21M.370 Digital Instrument Design
Lab assignment 2 - Due March 3 at 2:00pm

Deliverables:

1. A PD synthesizer patch. This can be one of the provided examples, a modification of the examples, or a patch you have created on your own. You only need to submit the main.pd patch.
2. A PD control patch, with at least 6 controls (sliders, radio boxes) sending control messages to the synth patch.
    1. At least one of your control elements will control 4 or more synth parameters.
    2. Your control patch will also send and receive messages to python.
3. A python script which sends data to PD to be used to control synth parameters.

Assignment description:

Our goals for this week are to get comfortable using sends and receives to control synth parameters, and to install and use python 3 to generate control data.

You will need to have Python 3 installed, with the python-osc and pyserial modules installed as well. Look at the Python readme on the github (see below) for info.

You will also need have git installed in order to be able to pull the 21M.370 git repo.

For your synth PD patch you can use one of the examples provided or make your own.

For your control PD patch, you should make a blank patch and then add UI objects, sends, any math objects to scale the UI objects, and also add labels as appropriate. Don't forget to use the 'properties' of your UI objects to add labels and configure how the objects look and function.

For your python script, you are welcome to use the example provided to send sequences to 8steps. For this assignment it would be sufficient to just add more sequences to the sequence array. But you are welcome to go beyond this if you wish, and we will talk about more uses for Python in the future. Be sure to watch the video on PD and Python, which will show how to use Python to format OSC messages efficiently for use with automatonism.

The list of resources for this lab:

1. Video tutorials discussing:
    1. message boxes in PD
    2. send and receive objects in PD and automatonism
    3. specifying oscillator pitch
    4. using OSC to send data between PD and Python
    5. Demoing the example synths for this week's lab
    6. Getting started with git and python 3.
2. Here is a link to the playlist for the class, with all videos: https://youtube.com/playlist?list=PLHQPaeahzMrMDn-4Z-JfLQWPv8Gj2jYJw
3. The GitHub Python readme gives information on how to install Python, and how to set up Sublime Text to use Python 3.
4. Three demo instruments are provided for this week's lab, and are located in NIME/PureData/Instruments:

1. ExtCtrlSubSynth is a subtractive synthesizer with a demo of a control patch saved within it: subSynth-ctrl.pd. This has all of the functionality needed to complete this week's lab.
2. WestCoast is a west coast-style synth, with wavefolding, a low pass gate, and FM modulation.
3. FMkeygrid is an FM synth which uses the `keygrid` object to use your laptop keyboard as a control. The 'keygrid' object is in the PureData/externals folder - you will need to add that folder to the PD filepath (PD->preferences->path).

5. Also check the github's PD readme for more on working with PD.
6. An example Python script, located in the github under Nime/Python/Examples.