Noah Larbalestier

Michael Casey

Music and AI

Final Project Writeup

**Materials/Sources:**

This project was challenging because I used a number of materials unfamiliar to me, and had to sift through lots of open-source code in two languages I had “learned” very recently to attempt to match libraries and operations with the magenta code I was familiar with and had used for Assignment 4. With help from Dartmouth Professor Michael Casey, Dartmouth Music Lecturer Scott Sanchez, and former Student Matt Keating’s final project, I used:

* [Google’s Magenta](https://github.com/magenta/magenta-js/blob/master/music/README.md) software including [MVAE](https://magenta.tensorflow.org/music-vae) and [MidiMe](https://magenta.tensorflow.org/midi-me) models for all generation.
* Music Checkpoints with the autoencoder from GitHub users Edwin Toh and notwaldorf, who have an [online GitHub repository](https://github.com/magenta/magenta-js/blob/master/music/checkpoints/README.md) on which these are hosted and easily accessible.
* My own code to render and process midi data into a form readable by the [VexFlow library](https://github.com/0xfe/vexflow), which was enormously helpful in displaying music notation on the staff when inline with my html script for my web-app. [(VexFlow site w/examples)](https://www.vexflow.com/)
* Former Dartmouth Student [Matt Keating’s MidiMe project code](https://git.dartmouth.edu/music-and-ai/code/-/tree/master/projects23W/MattKeating_MidiMeMusicVAE?ref_type=heads).
* Dartmouth Professor [Michael Casey’s Magenta.js scaffold code for COSC 89.29 Assgt.4](https://git.dartmouth.edu/music-and-ai/code/-/tree/master/MagentaTutorial?ref_type=heads)
* Classical Guitar Instructor/Composer [Aaron Shearer’s free instruction materials](https://www.aaronshearerfoundation.org/beyond-the-open-position/), and the “[Sheet Music Scanner](https://sheetmusicscanner.com/)” IOS app to convert classical music to MIDI for influence files. I used his first study in the second position for all my demos. (.mid in submission zip)//used bc simple

**Self-Evaluation:**

I am very pleased with the outcome of my project. Still, had I not begun experimenting with Google’s Magenta JS in assignment 4 and pushed through the technological challenges I faced, I would never have even thought my work possible. Coming from assignment 4, I didn’t even think my vision was attainable. In using only MusicVAE, I found that outputs just sounded terrible, and I had not come to understand the capabilities of MidiMe when paired with carefully selected input files. My original hopes for this final project were very similar to what I ended up creating—an application that can take in and display classical music for guitar from midi file input, have options for selecting things like octave and in the future, hopefully, positional playing along with tablature. From this display and influence file, I hoped for and accomplished building a system that would train on that piece, and produce/display similar music for classical guitar practice which fit into musical tradition, playing constraints, and my vision for learning how to play the instrument. I am what I consider to be a very conventional intermediate, teen guitarist—I started with zero musical expertise and made headway with the instrument in styles outside of classical tradition. However, I fell in love with classical music through my instruction, and while it was very difficult given my background, I learned to play and read music notation and appreciate the nuances of it over tablature for rhythmic, positional, and efficiency capabilities. Still, during my learning process I would often do things like reference my studies with tablature and faced a lot of frustration with complicated music theory holding me back. My application (for this project) came from my vision for a better way to learn, and I have already found in using it for a few hours that my reading improved greatly. There is limited instruction material for the classical guitar, and very limited intersection with generative AI that is transforming music and art with things like Google’s Magenta.js library, Dall-E 2, and more. I feel very rewarded to be in a place and have some understanding on bridging the gap between these two areas of study, hobby, and practice.

While my assignment 4 was messy and played music rather than displaying it, it allowed me to get comfortable with HTML and JavaScript, which I used for this final output. Expanding on my work with the music variational autoencoder model, and some of Matt Keating’s code, I explored the MidiMe software, gained a better understanding of rendering midi data, and saw how training a model on a single piece of music can make the most renderable, playable, and similar outputs given influence files which are intended for and playable within classical music tradition for guitar. I began messing around with MidiMe in my code scaffold from assignment 4, but was having issues with uploads, and decided to overhaul my previous work and rewrite most of my JavaScript code in line with my HTML. I found my upload button to be working and was able to log midi notes in the console once rendered from the file. (Not play them in a DAW as I had before, since this time I had no intention to) From this point, I had a somewhat functional single file that opened a web app and took in midi input, was able to train using the MVAE and MidiMe models and output new midi notes.

After this step, I knew it was time to put together the pieces I needed to create the output I envisioned. This was stepping into territory which there are very limited resources online for, but with some exploring, I discovered the VexFlow library. This code is extremely helpful and takes in arrays of notes of the representation like ‘c4’ and displays them as music notation. The challenge from here was to figure out when in my pipeline of uploading, rendering, training on, and sampling I would map midi notes to their octaves and note numbers, and how I would display them inside my HTML. After discussing with Professor Casey, I mapped notes and got their octaves easily by using modulo and integer division functions given they are divisible by 12, then concatenating the notes and octaves together. From this, I worked on displaying arrays of notes that VexFlow would accept, and this worked in a window but, as I found, with limited capacity for complexity. I found 16 notes to be the maximum number I could viably display, which led me to my limitations along with my MidiMe tests of more complex .MIDI uploads/influence files.

These limitations are certainly rooted in my fresh exposure to all of these languages and libraries, but this software is also still pretty rough. It handles simple input much, much better than complicated music across octaves or with chords and individual notes mixed into more complicated time signatures. Further, to go forward and display the files I would train on and sample from, I found that single note-at-a-time melodies were vastly easier to display than chordal melodies. To handle these limitations, I searched through my learning material from my Individual Instruction Program at Dartmouth under Lecturer Scott Sanchez and settled on using some of composer/instructor Aaron Shearer’s work, which fit these limiting constraints.

After coming to terms with and handling these limitations, I discovered that even with VexFlow and its integrated “VexTab” capabilities, rendering music or midi then into tablature would be even more difficult. Unfortunately, this is the biggest part of my hopes for the project which escaped. I have many other considerations involving music theory and user constraint selection for the generation, components for teaching, and more, but displaying tablature was something I did think was viable in the timeframe I had. Still, it is my next step in the continued development of this project.

As a whole, my project works extremely well and in my immediate experience, is useful even at its very limited capacity for learning and training in classical style on the guitar. I will continue to work on it and explore the field, as I love the feeling of bridging the gap between two things I enjoy which few people see any connections between. This course has enabled me to create things and especially this final which are more substantial and interesting to me than really any other courses in the past, and it has been extremely valuable for me. I hope to continue work within this intersection and explore possibilities for continued work on my final project.

Thank you, Professor Casey

Note: in my submission zip, I will turn in the shearer.mid file, which I primarily used, some other midi files, my working code, screenshots, and this writeup. I am happy to provide anything else.