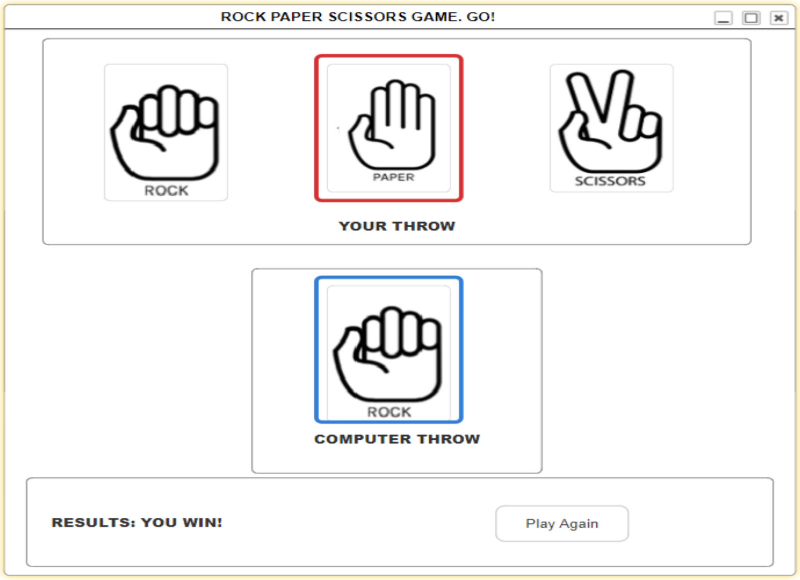
CSC372: Web Development
Assignment 4: Callbacks – Rock, Paper, Scissors
Due: Friday, March 7, 2025

In this assignment, you will practice putting together all the skills we have reviewed so far by building a simple rock-paper-scissors game on a single web page. You have creative freedom, but your web page should emulate game play by letting the user play against the computer. On the user's turn, they will pick their throw by clicking one of the 3 choices. On the computer's turn, a random choice will be made by the computer, then a winner will be determined. A sketch of the draft page is shown below.



Theme and appearance are up to you, but you must fulfil the following specs.

1. The page should contain at least three sections; the player throw, the computer throw, and the outcome.
2. The player throw section contains three figures, with images representing the three moves. When the user clicks on an image, a border should appear around it (to mark the player's throw), and the computer throw is then decided.
3. The computer throw section should show one image at a time. When the page is first opened (or reloaded), a question mark image should be displayed. While the computer throw is ongoing, each of the three play images should shuffle in every half second. When the computer decides its throw (randomly), then the chosen throw image should be displayed as shown above. The computer should spend a total of 3 seconds "thinking".
4. The winner of the game is then decided (computer, player, or tie), and the outcome section should be updated accordingly.
5. For some extra credit points, keep track of the number of wins, losses, and ties for each page reload and update on-screen displays. You can add a fourth section in the HTML file to display the current score. Make sure that you add a "reset" button that clears the counters and displays.

Most of your work will be in the callback functions. At minimum you will need functions to highlight the player choice, shuffle the images for the computer throw, decide the winner (your logic must account for all 9 throws), reset the game, etc.

To submit your work, use your GitHub pages site again. In your *<username>.github.io* repository, create a hw4 subfolder for your work. Organize your files so they are easy to locate in subfolders as discussed in class. Do not directly use styles and scripts in the HTML file. All styles should be in .css files, and all scripts should be in .js files.

Be sure to edit the homepage for your GitHub Pages site to include a link to this subfolder, just like you did before. When you push your changes to your github.io repo, your new sub-folder will also be deployed to your GitHub pages website.

Submit on Canvas: the link to your deployed page at github.io. This link should not have "github.com" on it!