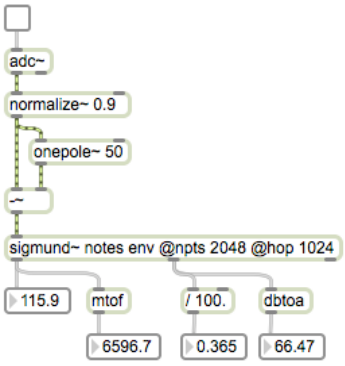
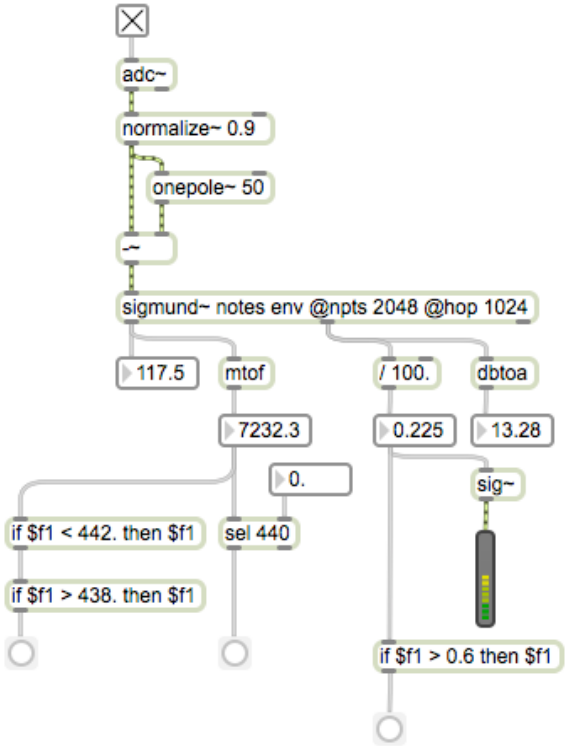
Composition: Electronic Media II
Spring 2013
[sigmund~] and [gizmo~]

1. [sigmund~] analyzes audio signal received in its inlet. Because we want to use this in a live situation, first connect the [adc~] to [sigmund~]. In the image below, [normalize~] is used for optimal signal strength. This feeds into a [-~], which is subtracting noise floor of 50 Hz via [onepole~], a low pass filter.
   a. Consult [sigmund~]'s help file for a full list of arguments and adjustable parameters. For our purposes, arguments "notes" and "env" will give you MIDI notes per the audio signal and amplitude in db out of the far left and middle outlets, respectively.
   b. Use [mtof] to convert MIDI notes to frequency in Hertz.
   c. Divide the env output by 100 to get a number between 0-1.



   d. Per the image above you may notice the "@npts 2048" argument and "@hop 1024" argument. These are parameters that limit [sigmund~]. "npts" is the number of points in the analysis, which must be a power of two and at least 128. "Hop" is the number of points between analyses, which must be a power of two and at least the DSP vector size (usually 64). The settings in the image above seem to work well.

2. Consider the module below. How is it operating?



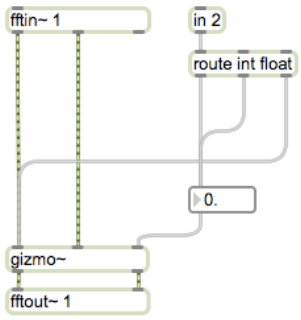
   a. For pitch tracking, we have at least two options. In the image above, [sel] is set to bang if 440 enters its inlet. This may prove unreliable in a live situation. A second option consists of a series of [if] objects.
   b. Amplitude tracking can also be tracked via [if] per the above image.

c. Discuss in class a solution that combines both amplitude and pitch tracking (say we want to trigger something when the performer plays an A440 quietly).

3. [gizmo~] uses fast Fourier transform to analyze the frequencies being input to Max.
   a. The short hand for fast Fourier transform is fft, which is found in a few different objects.
   b. In order to use [gizmo~], we must create an external Max patch.

4. In a new patcher window, create the following objects:
   a. [fftin~ 1]
   b. [fftout~ 1]
   c. [in 2]
   d. [route int float]
   e. [gizmo~]
   f. [float box]

5. Connect these in the following way and save this patch as "gizmo" in the same folder as your main patch:



6. In a new window, create a patch that receives and outputs live sound via [adc~] and [dac~].
   a. Include an object called [pfft~ gizmo_loadme]. This essentially means "do a fast Fourier transform to the patch *gizmo*," referring to the patch we just built in items 4-5.
   b. Include the twelfth-root-of-two semitone module used in other patches.
   c. Connect your patch in the following way:

7. Now if you turn on [adc~] and turn up the volume, you can hear your voice coming through Max. If you begin to change the semitone on the right of the patch, your voice is now transposed to that semitone in real time.

8. Both [sigmund~] and [gizmo~] are externals that you can find at: < http://www.maxobjects.com>