Project 1

IGME 330.03

Ian Effendi

Documentation

Direct Banjo Link [https://people.rit.edu/iae2784/330/projects/P1/project-1.html]

REQUIREMENTS

Usability and overall UX
I wrote the CSS for the page from scratch and developed a neat little color-coded system for the element classes.

Interaction Design
I have almost one of every type of input available (save for textboxes). There are two select options, 1 set of radio buttons, and multiple checkboxes and sliders for all your widget needs.

CanvasAPI
I provided lines, curves, circles, and a gradient when drawing. I was not able to use ctx.getImageData() in the final project.

Web Audio API
I enable the visualizer modes for frequency and waveform datasets. I use compressor, gain, biquad frequency filter, delay, and waveshaper nodes as well.

Media and Presentation and CSS/HTML
All sound clips are in the proper format and of the proper length. There are embedded fonts in the page. The page was not left in the default styling and semantic tags are used.

Code
No jQuery! No other JS libraries either! I use the 'use strict'; command. I used ES6 classes to hopefully make good use of the D.R.Y principle. Conventions are followed for naming variables and I tried to make every line end with a semi-colon - hopefully I didn't miss any of them. Comments are placed in the main class functions. No debugger statements are left. The only code borrowed or inspired is noted in the references below. The only thing I didn't do was follow the IIFE principle. I used globals.

Above and Beyond
I did more than one Web Audio API node and found the tempo of the song playing.

CHALLENGES

My largest struggle was navigating the way Javascript handles numbers. I wasn't exactly strict in my usage of parseFloat() and parseInt() methods, but, when it came time to calculate the delta time (for dealing with the circle pulsing), it was very close to the deadline that I realized my elapsed time wasn't updating. By this point, my codebase was so large and spread apart, that, it was hard for me to get around the issue without using globals. I ended up submitting the zip file of an older working copy, but, managed to fix the things that were causing me issues shortly after 6:00pm.

If I could do it again, I would certainly attempt the way I've structured my codebase in a different way. My next attempt will focus a lot more on the operation speed of the visualizer. As of right now, it slows down when all the effects are on and it doesn't stream large files, opting to load huge songs into the memory. If I had more time, I would have added the ability to read in the song's artists from the element dataset. I also wanted to spend more time styling the OS elements, but, alas, the normal styles and appearances for the controls will have to suffice.

RESOURCES

Web Audio API

[https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API]
I made liberal use of the Mozilla documentation when it came to utilizing commonplace functions, such as getting a random integer. Of note, Mozilla's website was very useful when it came to creating the Web Audio API nodes in the way that I did. For example, one specific link is for the random integer method: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random]. I also received the WaveShaper curve creation method from Mozilla.

Beat Detection using JavaScript

[http://joesul.li/van/beat-detection-using-web-audio/]
I also utilized some resources regarding calculation of tempo and beats in a song. It would not have been possible for me to figure out how to mathematically find qualities in a song, without using Joe Sullivan's reference material (blog post and code snippets).

C2C

[https://www.discogs.com/artist/109085-C2C]
I pulled most of the songs for this visualizer test from C2C's discography

(Exception: 'Sweet Dreamz' is by a youtube beat producer:
[https://www.youtube.com/watch?v=92nsfAVgUSM]).

Grade

I didn't have a partner for this project. I think I should get at least a grade of 80% - given the following penalties:

- Documentation wasn't submitted on time; it was late.
- IIFE wasn't accomplished.
- The imageData method wasn't used; I ended up using context.filter() methods instead for blurring, sepia, grayscaling, and inverting colors. I really wanted to do more but I've run out of time.

If you took off 5% off for each, I'd end up with that grade (including a 5% penalty for lateness). I feel that I went above and beyond with the extra things I did, regarding detection of tempo, styling the page, and utilizing the compressor, biquad frequency filter, and waveshaper nodes.