Wordmuse

A Senior Project
presented to
the Faculty of the Computer Science Department
California Polytechnic State University – San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science

By

John Montague Nelson

December, 2022

Introduction

Since the start of the twenty-first century, artificial intelligence has continued to build up and enhance our world. From autonomous vehicles to self-operated stores, and home devices, they are prevalent in every corner of our life. However, many of these technologies are built to encourage efficiency and not creativity. Thus, for my senior project I have decided to build something that will reignite the spark of my creative side with artificial intelligence.

My app, Wordmuse, combines words with artificial intelligence to create music. Built on Spotify's music API, user-entered words link to previously made songs, extracting tone, confidence levels, octaves, and other attributes. Then, those same attributes will create a song, averaging the extracted attributes from the given keywords. Marketed toward writers, musicians, and other artists, I hope to encourage such groups to participate in the growing era of artificial intelligence.

Project Motivation

Since entering university, I've felt that I have missed out on the opportunity to fully combine my talents in art and computer science. I understand much of an engineering degree is technical and mathematical, but many of my interests lie within the arts. Classes such as Computational Art and Interactive Entertainment Engineering have opened my eyes to the possibilities when combining computational skill with creativity, and since then I've been genuinely excited to do something artistic for my senior project.

Another motive for creating this music application is that it is highly tangible. Often, programs, research, and other technical projects are difficult to showcase to my friends and family in other fields. Foreign programming vernacular and tools can be difficult to grasp, while music is a universally known language. Thus, sharing a procedurally generated song is an excellent final project for my time at university, using many of my skills as a programmer to build something others can hear, feel, and understand.

Why this Project?

As a frequent guitarist and writer, I thought it would be neat to create something that nets the two together. So, I decided to create a converter from words to a music piece through music theory and artificial intelligence. And when mentioning my idea to friends and colleagues, many are confused and believe there is a disconnect between music and code. They argue that music is expressive and unquantifiable while code is mathematical and logical. However, after playing guitar and producing music, I understand there is a mathematical edge to the arts while there is a certain music/creativeness to code. Thus, I am motivated to share and demonstrate this correlation with others.

Goal of my Project

The main goal for my project is to create an application that can connect writers and musicians to the digital world and vice versa. By sharing my app with such people, I can share how code and art are connected, showcasing the possibilities and feats they can achieve when combined. Another goal of my project is to build an album of my best generated songs. By sharing my album of songs with the world, I will be able to reach a farther audience and encourage others to explore the arts with computer science.

Success of my project is measured by the accuracy and technical achievements of my final product. After my project is finished, I will do survey testing by asking participants to guess which word arrays match with each piece of music. Doing this will allow me to tweak my algorithm, giving priority to certain music attributes.

And, because I created a fully functional app, I will be able to attract more users, thus fulfilling the goal of reaching artists, writers, and other users.

Related Work

One related work to mine is AI Art Maker by Hotpot Ai. AI Art Maker takes in several keywords and generates an image based on the words provided. The results are enthralling, as the program makes completely unrelated objects fuse in a unique fashion. Below are some examples of images people have created using Hotpot's tool.

"Seagull Engineer"

"Elon Musk Spider"

The works above were recent creations by users of the app, as they combined completely unrelated ideas such as seagull and engineer to create something odd, yet surprisingly beautiful My work differs, as it is geared toward words and music rather than images. Thus, my product is reaching a different audience, and can specifically focus on writers and musicians. Hotpot's work was what inspired me, as it is very similar in technology but converts keywords to images, rather than keywords to music.

Design

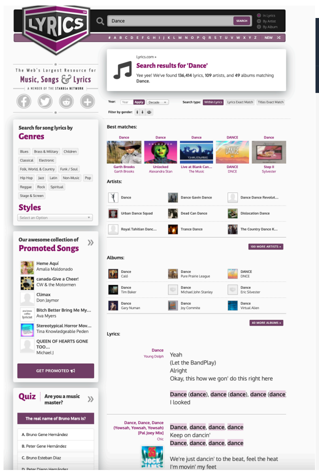
My product is a program that takes in user keywords and creates a piece of music based on the words provided. The main technologies I used were Spotify’s music api for extracting music attributes, BeautifulSoup for website parsing, and GarageBand for outputting and exporting music. Also, by having my full project publicly available on GitHub, I can share it with anyone around the world through a link.

The principles that guided my implementation were usability, creativity, transparency. Making my product easy to use will encourage more people to share it, creating a community around it. Likewise, I want my product to breathe creativity and showcase some of the works people have made. Like Hotpot AI, I included a recent works page where users can display their creations and songs. Finally, having my product transparent by sharing source code will encourage other artists and engineers to explore the arts with computer science.

Implementation

One of the most difficult questions I had to ask myself was what elements of music would determine my model? While seemingly trivial, when I thought more about it, I realized that music is expressive and has many different attributes associated with it. Elements such as vocals, rhythm, beat, melody, instruments, keys, and octaves were all debated, but I settled with confidence, beats-per-minute, octaves, and tempo. Thus, I was able to guarantee my music synchronized well and fused well. What was even more difficult was to determine what attributes took priority over others. While testing, I found that having a on-two-one correlation with the music attributes extracted would result in awkward music notes, weird tempos, and other results that felt non-musical. So, I did some user interviewing to find out what people prioritized in music creation.

To find the top five associated songs for a given keyword, I used Beautiful Soup in conjunction with lyrics.com to scrape the most relevant songs associated with each user input. Next, I took those five songs and used an html parser with musicnotes.com to derive the associated keys, instruments, and more from the given songs.


LYRICS
Dance
SEARCH
In Lyrics
By Artist
By Album
# A B C D E F G H I J K L M N O P Q R S T U V W X Y Z NEW
The Web's Largest Resource for
Music, Songs & Lyrics
A MEMBER OF THE STANDS4 NETWORK

Lyrics.com »

Search results for 'Dance'

Yee yee! We've found **136,414** lyrics, **109** artists, and **49** albums matching Dance.

Year: Year Apply Decade Search type: Within Lyrics Lyrics Exact Match Titles Exact Match

Filter by gender:

Search for song lyrics by **Genres**

Blues, Brass & Military, Children, Classical, Electronic, Folk, World, & Country, Funk / Soul, Hip Hop, Jazz, Latin, Non-Music, Pop, Reggae, Rock, Spiritual, Stage & Screen

Styles Select an Option

Best matches:

Dance	Dance	Dance	DANCE	Dance
Garth Brooks	Unlocked	Live at Blank Can...	DANCE	Step II
Garth Brooks	Alexandra Stan	The Music	DNCE	Sylvester

Artists:

Dance	Dance Gavin Dance	Dance Dance Revolut...
Urban Dance Squad	Dead Can Dance	Dislocation Dance
Royal Tahitian Danc...	Trance Dance	The Country Dance K...

100 MORE ARTISTS »

Albums:

Dance Caló	Dance Pure Prairie League	DANCE DNCE
Dance Tim Baker	Dance Michael John Stanley	Dance Eric Silvester
Dance Gary Numan	Dance Joy Commite	Dance Virtual Alien

40 MORE ALBUMS »

Our awesome collection of **Promoted Songs**

Heme Aquí – Amalia Maldonado
canada-Give a Cheer! – CW & the Motormen
Climax – Don Jaymor
Bitch Better Bring Me My... – Ava Myers
Stereotypical Horror Mov... – Tina Knowledgeable Peden
QUEEN OF HEARTS GONE TOO... – Michael J

GET PROMOTED

Quiz Are you a music master?

The real name of Bruno Mars is?

A. Bruno Gene Hernández
B. Peter Gene Hernández
C. Bruno Esteban Diaz
D. Peter Diego Hernández

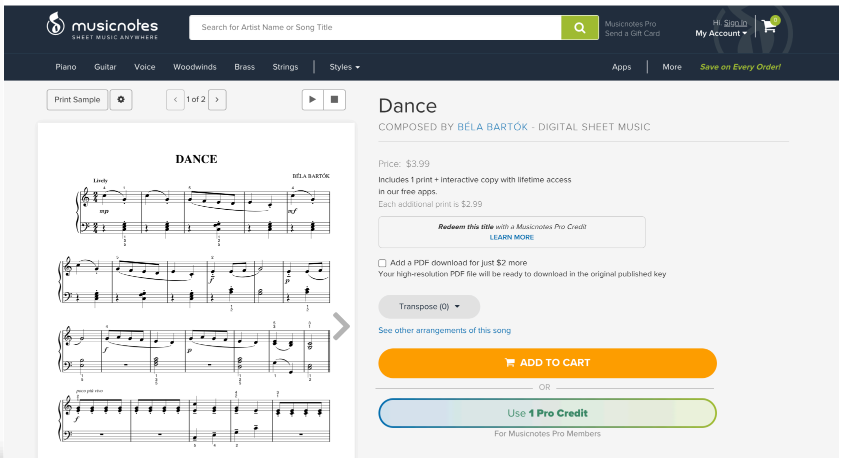
Lyrics:

Dance – Young Dolph
Yeah
(Let the BandPlay)
Alright
Okay, this how we gon' do this right here

Dance (dance), dance (dance), dance (dance
I looked

Dance, Dance, Dance (Yowsah, Yowsah, Yowsah) [Pal Joey Mix] – Chic
Dance, dance, dance, dance
Keep on dancin'
Dance, dance, dance, dance

We're just dancin' to the beat, feel the heat
I'm movin' my feet



Dance

COMPOSED BY BÉLA BARTÓK - DIGITAL SHEET MUSIC

Price: $3.99

Includes 1 print + interactive copy with lifetime access in our free apps.

Each additional print is $2.99

Redeem this title with a Musicnotes Pro Credit

LEARN MORE

Add a PDF download for just $2 more

Your high-resolution PDF file will be ready to download in the original published key

Transpose (0)

See other arrangements of this song

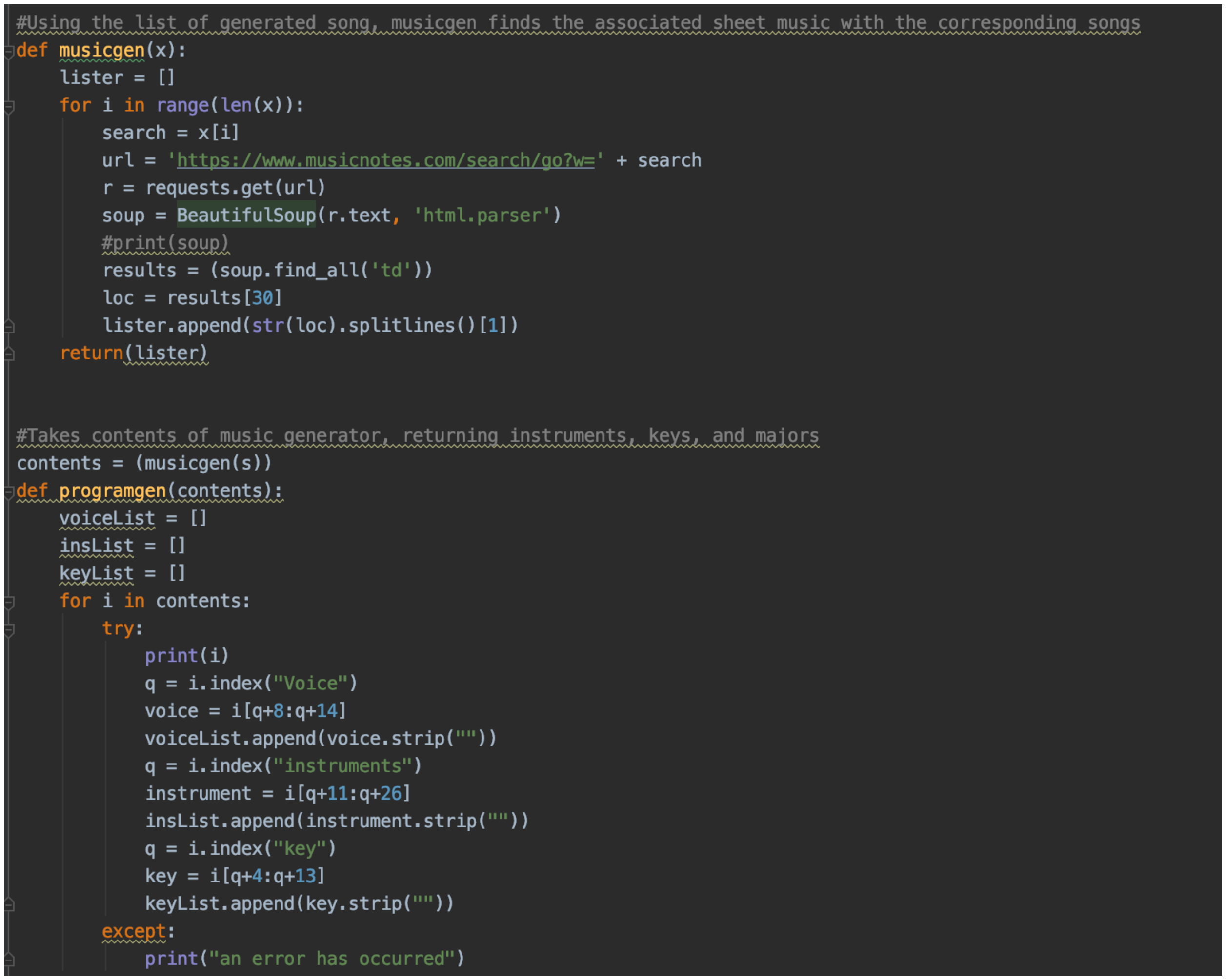
ADD TO CART

OR

Use **1 Pro Credit**

For Musicnotes Pro Members

This method ensured that I was able to grasp the greatest number of songs and filter for my desired result. Both lyrics.com and musicnotes.com have a vast dataset of music sheets, lyrics, and information about both, ensuring I would not be short of data. Likewise, using these websites with Beautiful Soup's html parser was a greater alternative than calling from a database or making API calls, as my operating costs are essentially zero. Below is a piece of code used to parse different songs based on keywords

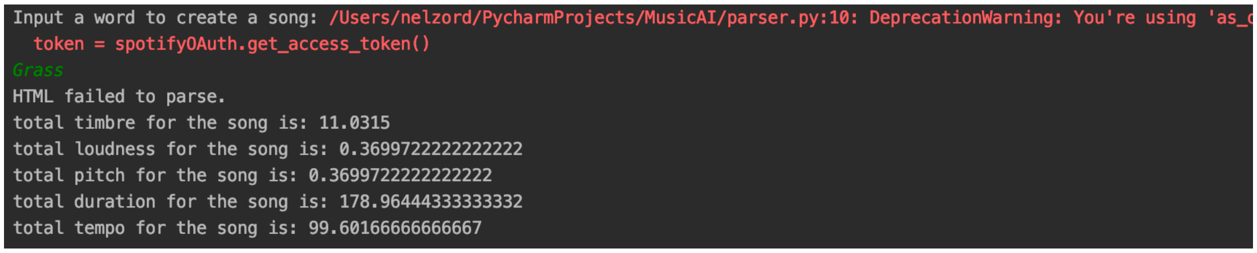


```
#Using the list of generated song, musicgen finds the associated sheet music with the corresponding songs
def musicgen(x):
    lister = []
    for i in range(len(x)):
        search = x[i]
        url = 'https://www.musicnotes.com/search/go?w=' + search
        r = requests.get(url)
        soup = BeautifulSoup(r.text, 'html.parser')
        #print(soup)
        results = (soup.find_all('td'))
        loc = results[30]
        lister.append(str(loc).splitlines()[1])
    return(lister)


#Takes contents of music generator, returning instruments, keys, and majors
contents = (musicgen(s))
def programgen(contents):
    voiceList = []
    insList = []
    keyList = []
    for i in contents:
        try:
            print(i)
            q = i.index("Voice")
            voice = i[q+8:q+14]
            voiceList.append(voice.strip(""))
            q = i.index("instruments")
            instrument = i[q+11:q+26]
            insList.append(instrument.strip(""))
            q = i.index("key")
            key = i[q+4:q+13]
            keyList.append(key.strip(""))
        except:
            print("an error has occurred")
```

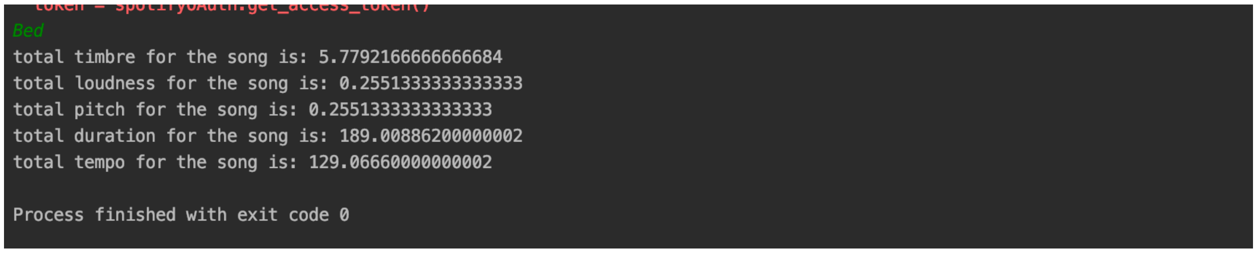
After scraping all the songs with the associated keywords, I determined what associated values would be connected to each word. To do this, I used Spotify’s music api, called the .search() function for each extracted song, and then called .audioAnalysis() for each result. This returned a Json string which contained all the desired attributes. Then, I averaged the results of all the keywords, generating my final attribute array. Below are some generated values created based on user input.

‘Grass’



```
Input a word to create a song: /Users/nelzord/PycharmProjects/MusicAI/parser.py:10: DeprecationWarning: You're using 'as_
  token = spotifyOAuth.get_access_token()
Grass
HTML failed to parse.
total timbre for the song is: 11.0315
total loudness for the song is: 0.3699722222222222
total pitch for the song is: 0.3699722222222222
total duration for the song is: 178.96444333333332
total tempo for the song is: 99.60166666666667
```

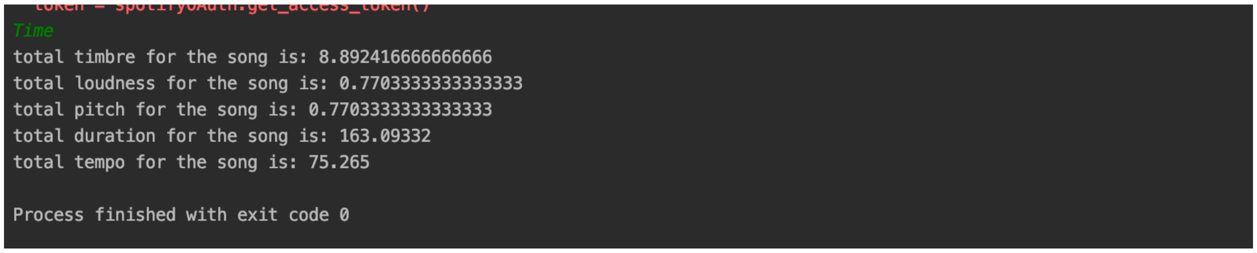
‘Bed’



```
  token = spotifyOAuth.get_access_token()
Bed
total timbre for the song is: 5.7792166666666684
total loudness for the song is: 0.2551333333333333
total pitch for the song is: 0.2551333333333333
total duration for the song is: 189.00886200000002
total tempo for the song is: 129.06660000000002

Process finished with exit code 0
```

‘Time’



```
  token = spotifyOAuth.get_access_token()
Time
total timbre for the song is: 8.892416666666666
total loudness for the song is: 0.7703333333333333
total pitch for the song is: 0.7703333333333333
total duration for the song is: 163.09332
total tempo for the song is: 75.265

Process finished with exit code 0
```

Analysis and Verification

Because music is difficult to calculable like a formula or equation, I conducted user testing with friends and other colleagues to understand if the model generated ‘good’ music. To do this, I gave users a quiz of five songs and five string arrays, asking the participants to guess which songs matched with each keywords string. Thus, users will be unbiased to ‘agree’ or ‘disagree’ whether a keyword array matches with a given song.

User Interviewing

For my first set of user interviewing, I asked colleagues what values of a song they felt was imperative for a song or piece of music. Most mentioned that basic music theory, song structure, and melody was crucial. Thus, I decided to focus on amplifying a song rather than creating a completely new song. This would allow the attributes to be more prevalent, and users could better listen to their keywords, being able to base it off a well-known song such as happy birthday.

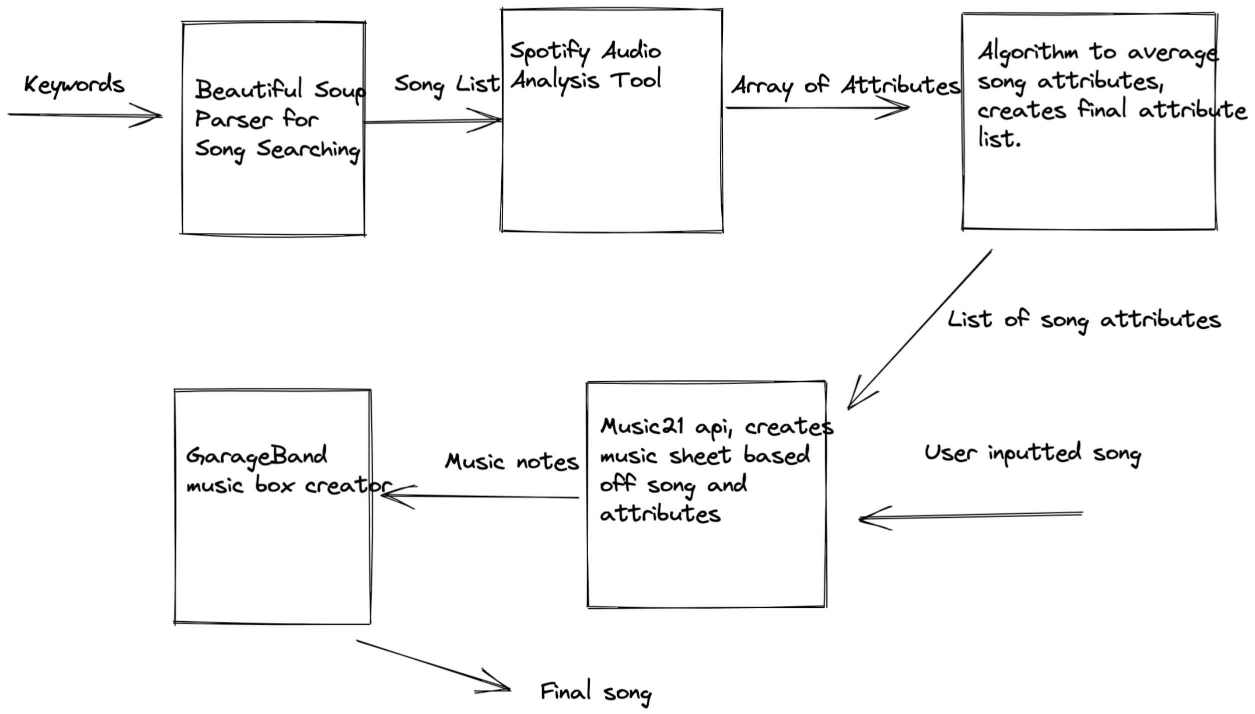
For my user testing, I decided to provide surveyors with five string arrays and five songs created by my algorithm. Specifically, I chose keywords that differed heavily, making it easier for users to denote differences from words. For the string arrays, I chose 'happy grass', 'sad blue', 'haunted hotel', 'dance jump', and 'love happy'. After interviewing five separate users, I found that users were on average able to identify three of the five songs or 60%. I thought of this as a success, as it was much higher than users randomly guessing. This also showcased that user entered keywords had a somewhat accurate output, and words such as 'happy' made upbeat music while words such as 'haunted' made slower and dirge-like music.

Future Work

One future implementation I look forward to in my project is expanding the list of words users can input. When testing my program, I found that many words were too complex or were not prevalent in enough songs. A solution to this could be to build a graph of similar words using a linguistics library, thus substituting complex words in with more common words. This will enhance the usability of my program, as it will be less prone to errors.

Another future implementation I am looking forward to is implementing computer vision with my current project. Because my program takes in an array of strings, there are many possibilities that can generate these strings. For example, I could use OpenCV to detect the most common objects from a photo and pipe those inputs into my program.

While computer vision is just one example, its use case showcases the modularity of my program. Hypothetically, anything that generates a list of keywords could be appended to the front of my program, increasing the complexity of my algorithm. This was done intentionally, as my program is loosely coupled, allowing for certain components to be interchanged easily. Below is a UML diagram depicting the structure of my program.


Keywords
Beautiful Soup Parser for Song Searching
Song List
Spotify Audio Analysis Tool
Array of Attributes
Algorithm to average song attributes, creates final attribute list.
List of song attributes
Music21 api, creates music sheet based off song and attributes
User inputted song
Music notes
GarageBand music box creator
Final song

As seen above, any part of my program can be taken out and replaced with a similar component. For example, Spotify API can be interchanged with another music analysis API. Likewise, a keyword generator could be attached before the first arrow, denoting how certain keywords are generated.

Conclusion

All in all, I felt my final program was a success, and I learned much about music theory and song composition along the way. As showcased in my user testing, on average, users could correctly identify keywords to songs among a set of word arrays and songs. Likewise, my final solution is present on GitHub, and can be tested by users around the world.

However, what was most rewarding about my senior project experience was how I achieved my original goal of fusing the disciplines of machine learning and music together. Throughout my experience of undergrad, I often felt I could not express my artistic talents with that of computer science. Many tasks in computer science are often logical and mathematical, having a concrete solution. On the off hand, music is expressive and there is never a 'correct' answer. Thus, mixing these two concepts was a rewarding resolution for my undergrad, being logical, while also being expressive.

GitHub Repository: https://github.com/Nelzord/music-ai

Works Cited

AI, Hotpot. “Ai Art Maker: Turn Text to Art.” *Hotpot Design*, 12 Apr. 2022, https://hotpot.ai/.

AI, Hotpot. “Ai Art - Seagull Engineer.” *Hotpot Design*, 10 Apr. 2022, https://hotpot.ai/s/art-maker/FlJht/lytygoENbjaiglRsqO27HD2X83vf?share=1&title=seagull+engineer.

AI, Hotpot. “Ai Art – Elon Musk Spirder.” *Hotpot Design*, 1 Dec. 2022, https://hotpot.ai/s/art-generator/8-28l3VlKOPbHi0Ky?share=1&title=Elon%20musk%20spider.

David Osborne, Steinway Artist and "Pianist to the Presidents", et al. “Sheet Music Downloads at Musicnotes.com.” *Musicnotes.com*, https://www.musicnotes.com/.

“Welcome to Lyrics.com.” *Lyrics.com*, https://www.lyrics.com/.