

Mykhal Parson

ENT 4499

Prof. Wilson

11/26/21

Reflection

Creating MIDI Colors was no easy task. Originally, I had thought of MIDI Colors as being an installation in which users could enter a large room and press chords on a large keyboard. The room would change colors based on which chords were entered by the user (minor chords were dim and dull, and major chords would be bright and bold). Due to my interest in music and teaching music to others (and teaching in general), I scaled this project down so that it can be a portable software accessible to anyone with access to a MIDI keyboard (or at least a computer that can have notes assigned to the keyboard). I thought I was able to only use p5.js, but I needed to use a text editor and JavaScript/HTML in combination with two libraries. I intended for MIDI Colors to be for kids around the ages of 8-10, but I learned that this program can be helpful to people of all ages.

My first problem was actually an electrical issue, and not a software issue. Before MIDI Colors, I thought I had a faulty MIDI keyboard (My keyboard would be on and its display lights would be on, but my computer wouldn't receive any MIDI information after a few minutes), which is hard to replace at times because of their prices. With the help of Professor Wilson and the MIDI Setup app on iOS, I was able to find out that my MIDI keyboard had no input issues. A very quick fix for this issue was changing my USB to MIDI cord to another cord, and that problem was gone. My first step after that was importing the p5.js and Web MIDI API sources into my html file.

One small challenge of creating MIDI Colors was the lack of information about the Web MIDI API. I spent time on several websites looking up different ways to use the API, but some projects looked similar and had similar notes. But, the lack of information was actually a blessing due to the fact that my options were limitless as long as I was able to combine it with the functions and parts of p5.js. The API was actually somewhat easy to use because the websites that covered it explained intro steps and the setup with great detail. I was able to take the project in any direction due to the two libraries being able to work with each other. Unfortunately, one small roadblock was the Web MIDI API receiving somewhat of an update over the Thanksgiving 2021 weekend. I received about six or seven errors from Google Chrome, and at first I wasn't able to solve the problem; it took me an hour or two to fix the problem.

Near the end of the semester, another challenge that slowed down the process of the project was me not having my specifications fully complete. Some weeks I would add on more specifications for my program. For example, I didn't include the intervals of major and minor chords in my specifications, but I was advised by Professor Wilson to include them because they still count as major chords. Another way that my specifications slowed down the process was because I didn't specify at the beginning whether or not the drawn piano keys would be

highlighted when users pressed their MIDI keys. I was suddenly pressed for time just when I had thought I was almost done with the creation of M.C. Unfortunately, I had to settle for an option of labeling notes with their corresponding MIDI numbers, which would have to be changed due to everyone not having the same amount of MIDI keys (starting from the C key in the third octave up to the C key in the fifth octave). I will fix this problem in further stages of MIDI Colors' development. Lastly, one specification that I listed, but didn't get to complete due to time was the implementation of the sound library. I will eventually learn the Web Audio API, but for now I had to put sound examples in my presentation. Due to me missing a few small specifications, my progress was backwards at time and I wasn't able to fully flesh out MIDI Colors.

A large chunk of the project was the JavaScript/HTML and p5.js coding. Despite the challenges in my project, I was able to visually represent MIDI Colors in a simple, but good way. I was able to implement all of the chord colors and their names easily with the help of p5.js. Also, I was able to display the MIDI numbers of the notes in two octaves with a keyboard under it. Lastly, the images of these chords were easy to implement. I decided to use a sun emoji for the major chords, and a rain cloud emoji for minor chords as a cherry on top. I felt that using those two images could help users connect minor chords with sad songs and dull colors and major chords with positive songs and bright/bold colors. After completing the code

In conclusion, creating MIDI Colors was not an easy task and I learned a few things on the way. MIDI Colors is far from being complete and it needs improvement. But for not having done JavaScript in a while, and using the Web MIDI API for the first time in combination with p5.js, I think I did a fairly good job. When the semester is over, I will continue to work on M.C. in hopes that I can develop it into a fully working program (or a game), and further develop it into a program for a large installation. A possible game idea could be a platformer in which the player progresses through levels and fights enemies by entering notes or chords within a certain amount of time. Due to the fact that a program like MIDI Colors can be useful for any age, I can use the JavaScript code and make it into anything I want. The most important thing I learned from creating MIDI Colors, was to break down each problem that you have one by one and try to tackle things individually day by day at a good pace.