Task 7 - Analyze User-Friendliness of Selection

In this task, I tested the user-friendliness of the user ranking process for different music samples. I tested with different selection sizes and beat totals to determine trends in time elapsed and rankings. This document walks through the methodology and results of this analysis. The concluding section of the document is a proposal for what work needs to be done based on this analysis.

Methods

I worked with the *demo–interactive-selection* method to test the user-centric fitness ranking interface using different selection sizes and beat totals. Selection sizes of 3 and 4 with beat totals of 26, 16, and 8 were tested. This made for a total of 18 trials. Time elapsed, melody ranks, and bassline ranks were all recorded.

```
( defmethod demo--interactive-selection ()
( setf popu ( initial-population ) )
( setf selection ( select-individuals popu ) )
( interactive-selection selection )
```

Results

The following spreadsheet shows the results of the aforementioned 18 trials.

	А	В	С	D	E
1	Selection Size	Melodies Ranks	Bassline Ranks	Time	Beat Total
2	3	6,6,7	6,3,5	1:23	26
3	3	8,4,8	4,3,4	2:12	26
4	3	5,4,5	3,6,3	1:56	26
5	4	4,8,5,4	2,6,2,6	2:23	26
6	4	4,7,7,4	3,5,2,2	2:09	26
7	4	4,4,4,5	2,2,2,5	2:40	26
8	3	6,3,4	4,2,6	0:58	16
9	3	3,6,6	3,5,5	1:18	16
10	3	4,4,6	4,1,2	1:27	16
11	4	6,3,1,6	3,3,5,3	1:38	16
12	4	4,4,6,2	2,1,1,2	1:19	16
13	4	8,5,7,7	1,5,2,8	1:54	16
14	3	3,4,5	4,2,4	0:50	8
15	3	5,4,5	4,2,3	0:45	8
16	3	4,2,5	3,2,2	0:42	8
17	4	4,5,5,4	4,2,4,1	1:04	8
18	4	8,4,3,5	1,2,1,4	2:20	8
19	4	3,6,6,5	2,6,3,2	1:06	8

Overall, the trials with the selection size of 3 coupled with a beat total of 8 had the shortest time elapsed, while the trials with a selection size of 4 coupled with a beat total of 26 had the highest time elapsed. Outlier times can be attributed to EasyABC; EasyABC can be quite slow/glitchy especially as the length of the composition increases.

In my opinion, the longer compositions produced more interesting results. Perhaps, 20 or 24 beats may be ideal for the beat total since those result in five or six measures, respectively. Sticking to one of those numbers may likewise minimize EasyABC's lag in loading compositions for playback. As for selection size, 3 or 4 seem to both be reasonable options. A selection size of 3 may be more ideal for longer compositions and larger numbers of generations.

Additionally, the bassline performed worse on average compared to the melody pairings. I think this is due to the fact that basslines are completely randomized. Perhaps, implementing a function to produce stepwise basslines may add some more interesting results in the initial population.

Lastly, I think playing 3 melodies at once produces confusing results that are too convoluted and detract from the goal of the project: exploring algorithmic composition that includes the user in the music results it creates. I tried experimenting with different MIDI instruments to differentiate the melodies, but that produced results that drowned out the other melodies and/or caused EasyABC to glitch.

What's Next

I am going to pivot from using three melodies to two melodies. The bassline generation will not get discarded entirely. Instead, it will be one of the options for generating melody2. I think it could be interesting to keep melody1 completely randomized, while eliminating the randomization of melody2 to see how that affects the rankings. There will still be two rankings. Instead of ranking melody1 and melody2 as a pairing, they will be ranked separately. This should create less convoluted results, while still incorporating already coded elements from the project.

I would also like to add some to the constraint system to produce more musically interesting starting samples in the initial population. I found that it was difficult to provide rankings one after another when all of the music samples lacked cohesion. Generally, the samples that used harmonization ranked better compared to pure chaos. I would like to add to the harmonization aspect and allow for harmonization across the octave for melody1 and melody2. I am also going to make the bassline generation stepwise to produce more interesting results.