I decide to use the music visualizer template and extend upon it. I'm have been very big of chiptunes and electronic music since my childhood, and an opportunity to work with music was a selling point for this extension. Also, I'm not good at drawing and don't draw much, so I brushed off the first template. As for the data visualization, I'm primarily working with data, and creating data representation is feeling more like a routine task that leaves almost no room for imagination. For me personally, music visualization application provides more in terms of creativity and fun, so I decided to try my best to extend the second template.

I'm very poor as an artist and because of it, I started with a very vague idea about what I want to do. The first idea was simple - just look at some different examples and use them as an inspiration, to create something similar. In our practice extensions, we focused on the ways to create some animations based on the information that we gather from our song. But after putting some thought into it, I decided to add a little bit of difficulty to the project and try and create something akin to the media player: music visualization cant be separated from the music itself after all. I think this way I can practice the skills and knowledge I already acquired and also push myself to learn something more.

Because I already have some idea about how to add new visualizations, I start with the player part. I spent most of my time reading about how to work outside the canvas and testing features I wanted to add one by one. Until now my project is more like a testing range, than a full-fledged application. Because of this fact, I have not used all the advanced coding techniques as of now, but in the process, I hope to implement all of them to create a better application. The biggest challenge for me is actually planning. I started without a clear game plan and worked in small steps: adding user input, adding control buttons, and so on. While my solutions worked by themselves, but when I started to add them all together, I faced so many new problems because of the poor planning. Without architecture, my project looks like a hot mess and surely will collapse on itself if I will continue to stack "blocks" of code atop each other. After the midterm, I plan to go back to the drawing board and start from scratch.

---------

Funny enough, but shortly after I started, I recognized that I don't know the science behind music, so I started by reading about how a computer handles sound. Next, I read more about the p5sound library, to better understand its functionality. I finished the planning phase with the general idea of my application. To me, the finished project must look like a standard media player that allows users to add, listen, and enjoy visualizations of their favorite songs that work in a web browser. As such, I separated my work into 3 parts:

1) Create and add new visualizations to the template project.

There are four main sources for the visualization: the frequency amplitudes, the waveform, volume, and beat. In the p5sound library, there is a lot of built-in functionality, so I can get almost all the information. I still look for a way to get the BPM of the song from the sound file, but it looks like I have to do this part manually, using Web Audio API. A main trick for the visualization is to find a creative way to animate these 4 data sources. This part has supposed to be easy, so I have started with the second part.

2) Create a web-based media player.

I decided from the start that my extension should visualize not only the pre-built track but his own music. So the user input was the first step. After that, I switched to the ways a user can interact with his tracks. I started with simple functions like start and stop, but then moved to the more complicated ones like shuffle or loop modes. I added them one by one but stuck for some time when their simultaneous use caused some unintended results. Now, when I see what I want from the player, and how these features should work - I already see the ways to improve my project and maybe add some additional features, that I like to implement in the final version. I thought this part would be moderately difficult. And because I had little to no knowledge about how to implement these features - I started from this part and implemented almost all features of the normal audio player software. At this point there almost no user interface, but I will fix it soon.

3) Additional features I like to add given enough time.

If I will continue to work on this extension, I like the ability to extract the BPM from the song and maybe use it as a base for some simple rhythm game. This part is the hardest, and more akin to the feature plans.

----

I plan to use a few next weeks to add some bonus features (there are several possibilities, but I still not decided. I can add MIC input mode, add more ways to work with playlist, add track timeline or maybe some other buttons) to the player, work on the visual part and better connect the player to the visualizer part. Also, I am thinking about changing the whole architecture of the player, but not sure about time constraints. I think to finish the player part before the end of January. This part can be tricky, and I intentionally leave more time for this part.

The second part will be based around the visualizer template and I plan to add at least 3 new visualizations (with an average of about 1 visualization a week) and dedicate the final week of February to combining Media player and Visualizator. This part shouldn’t be this hard, because the template already offers a good foundation, and all I need is to add some creative ideas.

The last 2 weeks of March are reserved for the testing and bug fixing of the application, but if everything goes right, I can spend this time to commit to refactoring or add some additional features to my application.

------

[GameDev.net -- Beat Detection Algorithms](http://archive.gamedev.net/archive/reference/programming/features/beatdetection/index.html)    
[Visualizing Music with p5.js](https://therewasaguy.github.io/p5-music-viz/)    
[Beat Detection Using JavaScript and the Web Audio API | Beatport Engineering](http://joesul.li/van/beat-detection-using-web-audio/)   
[Getting Started with Web Audio API - HTML5 Rocks](https://www.html5rocks.com/en/tutorials/webaudio/intro/)   
[Read files in JavaScript](https://web.dev/read-files/)