# General Notes

Pure Data (pd) is a visual programming language. It is often used for audio programming and manipulation but it is a fully realized language that can manipulate visual elements and perform many standard functions. It has been used to create dynamic music in various ways, including games. The most notable game is *Spore* (EA, 2009).

Programs in pd are called patches.

Note: If you want to hear the sound, you will want to set up the audio. The simplest way to do this is to click on Media > Standard MIMO. Make sure your devices are set up appropriate for your machine. The most important is the Output Device which should map to your sound card).

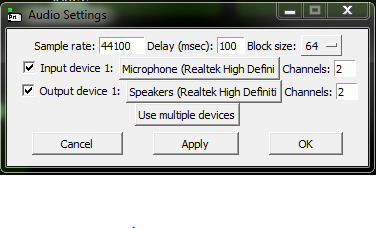


Figure : Audio Settings

Test the sound by clicking Media > Test Audio and Midi.

In order to see the various patches I worked on, open the pd-exended program and click File > Open).

Make sure the DSP is checked to hear sounds.

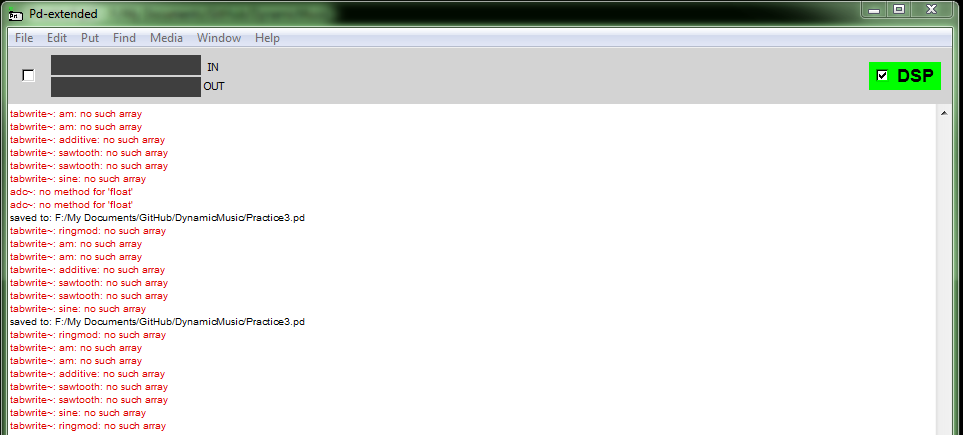


Figure : Output Console and standard window (note DSP in upper right)

# Practice1.pd

Some basic pd functionality. If you click on top items in each patch, it will perform various actions. This set includes the following:

* Top-Left: Click on “servus” it converts it to a symbol and puts the value (“servus”) in the symbol receiving box at the bottom
* Top-Right: Click on numbers and they go through add function.
  + If click on left numbers (1,2,3) it is immediately added and output.
  + If click on right numbers (5,4,6) then the value is added but put not output. Then clicking on one of the left numbers again will cause you to get the added numbers (e.g. click 5, then 2 and it will output 7)
* Bottom Left: Two variations of counters. Click on counter and it will add 1 to the number. If you click on the 0, it will reset.
* Bottom Middle: Trigger functionality. Click on radio button (called a bang) to start execution. The “trigger bang bang bang” fires off the 3 outputs on bottom of trigger in right to left order (so in the console you should see Print: 1, Print: 2, Print: 3)
* Bottom Right: Mapping to a List
  + Clicking on bang to left will output the values from right to left.
  + Clicking on bang to right goes through the trigger and outputs them in order. There is a print to the console and one that is visible in pd.

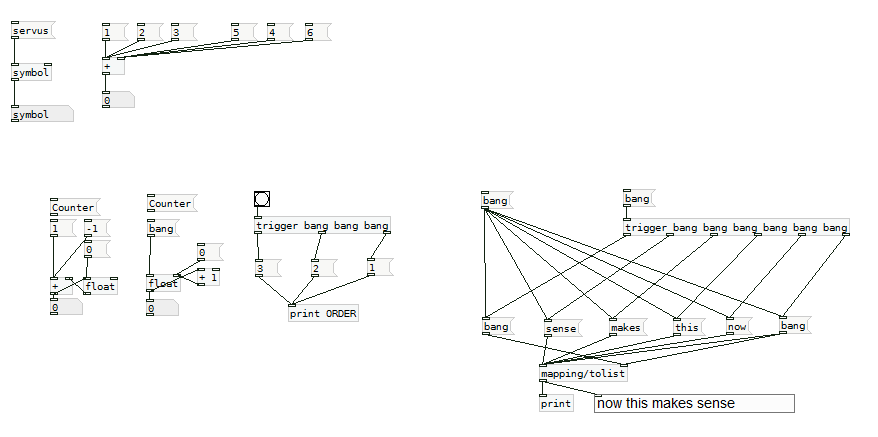


Figure : Practice1.pd patch file

# Practice2.pd

This one has a mix of experiments and notes.

* Top-Left: Hit bang to left and start a counter. Hitting one to right resets the counter.
  + If you double-click on “pd testCounter” you can find a sub-patch (equivalent of a function) that increments a counter after a delay of 1000 ms. Note that 1000ms is not always consistent (starts out much faster and then slows down). Something I will need to account for.
* Top-Right: If you click in the green box (a toggle) it will then execute a program that modifies the size and color of the vertical slider to the right.
  + Again there is a sub-patch (“pd modify”) that executes and does all of the actions.
* Middle-Left: This is an example of how you import an extension library and then below it is a waveform visualizer imported from the cyclone library.
* Middle-Right: A demonstration of the ability of pd to map using a “select” statement.
  + Clicking on 1, 2, 4 will automatically cause the accompanying bangs below to fire. The 400 and the symbol text do not map to any of the select arguments and are passed through to the print box.
* Bottom-Left: A grid that plays sounds based on position in the grid. Clicking in the grid and moving it around creates new sounds.
* Bottom Middle: An example of functions that convert a value from Fahrenheit to Celsius and vice versa. To use, click and drag in the top number boxes to increase/decrease the number value.
* Bottom Right: A wave-form generator. This is not hooked up to anything, left there as notes so I could find it again easily later.

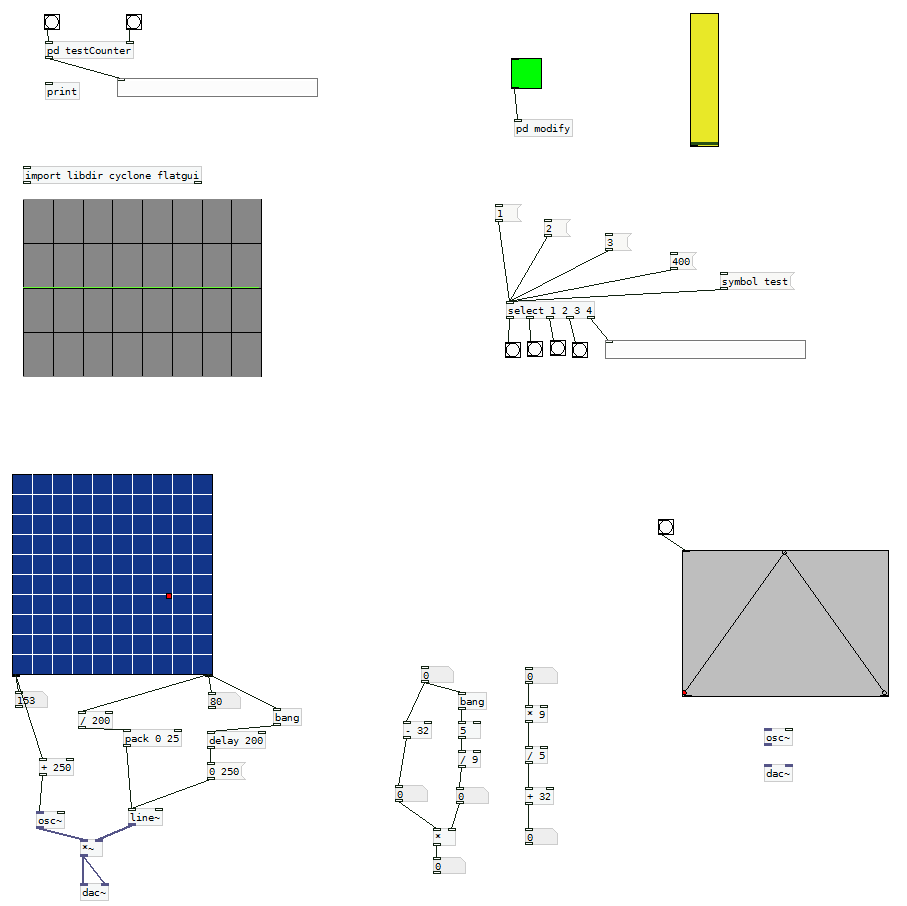


Figure : Practice2.pd patch file

# Practice3.pd

Warning: this one could hurt your ears.

This is a bunch of experiments creating various sound generation tools. Each is labeled. To use, click and drag in the horizontal bar below the label.

The toggles were an attempt to get the sound to not play until the object was activated, but it didn’t work.

Check the red boxes to toggle on/off graphs. Move the slides in light green/blue boxes to adjust the waves.

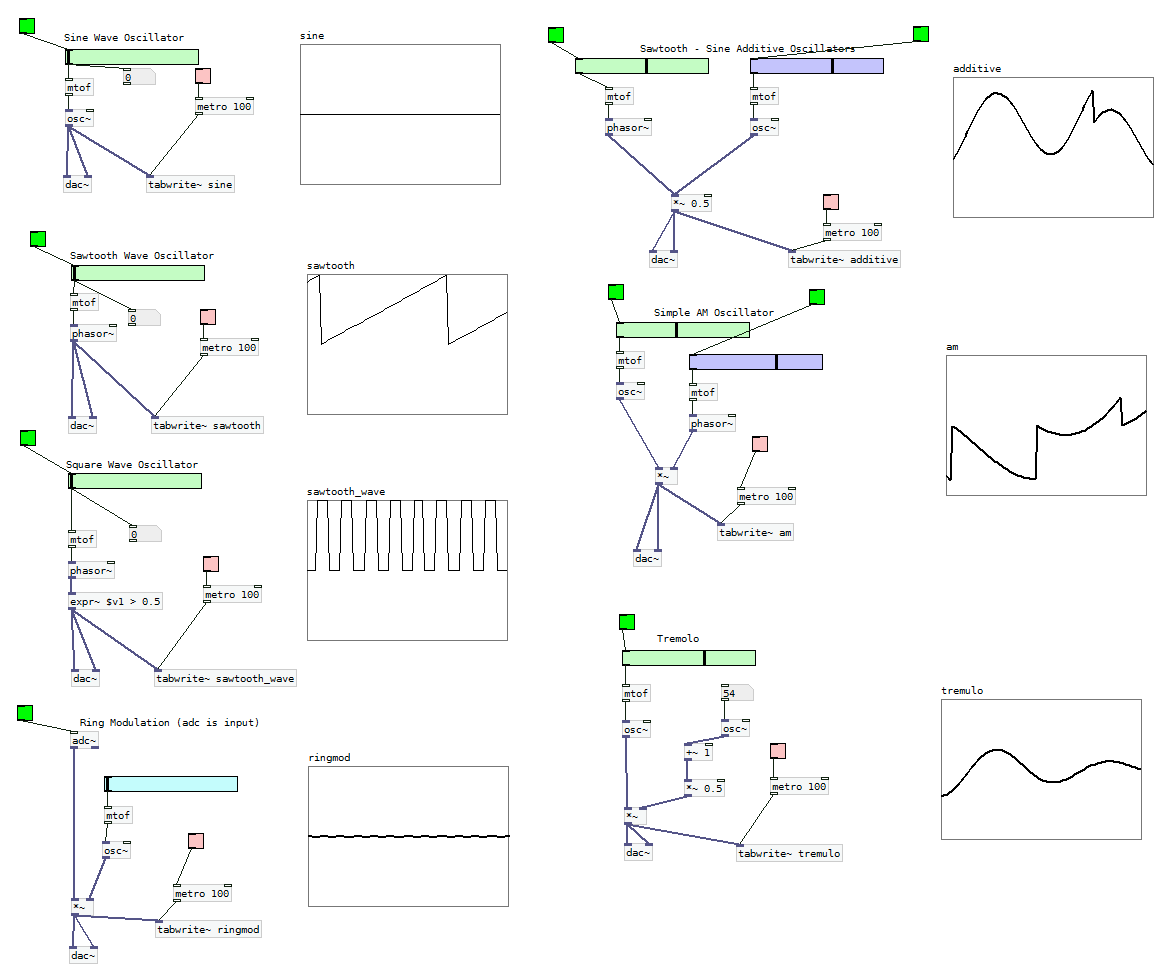


Figure : Practice3.pd patches

# Practice4.pd

This is a duplicate of Practice3.pd without the toggles or graphs. Made it for future experimentation.

# Practice 5.pd

Figure : Practice5.pd patches

More experiments with various sound generation modules. Red toggles in each patch toggles on the graph to the right of the patch.

* FM Modulation
  + The green horizontal slider adjusts the carrier frequency
  + The blue box (click in this and drag up/down to change the number) represents the modulation frequency in Hz
  + The purple box (again click and drag to change) changes the modulation amount in Hz
* Pulse Width Modulation
  + Green slider adjust MIDI note values from 0-127.
  + The blue box (again click drag to change) represents the pulse width.
* Pulse Width Modulation w/ square wave and LFO
  + Green slider adjust MIDI note values from 0-127.
  + The blue box (again click drag to change) adjusts the waveform by selecting specific frequencies between 0 and 1 (it takes number 0-100, divides by 100, puts into sine wave oscillator and then gets absolute value).