Journal Report 17
02/10/2020-02/17/2020
Irina Lee
Computer Systems Research Lab
Period 1, White

Daily Log

Monday February 10th

The key to implementing more complex crossover scenarios is to have a multi-round process, but my ability to do that is hindered by the fact that my current code tries to do everything in a single round. While it can handle a number of cases, a single round isn't enough to catch everything; at the same time, the way my code is structured right now isn't open to change. It's hard to find an entry point for decision-making about crossovers. I've decided to re-vamp how my code places fingering. It will keep all the core decision-making the same and still retain most of the helper methods (methods related to finding the motif won't be touched). The purpose of this change is to make my code more open to a multi-round system of placing fingering.

Tuesday Tuesday February 11th

The new version now places fingering in two rounds: the first identifies sections where lateral shifts (where the hand has to move up or down the keyboard) occur and chunks them appropriately. After chunking, it places fingering in using helper methods established previously. With the lateral shifts fingered, every note that has not yet been fingered is assumed to be a part of a static section, and it fills that in accordingly. Right now, I'm pretty much back to where I was before, but with more flexible code. With a few glitches, my current code can finger everything it did before.

Thursday February 13th

A new way of thinking about dealing with crossovers in succession would be splitting up sections into blocks of 2, 3, or 4. Crossing over at the index finger = 2, middle finger = 3, ring finger = 4. Essentially, I try to "sum" these blocks of 2,3,4 to a total. I've started pursuing this idea today, and I'm going to expand on it next week. Today, I finished coding for one crossover, but it cannot yet deal with crossovers in succession. I also put in code that identifies whether crossovers going up and down, and compensates accordingly.

Timeline

Date	Goal	Met
January 31st	Transfer my right hand fingering algorithm to my left hand	Successfully transferred right hand to left hand
February 7th	Algorithm can handle crossovers in succession	I focused on fingering chords in succession and completing pieces with little to no motif.
February 14th	Be able to handle more complex crossover scenarios	Can handle crossovers in Czerny's Exercise No. 1
February 21st	Be able to handle crossovers in succession (e.g. 2-octave scales)	–
January 28th	Integrate new version of "place fingering" method with the motif algorithm to generate fingering variations	–

Reflection

This week, I changed the structure of my code in preparation for more advanced crossover scenarios. Below is an excerpt of my code. Previously, all the methods shown below were thrown into a single method. Now, they've been split up into multiple rounds of scanning. The second image is how my code fingered Czerny's Exercise No. 1.

```
rmusic, crossovers = place_lateral_shifts(rmusic_arr)
print(rmusic)

static_sections = identify_static(rmusic)
print(static_sections)

rmusic = fill_static(rmusic, static_sections)
print(rmusic)

rmusic = fill_crossovers(rmusic, crossovers)
print(rmusic)

rmusic = fill_between_fives(rmusic)
print(rmusic)

export_file("Exercise No. 1", "Czerny", rmusic, [], [], [])
```