Ravi Raina

MUMT 301: Project Proposal

Abstract

After the completion of our final class assignment, which utilized Pizzicato.js to implement an online theremin instrument, I became very interested in diving deeper into JavaScript MusicAPI libraries and figuring out what I could build with these technologies. Once I had completed some research into what was offered in the Pizzicatio.js and Tones.js libraries, I came up with the idea to create a minimalist, online playable synthesizer. The synthesizer will consist of a responsive keyboard, a control panel with options to change various attributes and add effects, a drum sequencer that can be put on a loop while the synthesizer is played, and finally a record/play functionality for the local session. Creating this synthesizer will utilize many different technologies and concepts we learned over the course of the semester, including creating a styled website with HTML and CSS, JavaScript for creating the synthesizer functionality and DOM manipulation allowing our website to be interactive, utilization of API's and JavaScript libraries for sound generation and processing, namely Pizzicatio.js and potentially Tones.js. Depending on the timeline, I would also like to implement preset melodies that would be able to be selected above the synthesizer. The user could then overlay their own harmonies using the recording feature, add a drum loop or various effects.

Sub-Goals

A large reason as to why I chose to pick this project, and a JavaScript project in general, is to increase my proficiency in JavaScript on top of demonstrating skills I have learned in the course. By the end of the project, I would like to have reached a point where I am very comfortable with DOM manipulation, JavaScript style and syntax (specifically the different function variations and their use cases), as well as furthering my knowledge of HTML and CSS. Concerning the music aspect, specifically the drum sequencer, my goal is to learn how to design

sounds utilizing the libraries I have noted and their functionality. As neither Pizzicatio.js nor Tones.js offer “drum sounds”, I will have two options when it comes to implementing the sequencer: load in audio files as samples or attempt to create the sounds from scratch using the sounds and effects provided in the libraries. Ideally, I would like to implement them myself, however depending on my timeline this may be subject to change. The final goal of this project is to design a way to allow for recording/playback, at least in the user session, without the use of a dynamic backend or storage. I have brainstormed how to go about this implementation, with my current idea involving storing note information (frequency, synth type and effects, note duration) in an initialized array within our JavaScript file, as well as creating “empty” notes that track the duration of silence between each played note. Additionally, I am still trying to figure out the best way to go about allowing the synthesizer to be polyphonic, as my initial implementation using event listeners only allows the synthesizer to be monophonic, making note playback feel choppy and suboptimal.

Timeline

As of the time of writing, the current state of my project is a playable, monophonic keyboard with the ability to change octaves and a beta version of my recording idea implemented. I am aiming to split the work between the next two weeks, with next week being dedicated to developing the synthesizer’s control panel, figuring out polyphonic playback, and beginning the implementation of the drum sequencer. For week two, I would like to polish up the UI, implement the recording functionality fully, and finish the drum sequencer. However, as stated earlier, if I am struggling with the sound design for the drum sounds I may revert to loading samples into the program using Pizzicatio.js’s sound file loading functionality. This would be the final step in the project. Depending on how well the synthesizer functions, I may continue to develop it into a much more complex web instrument after the course is complete.