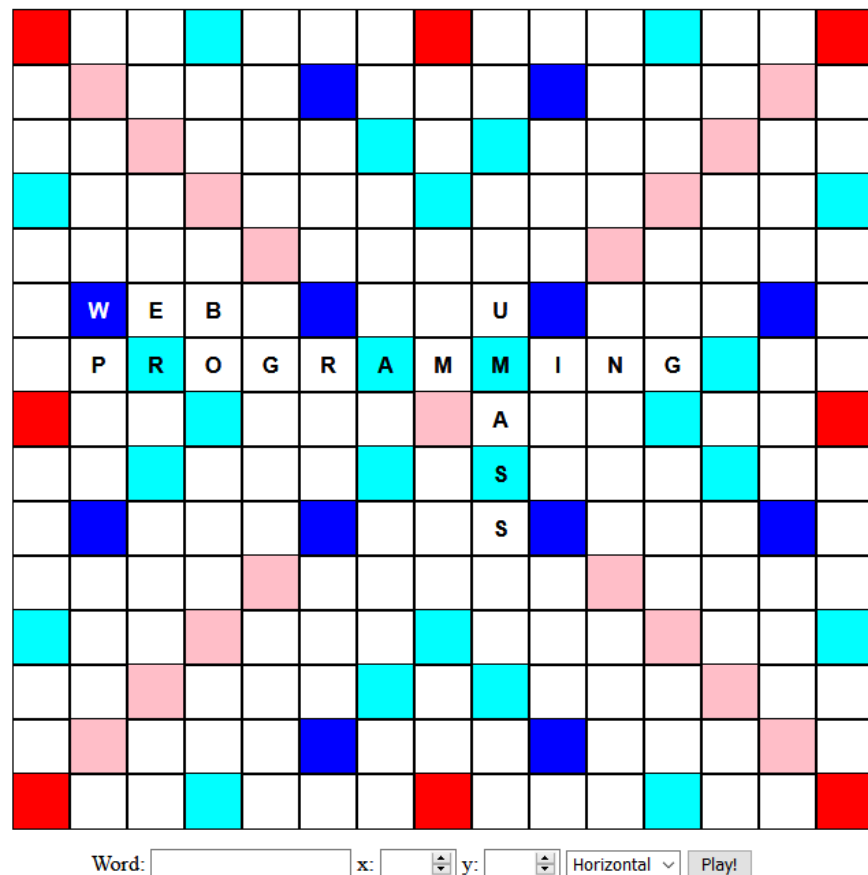
The reason this happens is your browser is blocking your page from loading local resources. While this may seem weird, you can see why it makes sense for it to deny websites from accessing arbitrary files on your machine.

There are numerous ways of getting around this when developing:

- Use Python's built-in web server: `python3 -m http.server`
- Use [the Live Server extension](#) on VS Code. Just install the extension, right click your file in VS Code, and select "Open with Live Server".
- Use [the http-server npm package](#) provided. Run `npm i` to install, then `npm run http-server` from your homework's directory.

What these do is start a local webserver on your machine, and serve all files as if they were coming from a website, and not your filesystem.

1. Rendering the Board



For this homework, you will update your Game class to make it capable of rendering the board using HTML. You can be as creative as you want, but need to meet the minimum requirements below (as seen on the image above):

- You should display the 15x15 grid using CSS grids, which **you must generate programmatically** (via JavaScript; your HTML cannot contain any grid items!).
- Special tiles should be colored (same special tiles should be the same color, see above) **using classes and CSS styles**.
- There should be controls at the bottom, asking the user for a word, a starting position, and a direction, as well as a submit button. When clicked, the board should be updated with the new word. If the new word does not fit or would go out of bounds, nothing should happen (but no exception should occur either).

Again, feel free to be more creative (column / row headers, high quality graphics, sprites, 3D animations, narration, videos, controller schemes, cute cat pictures ...), [we want you to express yourself](#).