Owen Avon
March 23, 2022

Project Overview

My final project relates around the theme of audio. In specific, I have created, and will continue to create an application that centers around the theme of audio input manipulation, with the use of JavaScript, jQuery, p5.js, getUserMedia, and Web audio API. Simply put, the application will require the user to record 30 or more seconds of audio input. Once recorded, the user will be able to listen to their recording, and add multiple effects with the use of sliders in real time. Lastly, the user will also be able to choose from existing sound filters, in which they can add additional effects to, if so desired. The filters are still under conception, but may relate to the below topics:

- Analog radio
- Underwater interpretation
- Alien interpretation
- Child voice
- Robotic 20th century computer voice

Once the user is satisfied with their recording, they will have the ability to listen and download the edited track as a .wav file. Additionally, a graphical visualizer inspired by the classic iTunes visualizer, will be available to enhance the auditory experience, and allow for a higher degree of user engagement.

Exploration

I am debating about adding a random "filter" button that the user can click to generate an unexpected sequence of applied effects. This would allow the app to remain intriguing, and provide further excitement and creativity. I may even perhaps add a backwards function that randomly gets applied after ever X number of applied effects. Said function would apply the opposite of what the user is tying to apply. Example, if the user pans the audio to +1.3 (right channel), the application will automatically pan the audio to -1.3 (left channel). This sense of opposite reaction could be interpreted as the machine attempting to tell the user what is best for them… an intriguing approach that I may explore.

Artistic Vision

My intended artistic vision is to make the application look modern, sleek, and simple. This will be achieved with the use of selective colour, and attention to modern design. For example, the colour palette will include shades, (off white, greys, etc.) with one or two accent colours. In relation to modern design, I intend on adding the border-radius property to most / all div's. This decision is made solely due to modern design techniques. I am leaning towards creating a horizontal framework viewport, similar to what might be found on a mobile application. The design will include various animations that will help guide the application in both aesthetic and effective objectives.

I plan to include jQuery UI widgets, such as dialog boxes, accordion and potentially a progress bars. This being said, I find that the default style, and themeRoller style options to be very outdated, and undesirable. Therefore, I plan to reverse engineer the widgets and apply custom CSS to the individual classes to create a modern look and feel across all visual elements.

Owen Avon

Technical Challenges

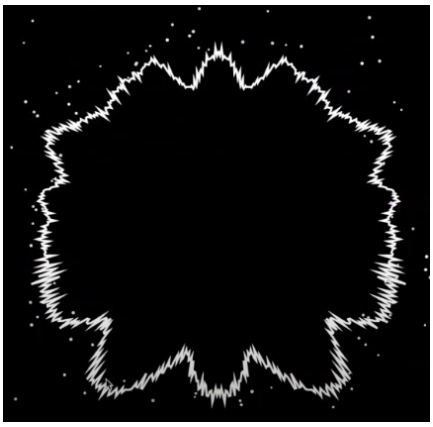
As it stands, I believe I will be faced with two main technical challenges. Primarily, the challenge of embedding p5.js within a jQuery dialog box. Perhaps there are better, and more efficient ways to do this? I thought of this approach as I would like to take advantage of p5.js processing power to help assist with the visualizer. That being said, if this approach is not obtainable, I will resort to creating the visualizer with vanilla JavaScript / jQuery. Internet research brought me to David Lazic's example of "audio-visulaizer" on GitHub, as well as Nick Jones example on CodePen. The later example is currently incorporated in my prototype to illustrate my intentions. I plan to heavily modify the example to meet my needs. Ultimately, the idea is that user will click on the "Visualizer" button, and then be able to see their personal recording as a visual representation.

The second technical challenge that I will be faced with is having the ability to modify the audio track in browser, without the use of a backend database. My assumption is that the browser can edit on top of the audio recording from *src = URL.createObjectURL(new Blob(recordedChunks));* recording, and cache it on the front end? I am not quite sure how to proceed with this. In fact, while creating the prototype, I have been struggling with targeting the src file. Most online examples refer to a "input type = "file" rather than "src". Due to time constraints, the protype targets a "file" to demonstrate the visualizer, along with other effects. I will likely reach out in the near future for guidance regarding this technical challenge.

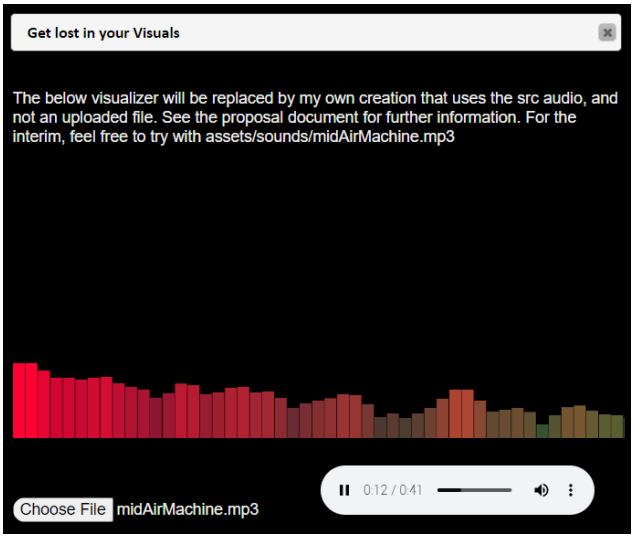
Owen Avon

Visual Sketches / Inspiration

The below sketch is a screenshot captured from the YouTube video, "Code an Audio Visualizer in p5js (from scratch) | Coding Project #17". I genuinely like this circular animation, and the added responsive particles, as they provide depth during the low frequency portions of the audio track.



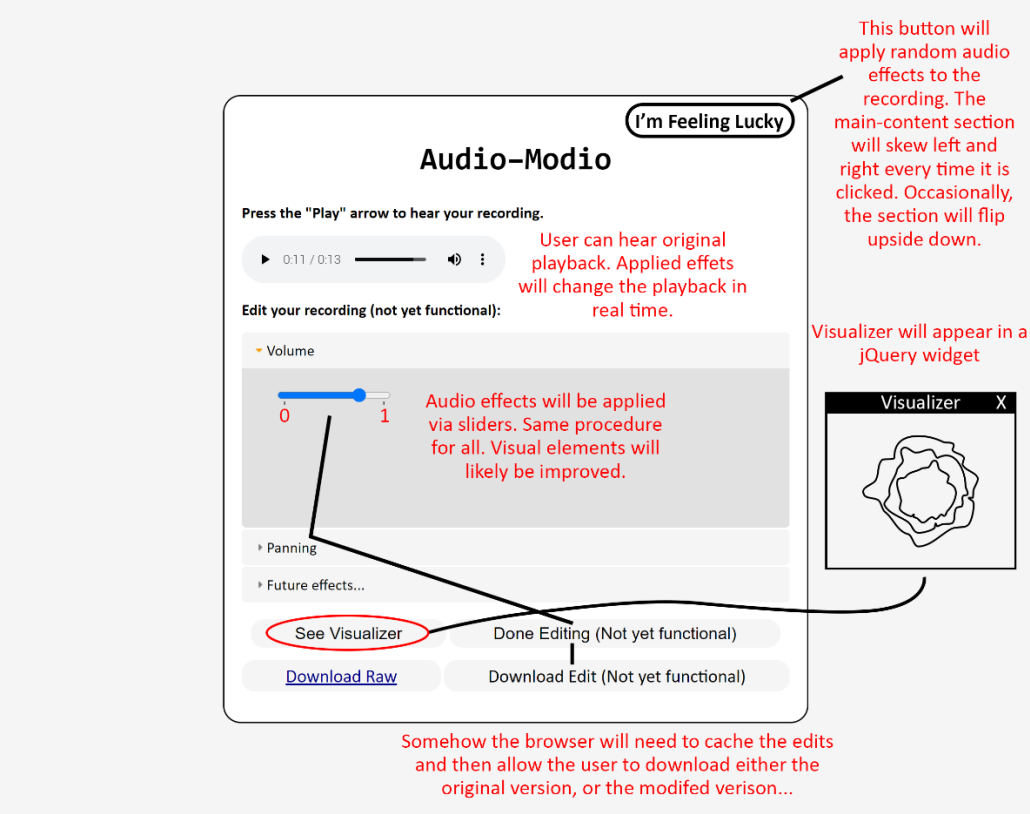
The below screen shot is my current visualizer. This visualizer does not use p5.js, and does not refer to the src audio. It was implemented for illustrative and justification purposes only.



Owen Avon

Visual Sketches / Inspiration cont.

The below screenshot annotates my current protype, while also indicating my future project plans.



Conclusion

I find this concept very fascinating, as every user experience will be personal, unique, and meaningful. I believe a strong application is qualified by one that creates engagement, surprise, and excitement every time it is accessed.