Music 171: Computer Music I
Assignment #2,
Due: Friday, October 18, 2019

In this assignment, you should create two loops having the same frequency sequence, that are played together but with one at a slightly faster rate, so that it gradually moves “out of phase” with the other. Depending on the rates, after a period of time, the loops should return to being “in phase”, that is, they will both be at the beginning of the loop at the same time. For instance, if it takes 1 s for one iteration of the loop and 1.5 s for the other, they will be back “in phase” every 3 s (after one loops 3 times and the other twice). Interesting effects arise if the differences between the timings is more subtle, such as 1 s against 1.1 s (in this case, after how many seconds would they be back in phase?).

To create a loop:

1. Make an array/table that will hold 8 frequencies in the range 200 - 1400 Hz. Example frequencies are given below but you may use your own.

   Use a message box to set the array's

   - size: 8
   - bounds: 0 1400 (top left x, y coordinates) 8 200 (bottom right x, y coordinates).
   - labels: increment x-axis by 2 and y-axis by 200
   - values: 990 770 880 1210 1320 770 1320 880 (starting at index ’0’)

2. FOR EACH LOOP (begin with one and then duplicate):

   - Using the `tabread` object, read from the array at consecutive indeces (see **counter** below) and use the value to control frequency of an `osc~` object;
   - Use a **counter** to automatically increment the index by 1 (at regular time intervals):
     - hold index value in a `float` object (right inlet) until a “bang” (left inlet) releases it;
     - use the `mod` object (modulo) to ensure indeces don't exceed the bounds of the array (range is 0 to 7);
     - use the `+` object to increment (add 1 to) the index before sending it back to the `float` object.
     - use a `metro` object to generate bangs at a regular rate; that is, to read through the entire array every second, `metro` should be set to 1000/8 = 125 ms.

3. Play each loop out a different channel of the `dac~` using the `line~` object to turn the sound on/off.

4. Use a single toggle to start/end both loops. Make sure that when starting again, both loop indeces are reset to 0 so sequences are “in phase”.