Journal Report 1
9/02/19-9/09/19
Irina Lee
Computer Systems Research Lab
Period 1, White

Daily Log

Monday September 2

Labor Day

Tuesday September 3

Using Sibelius, I created 3 MIDI test case files and successfully converted them to text files using the online converter Flash Music Games. Below is an example of one such text file, containing information for four quarter notes in the order C5, D5, E5, F5.

0 Meta TrkName " " 0 On ch=1 n=60 v=76 256 Off ch=1 n=60 v=0 256 On ch=1 n=62 v=77 512 Off ch=1 n=62 v=0 512 On ch=1 n=64 v=78 768 Off ch=1 n=64 v=0 768 On ch=1 n=65 v=78 1024 Off ch=1 n=65 v=0 1025 Meta TrkEnd TrkEnd

Every two lines represents one note played. After researching the components of a MIDI file, I wrote script that could parse note values from the text file. The main question right now is how I should store the information. I originally considered using a dictionary with note information as keys and fingering as values but realized that dictionary keys are stored as sets and thus cannot contain duplicates. My second thought was to use an array of tuples, but tuples are immutable and modifying them can become a hassle if it needs to be done frequently as it will in this project.

Thursday September 5

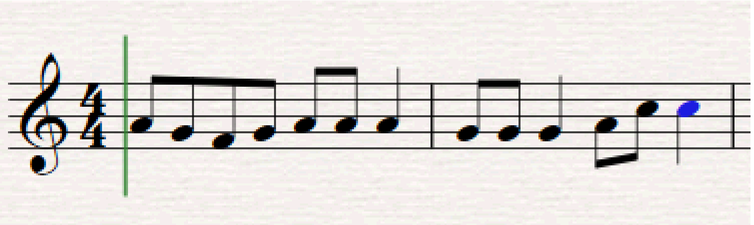
I decided to use an array of arrays, with the outer array representing the entire piece and each inner array representing either a note or a rest. Each of the inner arrays has a length of three, with the first element being the MIDI time, the second element being the MIDI note value (e.g. 60), and the third being the fingering (blank right now). In MIDI time, 256 represents one beat (or the duration of a quarter note) in 4/4 time. One measure would be 256 * 4 = 1024 MIDI time values. I also made an alternate "pretty" array, which uses more traditional methods of representing note values (e.g. C5).

Timeline

Date	Goal	Met
–	–	–
August 29th	Download Sibelius and create MIDI file test cases	Yes, I created several major scales as test cases.
September 5th	Convert MIDI to text file and write script to extract all necessary information	Yes, I created an array of arrays, with the larger outer array representing the entire piece and each inner array representing a note or rest.
September 12th	Write script to finger C, G, D, A, E, major scales	
September 19th	Write script to finger all 12 major scales	

Reflection

Given any piece of music written in the treble converted to a MIDI text file, I can extract note numbers and note durations, as well as process where rests occur in the piece. Below is music example taken from ”Mary Had A Little Lamb” that I inputted into Sibelius.



First Array: [[128, 67, None], [128, 65, None], [128, 67, None], [128, 69, None], [128, 69, None], [256, 69, None], [128, 67, None], [128, 67, None], [256, 67, None], [128, 69, None], [128, 72, None], [256, 72, None]]

”Pretty” Array: [[128, ’A5’, None], [128, ’G5’, None], [128, ’F5’, None], [128, ’G5’, None], [128, ’A5’, None], [128, ’A5’, None], [256, ’A5’, None], [128, ’G5’, None], [128, ’G5’, None], [256, ’G5’, None], [128, ’A5’, None], [128, ’C6’, None], [256, ’C6’, None]]

Next week, I will start by writing script that will be able to finger pieces that do not require crossovers, like pentatonic scales and the musical excerpt shown above. I’m thinking about writing a method that will be able to calculate or estimate the number of crossovers required in a piece before it even starts putting in fingering.