Lesson #5: Various Matters Pertaining to Musical Markov Processing

What's it all about?

This lesson involves listening to a couple of very different short melodies, with the intent of discerning salient differences between the two. It briefly discusses musical knowledge representation, and focuses on a number of example representations. It then turns to the question of how you might work with three of these representations in a productive manner.

Exercise #4: Listening to two tonal melodies

Please give a listen to each of two melodic fragments, Beehoven's "Ode to Joy", and Turk's "March", with an ear towards identifying the most salient differences between the two. In fact, please listen to each of the melodic fragments two or three times. You will find simple renderings of each on the course page:
`http://www.cs.oswego.edu/~blue/course_pages/2022/Spring/Cog356/`

After listening, please identify what you believe to be the **two** most salient differences between the two melodies. That is, try to identify the two most significant characteristics (features) in which the melodic fragments differ. Then, please write down your two thoughts on the matter (difference 1 and difference 2), each in just a short sentence.

JFugue

JFugue is an open-source music library that allows you to play and compose music using the Java programming language. It uses a representation for musical instructions known as "Staccato", and it has a simple mechanism for processing strings of these musical instructions. It is a MIDI based library, so you can, among other things, write a Staccato string to a MIDI file.

The JFugue notation

Pitch classes are represented as follows:

```
C D E F G A B Cb Db Eb Fb Gb Ab Bb C# D# E# F# G# A# B#
```

Pitch classes can be mapped onto pitches by adding a register indicator, a number between 0 and 10. If no register is explicitly indicated, the fifth register is implicitly assumed. (Thus, for example, `C` means `C5`.) The following pitches are among the more commonly used:

```
C4 D4 E4 F4 G4 A4 B4 Cb4 Db4 Eb4 Fb4 Gb4 Ab4 Bb4 C#4 D#4 E#4 F#4 G#4 A#4 B#4
C5 D5 E5 F5 G5 A5 B5 Cb5 Db5 Eb5 Fb5 Gb5 Ab5 Bb5 C#5 D#5 E#5 F#5 G#5 A#5 B#5
C6 D6 E6 F6 G6 A6 B6 Cb6 Db6 Eb6 Fb6 Gb6 Ab6 Bb6 C#6 D#6 E#6 F#6 G#6 A#6 B#6
```

These pitch identifiers constitute basic JFugue note playing instructions. As is, the duration of a note is assumed to be a quarter note. That said, the duration of the note can be represented by appending any number of occurrences of the following letters to the basic pitch instructions: `W` (whole note), `H` (half note), `Q` (quarter note), `I` (eighth note), `S` (sixteenth note). Thus, here are some examples of notes of different pitches and durations:

```
C5W C5H C5Q C5I C5S C5HQ Ab4W Ab4H Ab4Q Ab4I Ab4S Ab4HQ
```

What is JFugue good for?

JFugue is serves as a music knowledge representation which is fairly intuitive for the human consumer, yet straightforward for the (Java) machine to process. How can it be processed my machine? Here are two very different ways:

1. You can write Java programs which import the JFugue API.
2. You can feed JFugue instructions to a little Java program called SimplePlayer.

A tutorial video showing how to feed JFugue instructions to the SimplePlayer program will soon be presented.

Some JFugue resources

The following link will take you to the JFugue manual, in case you would like to learn more about the JFugue music knowledge representation.

`https://usermanual.wiki/Document/The20Complete20Guide20to20JFugue2C20Second20Edition2C20v200.723471053/help`

This link will take you to the "origin story" of JFugue (a light little story), written by its creator:

`https://medium.com/@dmkoelle/origin-story-the-creation-of-jfugue-e361fa358702`

Midi

Midi, the musical instrument digital interface, is the standard music knowledge representation for machine to machine communication. The midi representation plays the role of interlingua with respect to translating from one music knowledge representation to another. For example, we will see midi as the language through which we can convert strings of JFugue instructions to MP3 files.

In the aforementioned tutorial video, we will see how to convert a string of JFugue instructions to a midi file, which is the first step in a two step process of converting JFugue to MP3.

MP3

MP3 is a popular file format for music files. Most browsers can play MP3 files, as can any number of music playing programs.

If you have a midi file, you will often want to convert the file to MP3. For that, you need a midi to MP3 converter. An online midi to MP3 converter is referenced from the course web page.

JFugue → Midi → MP3

Unfortunately, the tutorial video that I created a while ago to convert a sequence of JFugue instructions to an MP3 file is now out of date, simply because the midi to MP3 converter that I used at the time is now among the missing. Still, I think that the video will be useful to you. Just remember to use the midi to MP3 converter that is now referenced from the course web site, rather than the now defunct SolMiRe. Here is the tutorial:

`https://www.cs.oswego.edu/~blue/course_pages/2022/Spring/Cog356/video_tutorials/JFugue2MIDI2MP3.mp4`

A perspective on Markov music

The two pieces featured in your "Markov Analysis and Algorithmic Composition" assignment require quite a bit of work (hand computation), for what they were worth in terms of musical sound. But the point of the assignment isn't to generate music of any substance. Rather, it is to afford you a tangible opportunity to engage in suggestive processing on both sides of the Markov model, both analysis and generation.

In order to enrich the experience, and remind ourselves of the fact that these computations are generally done by machine, I thought we might listen to a short piece generated by a Markov process as we individually watch an encapsulating video. The piece makes use of a Bach piece as corpus, that is, as material to feed to the Markov process. I chose the video mostly because the line by line computer output, which suggests that features beyond just pitch and duration are being modeled, is a constant reminder of the machine which serves as the computational infrastructure for generating and rendering the music.

Please spend just 2 minutes with the video at the following location, doing your best, as you listen, to sort the good from the bad, the interesting from the ignorable, or any other dichotomous qualities that come to mind.

`https://www.youtube.com/watch?v=lIOiAKOx4vA`