## 

Latent Melodies: Interactive Melody Generation with MusicVAE

Nathan Katzman

A Project Submitted to

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Applied Computer Science

School of Computing

April 2025



The signatures of the individuals below indicate that they have read and approved the project of <Nathan Katzman> in partial fulfillment of the requirements for the degree of Master of Science in Applied Computer Science.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<name of project advisor>, Project Advisor Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<name of GPD>, Graduate Program Director Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 <name of unit head>, Unit head Date

# Abstract

Interactive Melody Generation is a web-based application that allows users to effortlessly create and listen to generated melodies. The app uses Magenta and MusicVAE code to transform a melody into a complete musical sequence with simple actions. Designed with the intent with the interface allowing users to experiment on the main page without having to navigate away. This project aims to make music creation more accessible to users of all levels and encourage a more interactive and engaging experience with artificial intelligence in music.

# Introduction

Creating and composing music often involves knowledge of instruments or music theory, which can prevent a beginner or those without training. Even though MusicVAE offers great capability for music generation, they are not necessarily accessible for non-experts due to their technical complexity and lack of user-friendly interfaces. The project addresses the challenge of making AI-powered music generation more interactive and user-friendly. The goal is to design a web-based application that allows the user to easily generate and listen to melodies without understanding machine learning or MusicVAE. The application lets users to type in prompts or selected piano keys to define musical ideas and then use MusicVAE to generate the melody based on the input. The user can interact with the generated music directly from playing, stopping and adjusting the melody within the single, friendly interface. By reducing the complexity of the interactions with MusicVAE and focusing on accessibility, this app lowers the barrier to create musical expression and encourage experimentation, even for users without a musical background.

# Background/Related Work

In this section, describe any related work. Describe and/or reference existing related work/projects. Critique the existing work. Where is it strong? Where is it weak? How does your work relate?

The related work that that relates to the project is MusicVAE which is a section of the overall project known as Magenta. MusicVAE is a machine learning model that lets the user create palettes and use musical scores. The project uses latent space models to take variations within a high-dimensional dataset and convert it into a lower-dimensional dataset to represent how data is used within a melody. The process is used to increase the expression, realism and smoothness of a melody allowing the melody to be more natural and engaging to the user. The current work is strong in making sure that complex music is simplied such that a user who might not be an expert in music can create melodies and music. Also, MusicVAE is able to create take any user input and convert it into a working melody making it useful for musicians. However, MusicVAE is lacking the ability for the user to create music within the same page without having to navigate to other parts of the magenta program. My work relates to this since my project concept was to create a working app which takes the concepts from MusicVAE and places it within the same page such that a user could create a melody prompt to playing the same melody, therefore making the process simpler those who are not experts in music.

# Methods

How I did my project was I read on how Magenta and MusicVAE works and read the pages in order to understand the concepts on how the code works. Then, I downloaded the files so I can access the MusicVAE code, which happened to be in typescript. Next, I implemented code for my application within the typescript from typing the prompt to running the melody. In order to run the code, since I was using VScode, I had to install tsc from the VScode terminal to convert the Typescript to Javascript and then modified the Javascript and added code to an HTML file ensuring that the code works. Finally, I ran the code through a liveserver and my results would be displayed within the HTML. To the right, is a diagram of the App workflow showing how the app functions from the user typing a prompt to stopping a melody. What the project does is that it takes a prompt that can be piano keys(A-G) can be generated through clickable buttons or typed by the user as text. Then the Call\_melody() function is called which processes the input given from the user, prints that the melody has been called from the function and sets the current melody to the user input. The Call\_prompt() function handles the input prompt such that it gets the melody from the prompt generated from the user, and then sets the input to the current melody. The store\_Mel() function takes the user generated melody or sequence and will take the item to store the melody within the melody class. If the function does not encounter a melody, then it will not be able to store it within the melody function. The retrieve\_Mel() function takes the stored melody that was created with the store\_Mel() function and retrieves it by getting the melody from the melody class if it exists and gives the melody to the user. Finally, the user can generate the melody by using various functions. The start\_Mel() function takes from the retrieve\_Mel() function as a new melody. The new Melody will take from the MusicVAE sound, decode into usable sound and then play the melody for the user. The time\_Mel() allows the user to determine how long the melody is going to be played for until it is stopped. Finally, the stop\_Mel() function stops the melody after a speicifc timeframe, unless the user did not choose a melody or the user does not want the melody to end. The technologies that I used included: VSCode, Terminal, MusicVAE, Typescript, Javascript, and HTML. The technologies that I was already familiar with from previous courses were Javascript, VSCode, Terminal and HTML. However, the technologies that I needed to learn during the course of the project included Typescript, and MusicVAE.

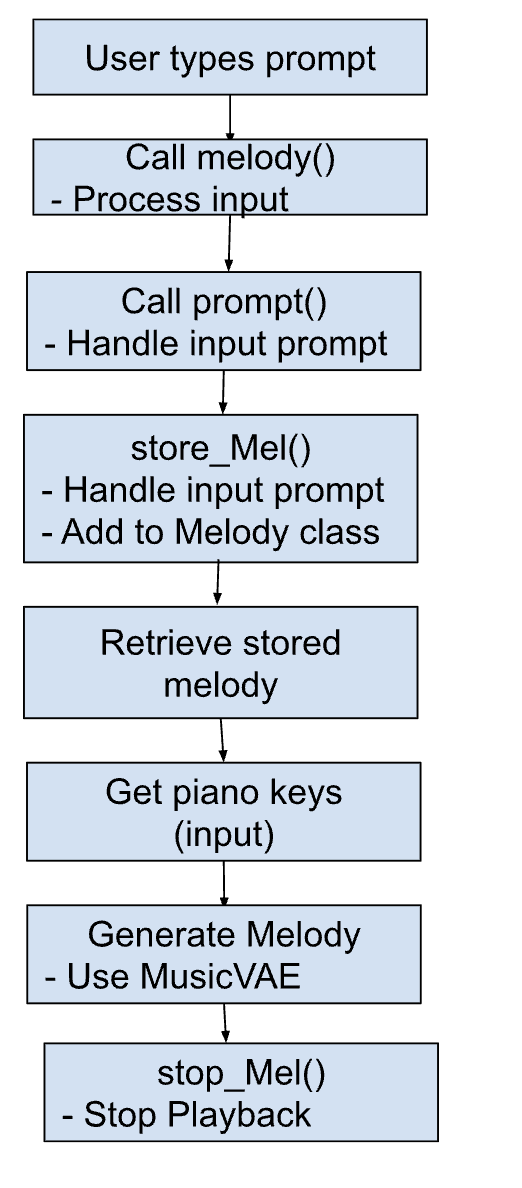
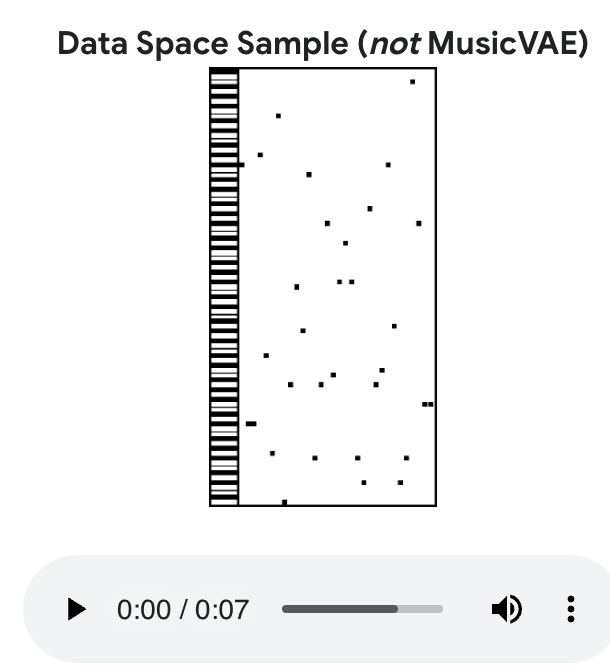


Diagram of the App workflow from typing prompt to stopping the melody.

# Results/Discussion

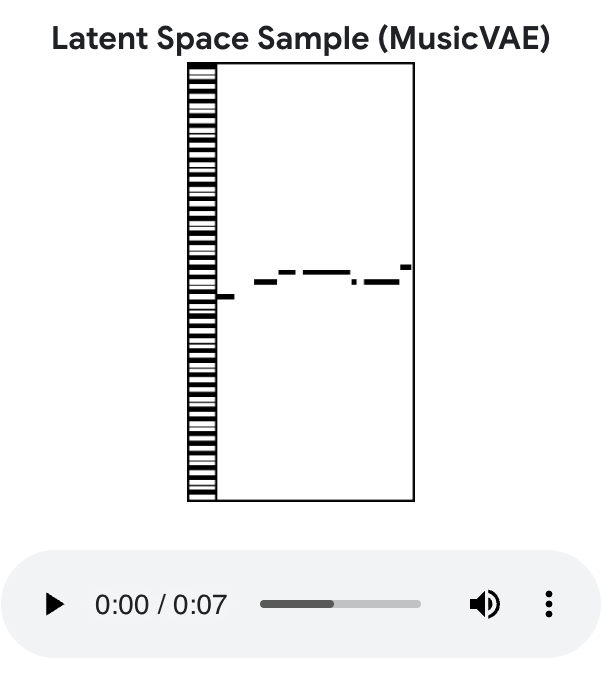
Graphs, plots, etc., and discussion of their implications. Address the adequacy, efficiency, and effectiveness of your project. Again, communicate visually when possible (i.e. graphs/charts). Be honest in the evaluation of your work. Describe any weaknesses and false starts; highlight strengths and accomplishments. These items help with your professional growth and enhance your independent learning ability.

Throughout the project, I was able to get results relating the MusicVAE relating to latency space. The results indicated that when MusicVAE was not being used, the melody appeared to be more scattered, less realistic, expressive, and it did not play the notes that were expected as seen in the figure below.



Example of what the melody played looks like without MusicVAE.

This would make it difficult for an unexperienced musician to follow the melody and understand on a conceptual basis. However, when MusicVAE was being used, the melody was significantly more uniform, more realistic, expressive and it played the notes that one would expect a musician to play. This can be seen more clearly with the figure below.



Example of what the melody played looks like with MusicVAE.

What the result of this research indicates is that MusicVAE’s job is to go into the higher dimensional concepts of a data space. There, the application takes the model, encodes it, and revises the code such that it is similiar to how an actual user would play music. There, the user can decode the sample in latent space and play the melody. The adequacy’s that I faced during the project was that I had to create code through typescript, javascript and HTML to ensure that the code would run, be able to type a prompt and press buttons to ensure that they would eventually use MusicVAE to generate and play the melody. The efficiency of my project was that I was transforming an application that may be too complex for someone who is not a musician into an easy to use application that only uses one page. There, the website can type the prompt all the way to ensuring that the user can generate any melody they want with sound. The effectiveness of my project was that I was able to create the framework such as how will the user type the prompt, where will the buttons be positioned and using code with MusicVAE such that it creates the structure to ensure that the page is structured to make sure that the project works. During the project, I had various weaknesses and false starts. For my false starts, I was initially working with the typescript and HTML. However, when I attempted to run the code, I noticed that running it through terminal would cause an error which made it difficult for me to figure out where there were errors and what needs to be fixed in order to run. What I had to do to actually start on my code was covert the existing Typescript code into Javascript then use a liveserver to run the HTML with the Javascript to ensure that the code worked. Throughout my project, my biggest weakness was that I had difficulty figuring out how to connect the Magenta, Melody class and MusicVAE into the code. To start, I thought that if I added them through imports just like the rest of the app, the code would actually function, then the code would work as intended. However, I noticed that when I attempted to run the results, I would have errors pressing the buttons for my code. When I was revising the code, I had to revise where I was adding the applications to the code and the structure of the code itself. Even though I was able to get the buttons into a useful state, they still did not function as intended because the implementation of connecting the MusicVAE to the rest of the code was not working as I thought and it causes an error since it is unable to be found. This was despite being able to fix my buttons as since the actual Magenta is still not working, the buttons would cause an error as a result. However, the strengths that I had within the project was that I was I refused to give up when something went wrong with my code. For example, when I realized that the Typescript was not working, I converted the code into Javascript and then added a liveserver and HTML to make sure that the code would run. Another strength that I had was that I was able to use Javascript and HTML, terminal and VSCode efficiently. I had experience with these programs since VSCode is my default source for adding and running code. I can use terminal to access and run code, and take the code into github such that the code can be stored. Also, I have experience with Javascript and HTML since I used these programs on projects throughout university and my time at GVSU. My accomplishments throughout the final project were that I learned how to write code in typescript and how to convert it in VSCode such that it can be run. I was able to create a structural framework which includes how to type the prompt, and use buttons to call the melody, the prompt, storing the melody within a class in the application, retrieving the stored melody, getting the piano key input, generating the key, and starting, determining the time of the melody and stopping the melody.

# Conclusions

Throughout my project, I was able to have various accomplishments. I was able to learn how MusicVAE, Magenta and how the melody is altered such that it can be used by the user. I learned how to type code in typescript and how to convert and eventually execute it through VSCode. I was able to type code using the existing framework provided by Magenta and MusicVAE as well as create new code using typescript, javascript and HTML. Despite the final code was not was I expected, I was able to create a framework which is able to type the prompt to ensuring that the melody can start and end and be run from VSCode. For future work, there are still issues that would need to be directed. The steps that are needed to complete the system are: research and then implement code using typescript such that the buttons are connected to the MusicVAE allowing the website to actually play the music, keys and other functions of the code. Make fixes to the tyepscript, javscript and Html such that the code is cleaner, debugged and is works with the webpage. Another issue I am having a problem is figuring out how I can connect the MusicVAE and magenta apps to my code. I tried making them imports, var, and exports. I think it would good to do research and revamp the code once a solution to this issue is found. Ultimately, fixing these issues should make the existing code more functional and allow the website to actually produce music from MusicVAE within a single page. Outside the current issues, I hope to work on more parts of the Magenta code. It would be interesting to apply what I did for MusicVAE and see if it can be done for NSynth, SketchRNN, and PianoGenie. Just like MusicVAE, each of these programs offer unique aspects that can be accessible for any user if the code is implemented. This would allow for the user to not only access melodies, but also sketches, synthesizing audio, and manipulating the keys of a piano to produce a song. The target users of my app will be for anyone who is intrested in music, from inexperienced users who may want to try out the melody keys to an expert who uses the app to practice their melody before playing an instrument.

# Appendices

If applicable

# Bibliography

If applicable

* [“MusicVAE: Creating a Palette for Musical Scores with Machine Learning.” Magenta, 15 Mar. 2018, magenta.tensorflow.org/music-vae.](https://magenta.tensorflow.org/music-vae)
* [Magenta, magenta.tensorflow.org/. Accessed 28 Apr. 2025.](https://magenta.tensorflow.org/)