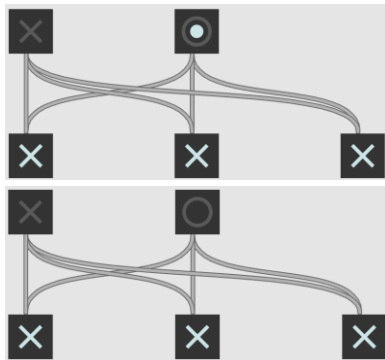
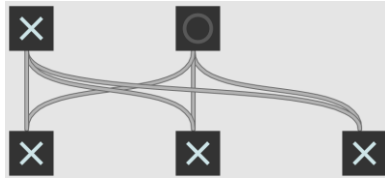


## Lecture 13 – Max/MSP Basics II

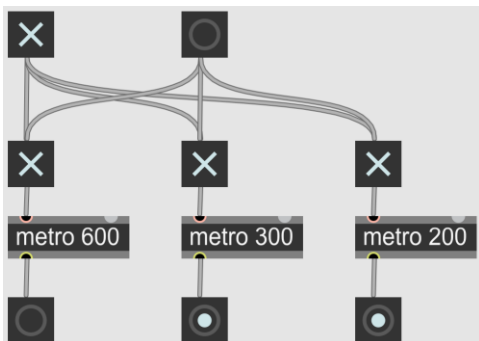
Instructor: Hao-Wen Dong

### Example 1: Polyrhythm (“1\_polyrhythm.maxpat”)

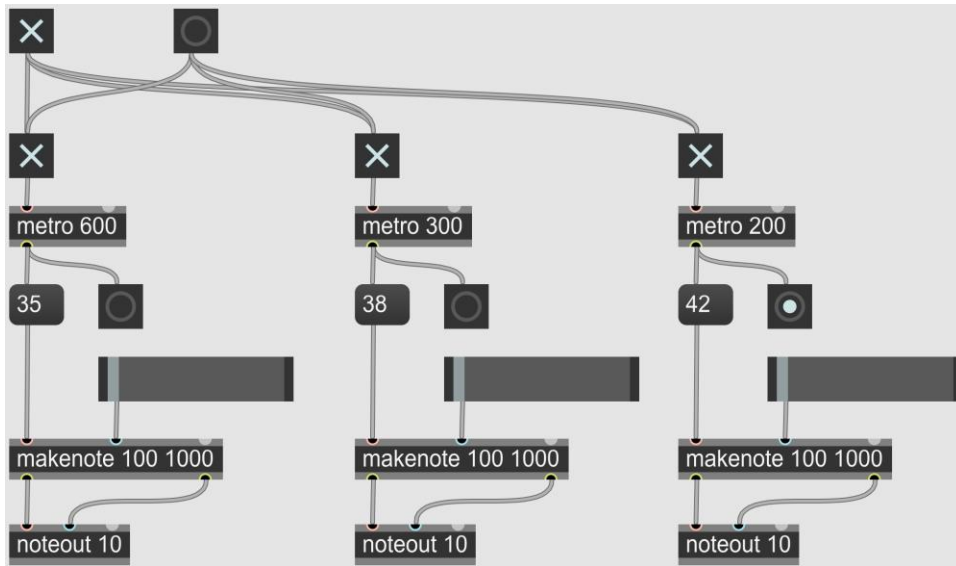
- Toggles & buttons
  - A toggle maintains the state
- A button sends out a one-off bang message



- “metro” objects



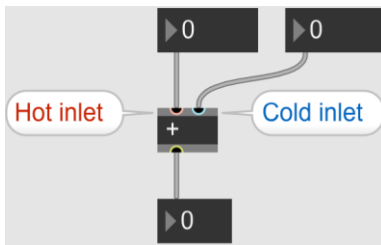
- Polyrhythm!



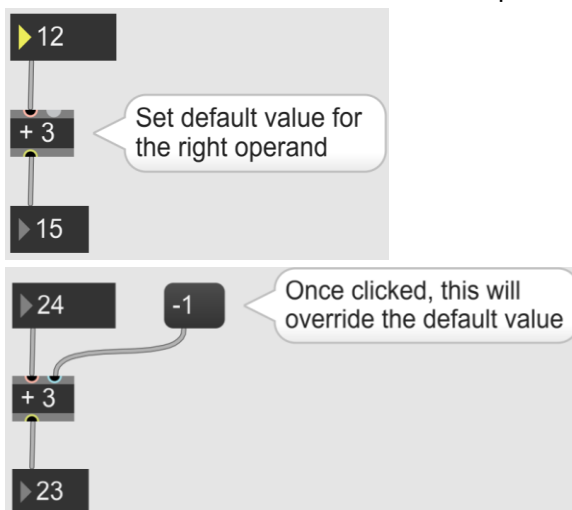
(Note that “noteout 10” set the default MIDI channel to 10)

## Example 2: Math (“2\_math.maxpat”)

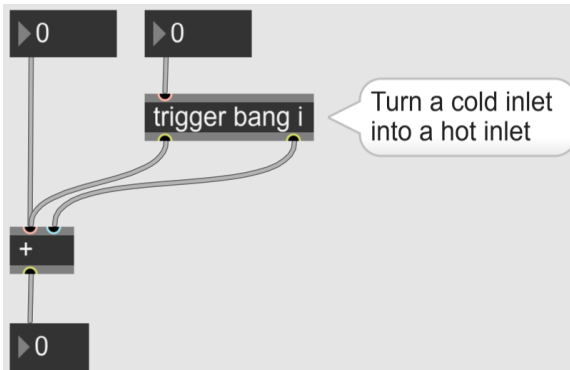
- Hot inlet vs cold inlet



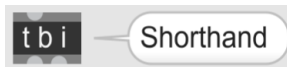
- Set default value (acts like the “+=” operand)



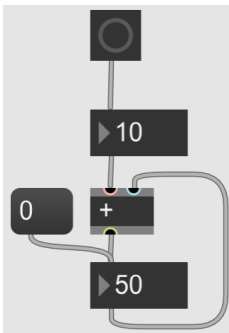
- Use “trigger” object to turn a cold inlet into a hot inlet



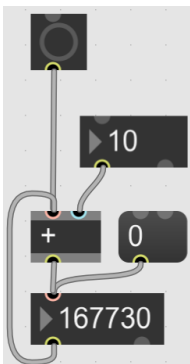
- Shorthand



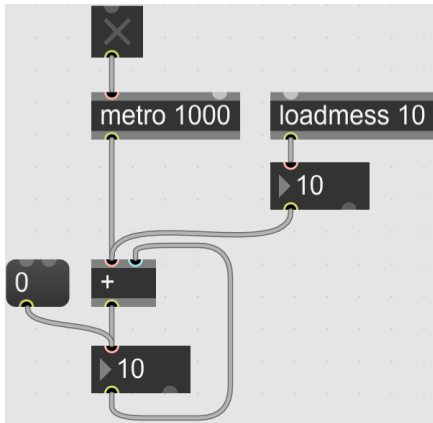
- A simple counter



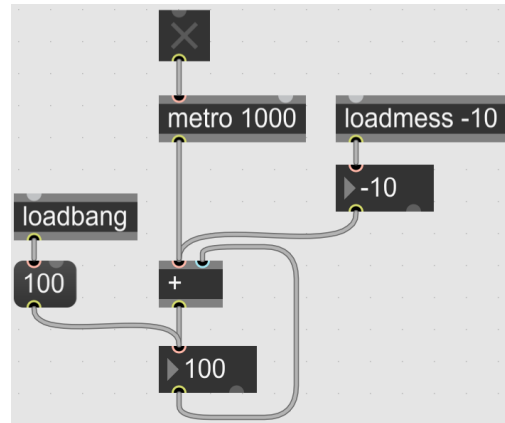
- What if? ...stack overflow!



- An automatic counter



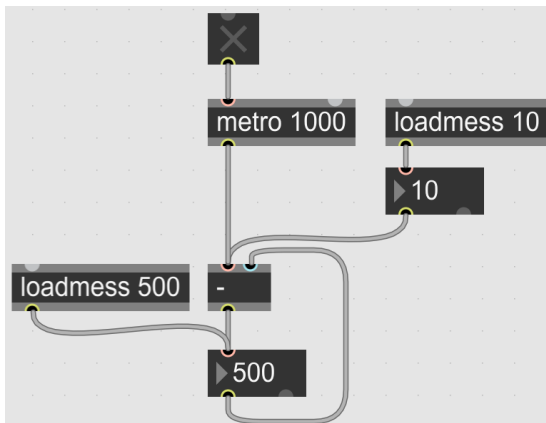
- An automatic timer



- Use a "loadmess" object to send a message at load time



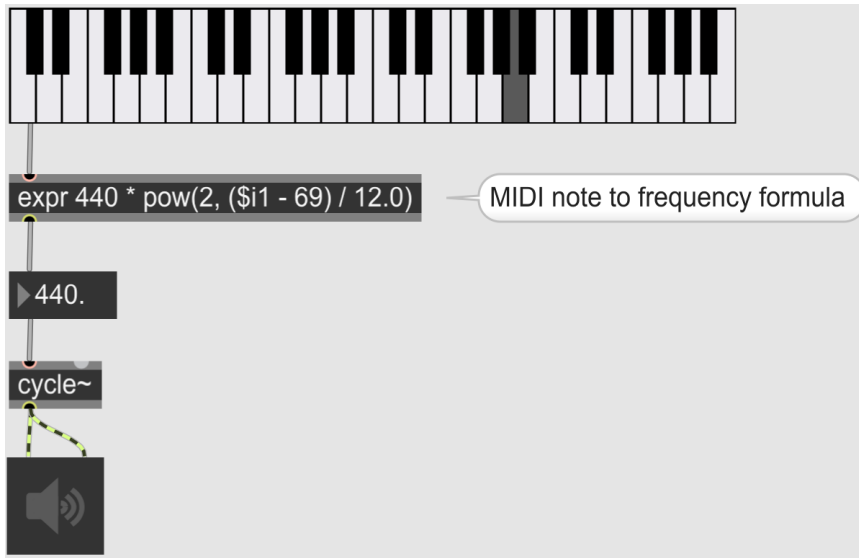
- What if?



- Use the "expr" object for math expressions

expr 440 \* pow(2, (\$i1 - 69) / 12.0)

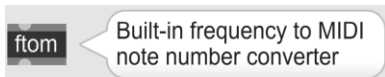
- A simple sinusoid synthesizer



- Use “mtof” to convert MIDI note number to frequency

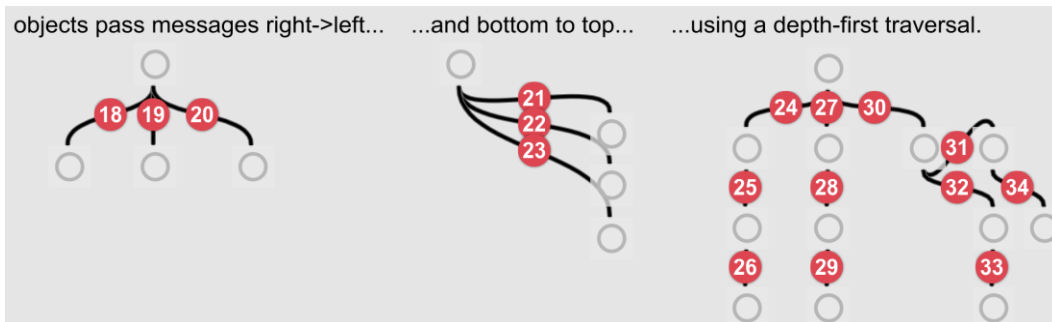


- Use “ftom” to convert frequency to MIDI note number

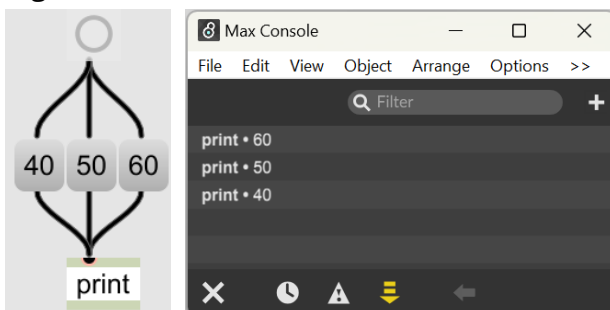


### Example 3: Message Ordering (“3\_message\_ordering.maxpat”)

- Right-to-left, bottom-to-top!



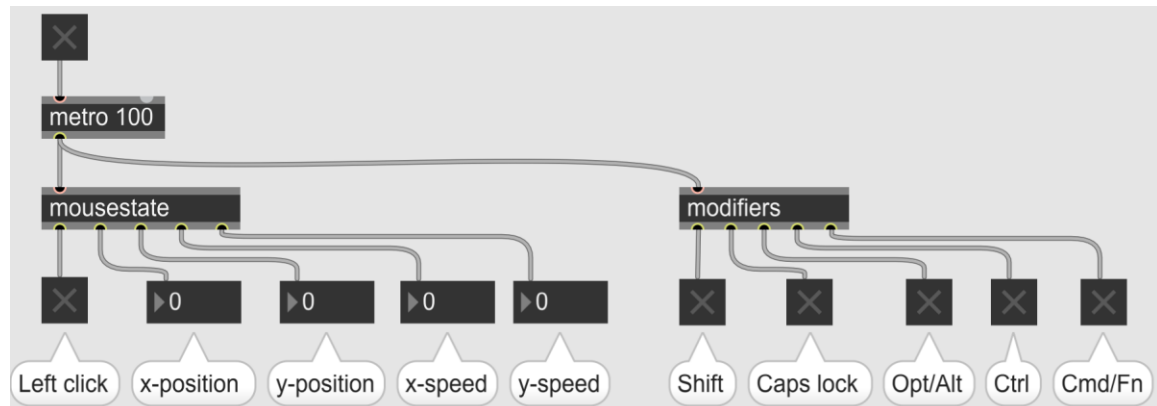
- Right-to-left, as shown in the Max console!



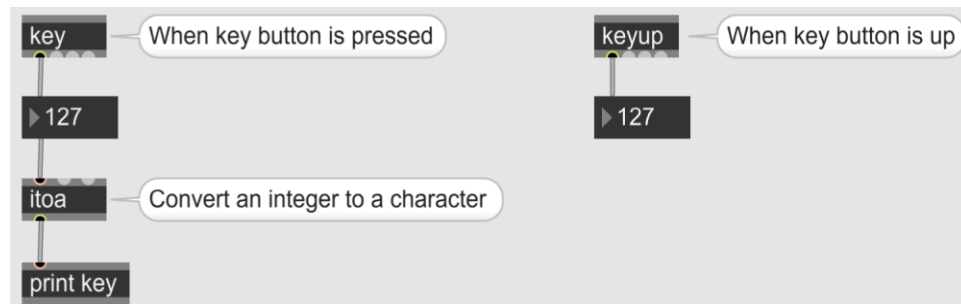
## Example 4: Mouse and Keyboard Controls ("4\_controls.maxpat")

- Use the "mousestate" object to get the "left click", "x-position", "y-position", "delta-x", "delta-y"

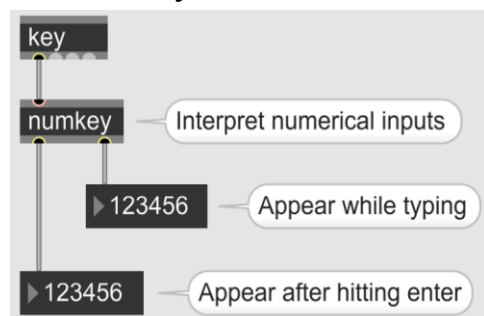
Use the "modifiers" object to get the "Shift", "Caps Lock", "Opt/Alt", "right click", "Cmd/Ctrl"



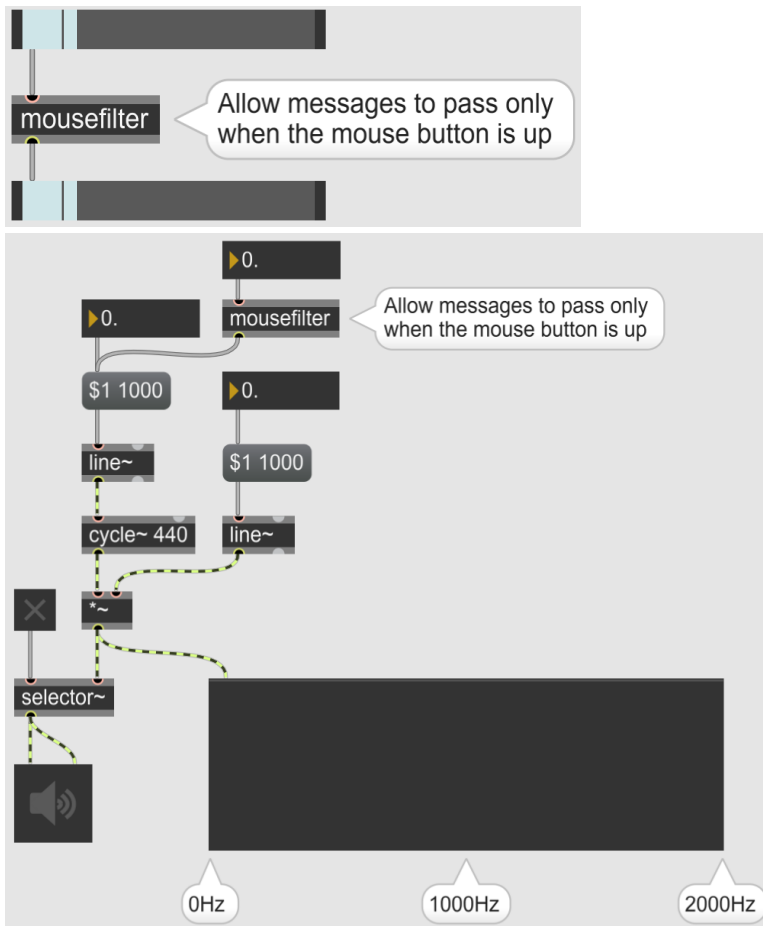
- Use "key" and "keyup" objects to read the keyboard inputs



- Use "itoa" object to convert an integer to its corresponding ASCII character
- Use "numkey" to handle numerical inputs → hit enter when done!



- Use the "mousefilter" object to control the data flow → the data can only pass through when the mouse is up

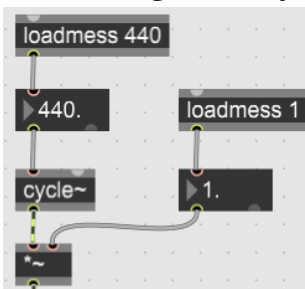


### Example 5: Simple Synth ("5\_simple\_synth.maxpat")

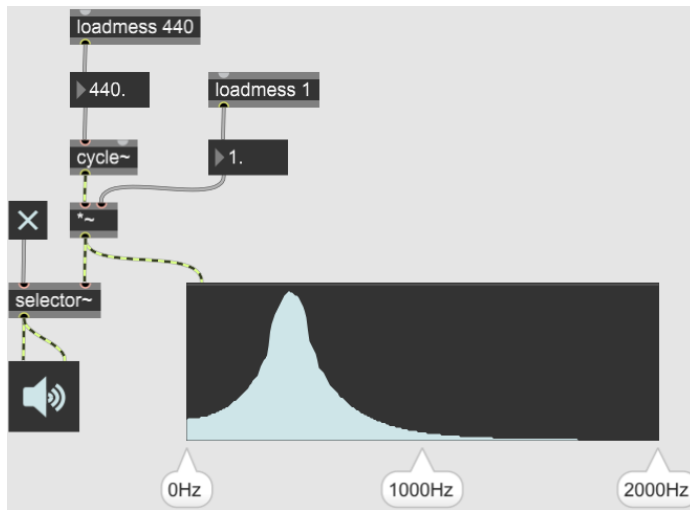
- Use the "cycle~" object to create a sinusoid oscillator



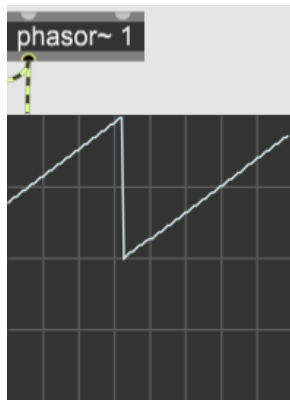
- Set the magnitude by "\*" → multiplication of signals



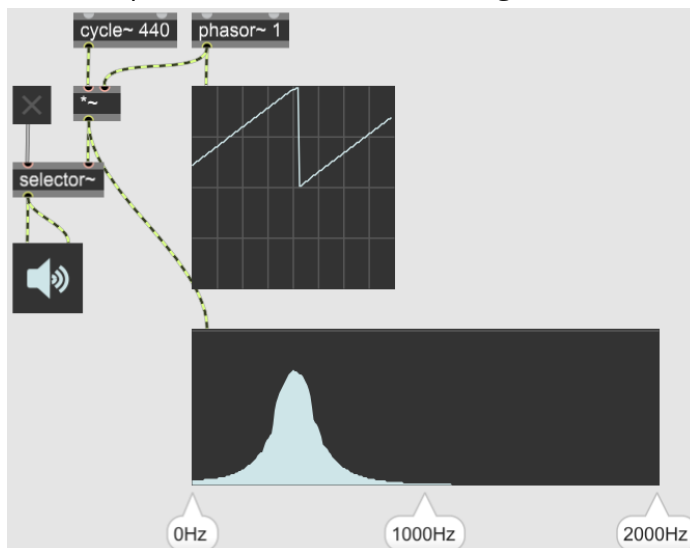
- A simple sinusoid oscillator



- Use the “phasor~” object to create a saw-tooth signal

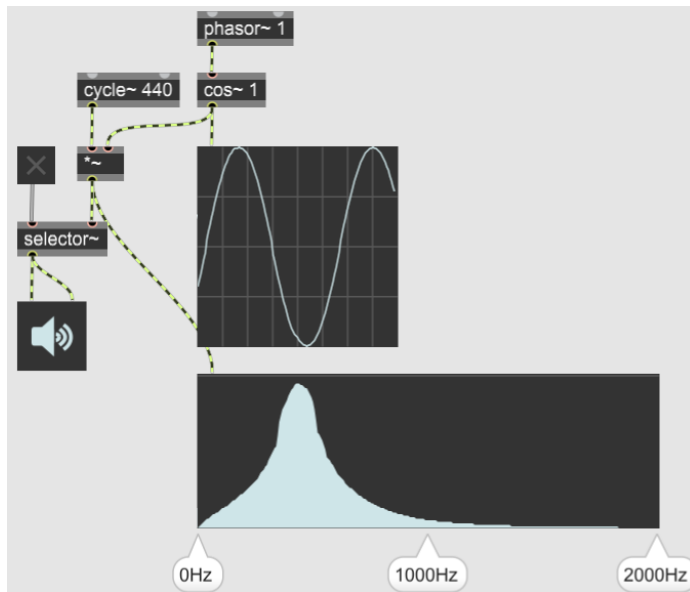


- Use the “phasor~” to control the magnitude

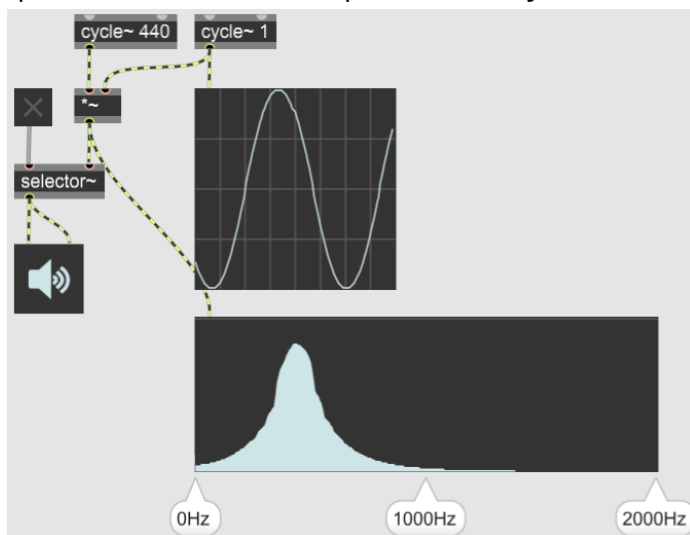




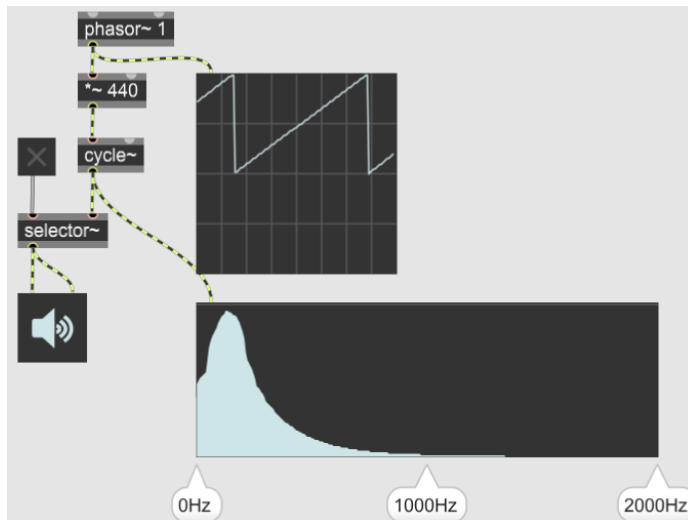
- Turn the “phasor~” object to a sinusoid wave



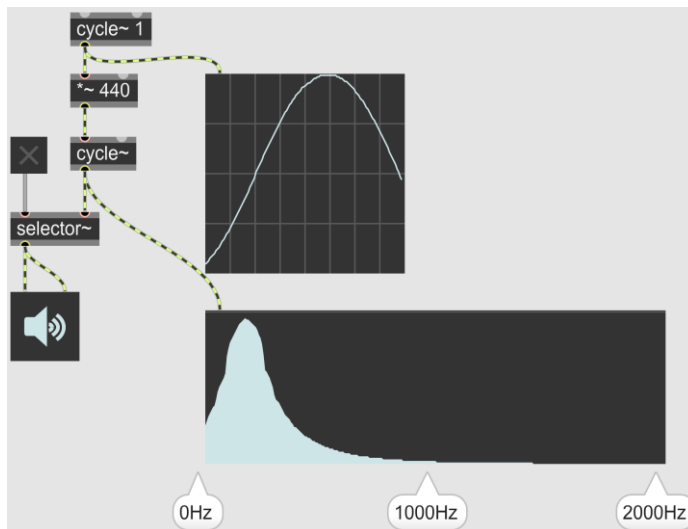
- “phasor~ 1” + “cos~” is equivalent to “cycle~ 1”



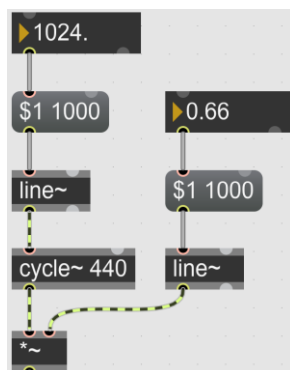
- Use "phasor~" to control the frequency of the sinusoid oscillator



- Use "cycle~" to control the frequency of the sinusoid oscillator

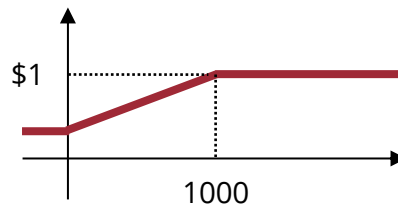


- Use "line~" to create a linear ramp generator



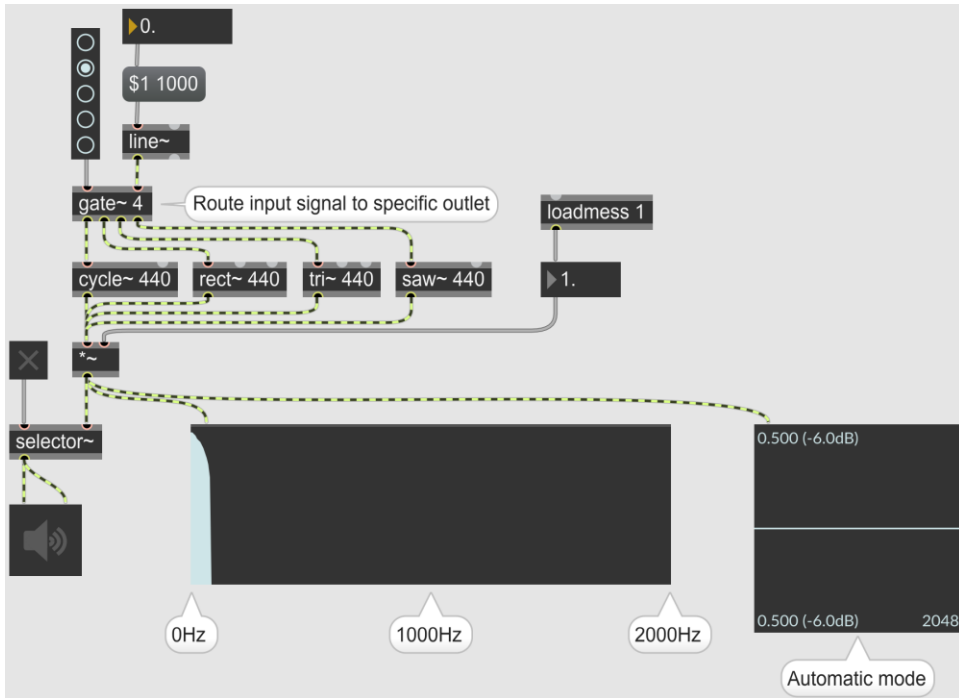
- 
- The screenshot displays a Pure Data patch for generating a sine wave. The patch starts with a message box '440.' connected to a 'line~' object with a size of 1000. This is followed by a 'cycle~ 440' object, which is multiplied by a signal from a 'line~' object (also size 1000) at a multiplier of 1. The result is then multiplied by a signal from a 'selector~' object. The output is connected to a speaker icon. A spectrum analyzer shows a peak at 440 Hz. A scope shows a sine wave with an amplitude of 1.000 (-0.0dB) and a frequency of 601 Hz. The scope is labeled 'Automatic mode'.

- 
- 1.
- \$1 1000
- line~



### Example 6: Gates & switches ("6\_gate\_switich.maxpat")

- Use "gate~" object to route input signal to a specific outlet



- Use “selector~” object to route the signal from a specific inlet to output

