

WDD DJ

Required Tool: Text Editor and Google Chrome Web Inspector

Welcome to your last homework of the decal! We hope it's been fun doing these homeworks, and that you guys now feel readily equipped to make anything your heart desires from here on :)

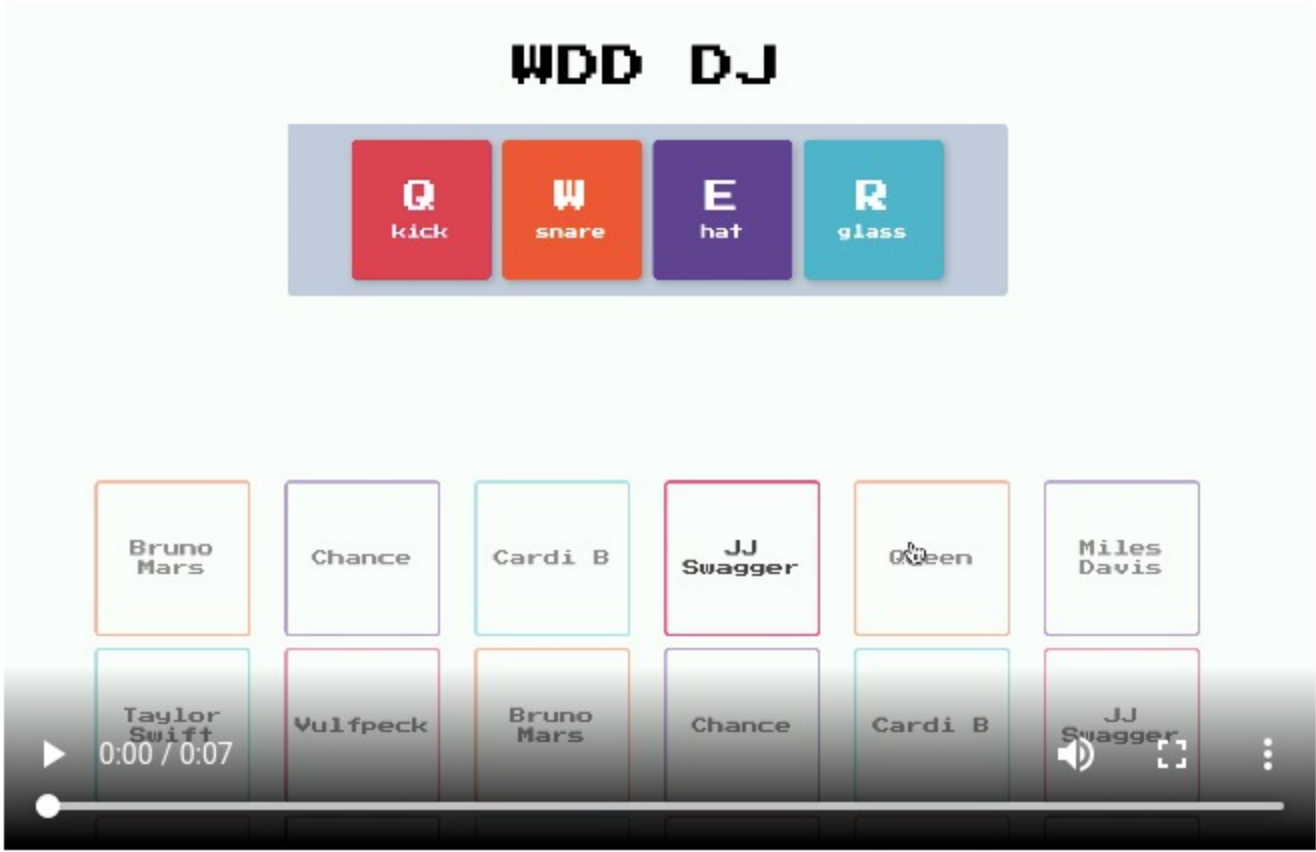
This week, we will work with javascript and more jQuery functions to turn your computer into an instrument.

Assignment

You can download the assignment file here:

Download File

At the end of this homework, you should have something like this:



Background info

For this assignment you will need to refer to both the HTML and the JavaScript files, but you will only need to alter the JavaScript.

Let's get familiar with a few new things in our JavaScript file. Take a look at the funky looking `sound` variable. This is the JavaScript version of a python dictionary, which is a JavaScript object. If you're unfamiliar with this syntax, each item to the left of `:` serves as the "key" to the "value" to the right of the `:`. In our case we have numbers as keys corresponding to strings as values (i.e. `81 : "kick-sound"`). We use the key of the object to access the value. In this specific example, we could get 'kick-sound' by doing this `sounds[81]`.

The numbers for this specific object are special, they represent each key's key code. We need this to know what key has been pressed.

Feel free to analyze our sound function in the JavaScript file. We have left comments describing how it works, but you don't need to know this information to complete the lab. *What you should know* is that on a given key press, the key code is used get the correct soundid from the JavaScript object, `sounds`, which is used to get the audio file from the html.

Instructions

Without visuals, it's kind of difficult for the user to figure out what is happening. So let's use our knowledge of jQuery and conditionals to create a light overlay layer when the user presses a key. We will have 5 different statements -- one response for each key in `sounds`, and one more for any other key.

- Using `e.which` (what is this?) and the specific key codes for each key, how can we check if it is true that a specific key is pressed? Create a conditional for each of the key codes we listed in `sound`. `e.which` can be interpreted as the value of the key that was pressed at that moment.
- Within each conditional, show the appropriate `button-overlay` and then fade that overlay out after 100 milliseconds. You should now have 4 complete conditionals.
- Add one final statement for when a user presses any other key.
- Within this condition, let's demonstrate the power of the cowbell. There can never be too much cowbell. With jQuery, add some text to the `#display` about anything you want (preferably something about the mighty cowbell) and don't forget to show it since `#display` is set to none. Then fade that out after 400 milliseconds.
- For a better user experience, the staff recommends adding `$("#display").stop(true, true);` before the start of your conditionals -- it will interrupt any previous animations before starting a new one.

As a DJ, you need quick access to all of your music, but you're also lazy and don't want to independently write up each artist in your html. In web programming, loops are used to keep our code clean and let the computer do the repetitive tasks.

Add a for loop in the JavaScript that will run up to 30 times. For each iteration, append an `.artist-item` with an artist name from our `artists` JavaScript object to the `#artist-container`. We can access an artist in `artists` by their key. We have 8 artists, but 30 items, so how can we cycle through the artists?

► Hint

Mission accomplished! Admire your work and jam out!

Submission

Submit the following files as a **zip** file through Assignments under Resources tab of the WDD Portal:

- index.html
- assets/css/style.css
- assets/img/board.svg
- assets/js/script.js
- assets/sounds/cowbell.wav
- assets/sounds/glass.wav
- assets/sounds/hat.wav
- assets/sounds/kick.wav
- assets/sounds/snare.wav
- design_responses.html

Your assignment must be submitted as a **zip** file. Submission will automatically fail if your submission does not contain the index.html or if is not a zip file.

This assignment is due by Monday, **April 29th** at 7:00 PM.