

Instructions before using Pd

1. Before using any CCRMA computer you will need to have a CCRMA computer account and have set a password.
2. There are two ways to run a program in Linux. You can go to the Main menu -Sound & Video- and seek for the name of the program or -the more recommended way- just typing the name of the program in a terminal. To open a terminal you will need to make a right click on the screen. Then go to New Terminal and left click.
3. Create a working directory for storing the files that you'll make and save. In the terminal, type the unix command
mkdir pd-lab
...or some other name of your invention instead of "pd-lab." Then, set that to be the working directory with the command
cd pd-lab
4. Run the Jack audio server with the command
qjackctl &

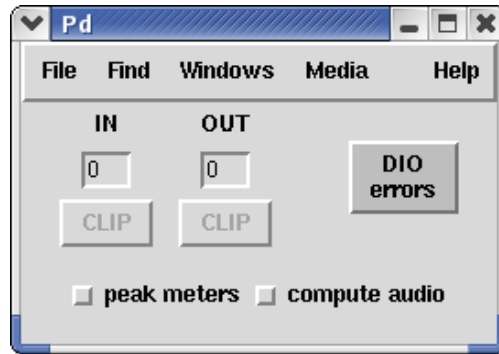
and then click Start



4. Run Pd -a real time graphical programming environment for audio and graphical processing- typing:

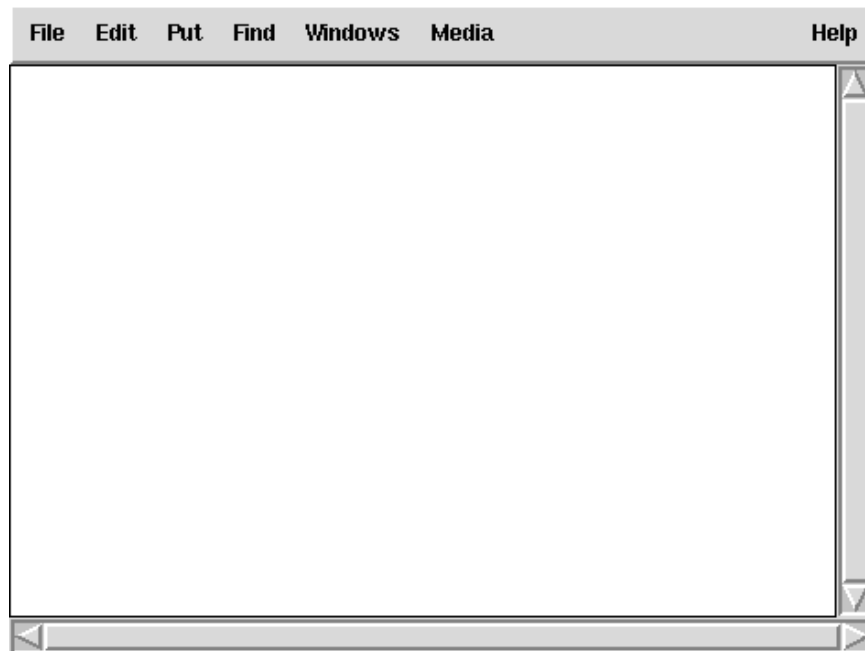
pd -jack&

this is what you will see if Pd is working

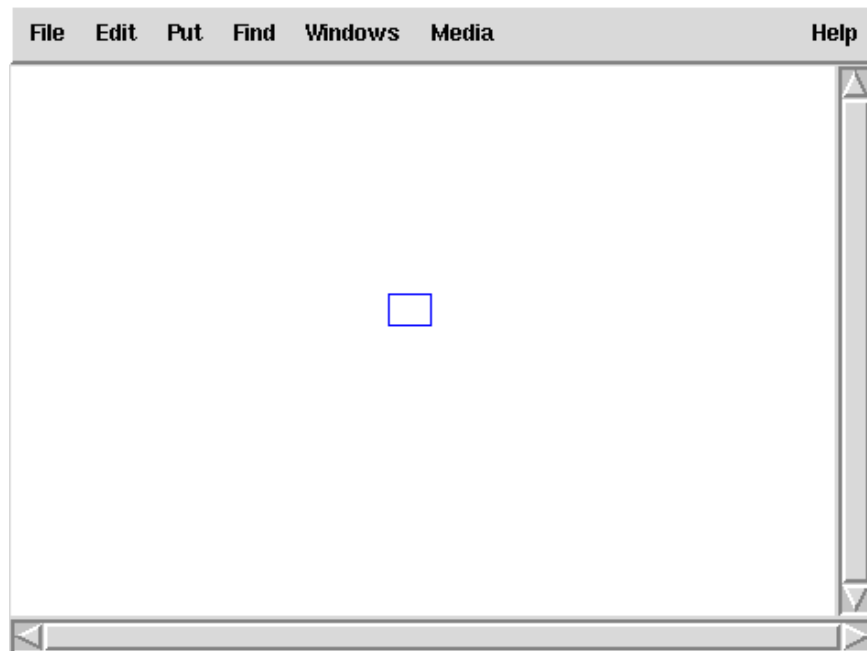


The basics of Pd

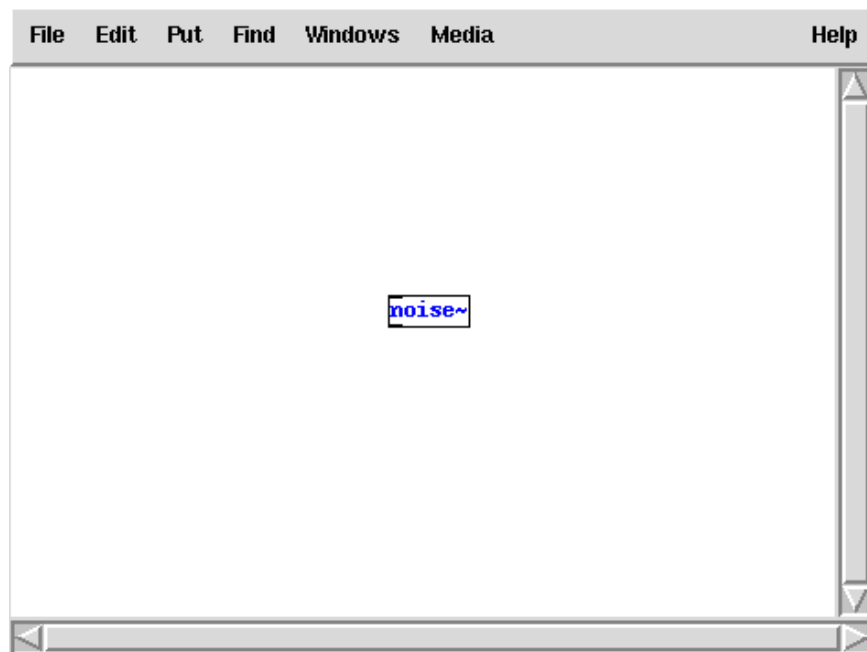
First, make a new patch (**File - New**, or **ctrl+ n**)



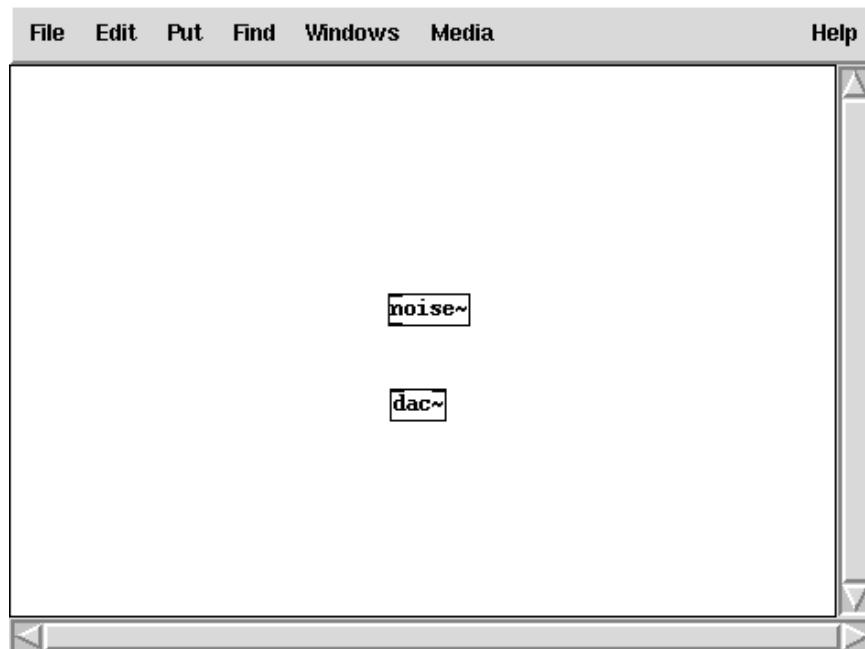
Now we can begin to create in the patch window some basic sound like a white noise or noise generator. For that we will need to create an object (**Put - Object**, or **ctrl+ 1**)



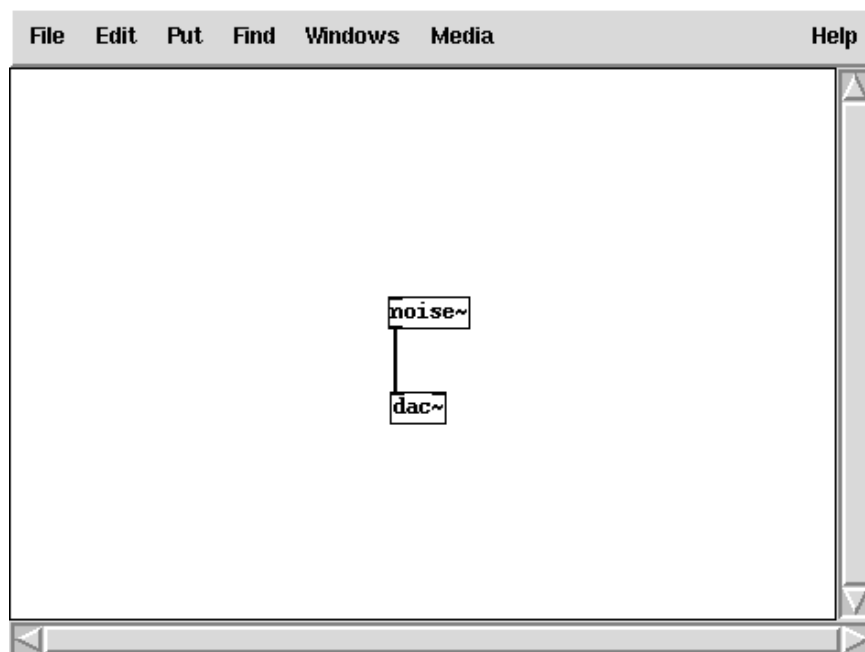
Inside the box type **noise~** and click outside the object box to 'activate' it. The ~ at the end of the name means it's working at audio frequency rate.



To hear the sound, a digital-to-analog converter is needed. Put another object named **dac~** just following the previous steps.



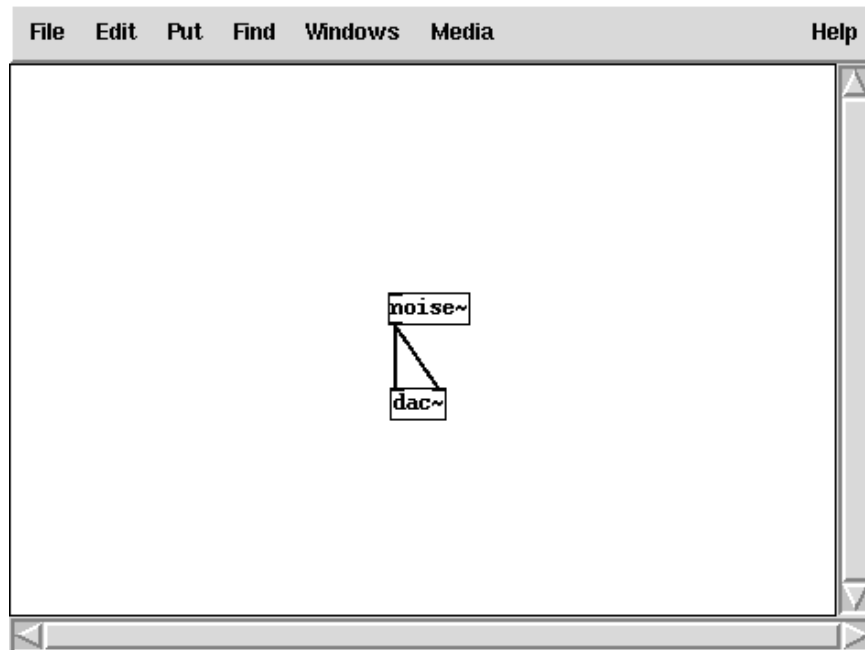
By dragging the mouse, connect the bottom-left corner of **noise~** (which is the output of the object) to the top-left of **dac~** (the input to the left channel of the **dac~** object). Notice that the cursor turns into a small circle when placed on the "nodes".



Turn on the audio (click **Compute audio** or go to **Media - Audio - On**, or **ctrl+ /**), and check the noise coming from left channel. (To turn off click again on **Compute audio** or go to **Media - Audio - Off**, or **ctrl+ .**)

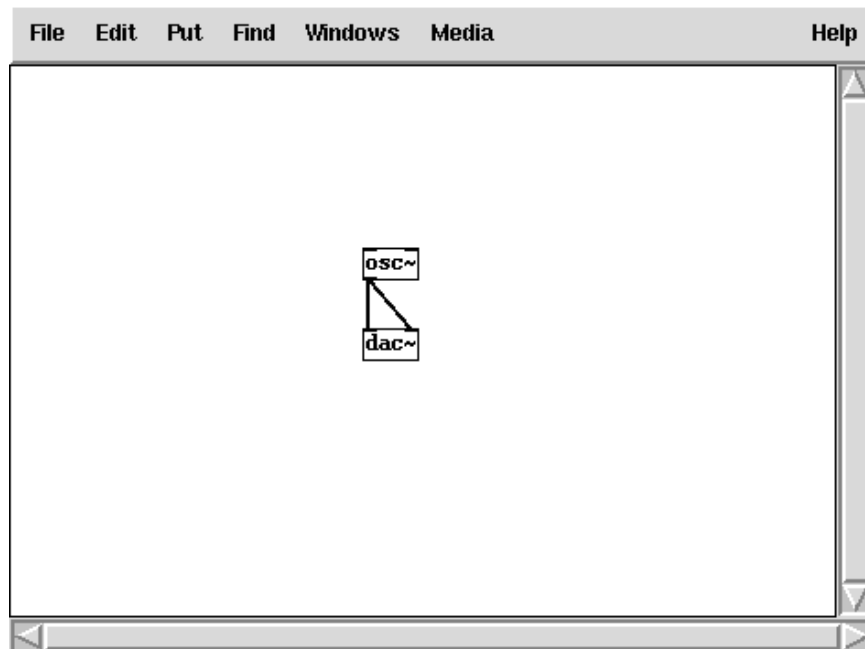
If you want the same noise coming from both channels, make another connection from

the **noise~** object to the top-right of **dac~**.

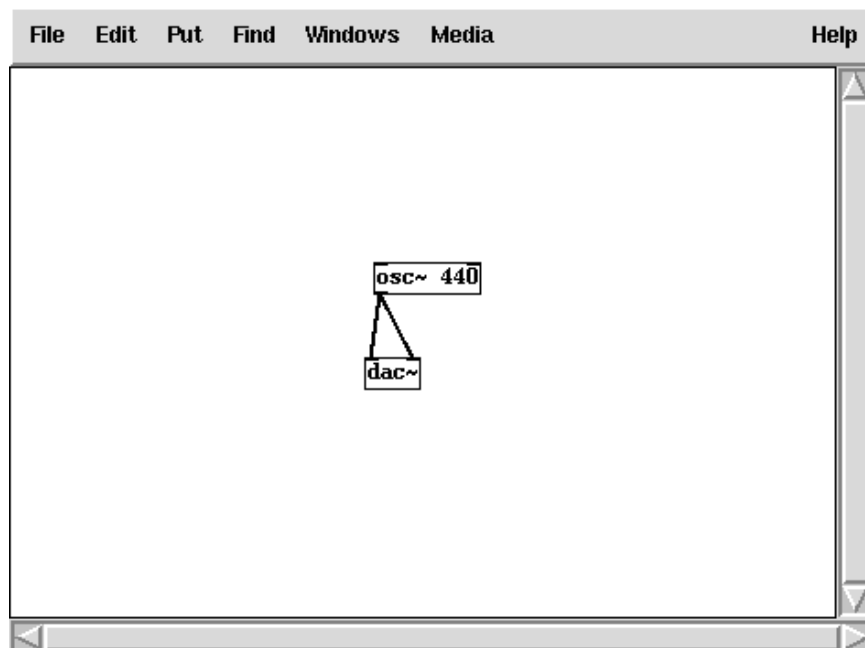


Saving your work is simple. Menu command **File - Save** will prompt for a new file name. Once it is saved, get it back with **File - Open** and it reappears. But, notice it will open with Edit Mode turned off. If further editing is desired, turn on Edit Mode with **Edit - Edit Mode** or the equivalent keyboard command, **ctrl+ e** (this “toggles” the mode on or off).

Now, instead of a noise generator we will create a periodic sine wave or sinusoidal oscillator. Create an object typing inside **osc~** and again connect the bottom left of the object to the two inputs of the **dac~** object.

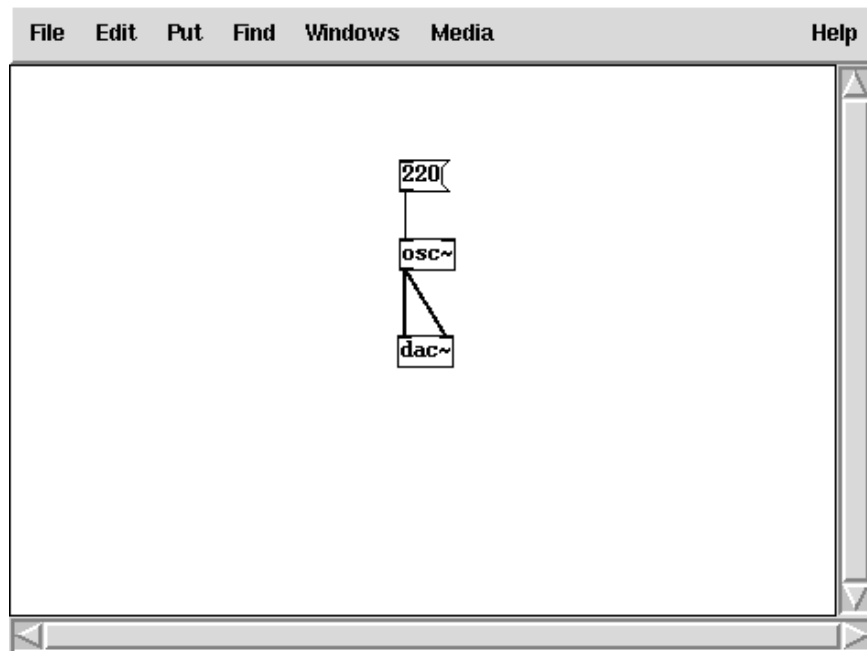


Notice that there is no sound coming from your speakers since the **osc~** object needs a number to determine the frequency of your oscillator. The object defaults to 0 Hz. frequency, so at this point even though it is sending a signal to your audio output, you won't hear the sound of an oscillating wave. Like many objects that require parameter specifications you can add an argument to change its behavior. Use **osc~**'s initial frequency argument to bring it into the hearing range. Type in 440 to hear the pitch corresponding to the *A* above *middle-C*. Save this patch forever... it's useful for testing audio output and tuning instruments.

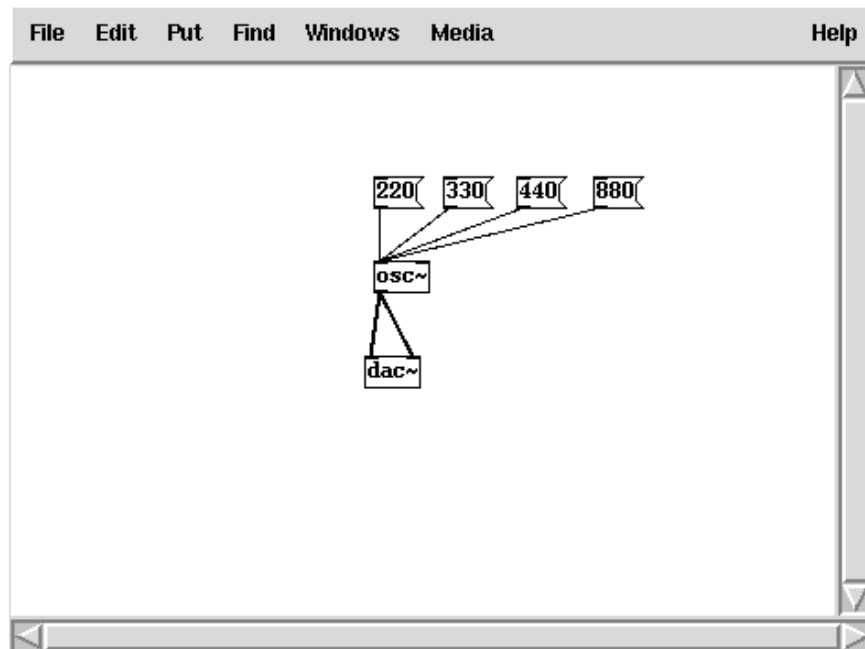


For more information about the input parameters that control an object, see its "Help patch" by right-clicking on the object and choosing "Help". Reading the information that appears in the one for `osc~`, you will know that the left input is for frequency and the right one for phase. We can take advantage of these inputs for real-time changes when running the patch in "Performance Mode" (as opposed to Edit Mode).

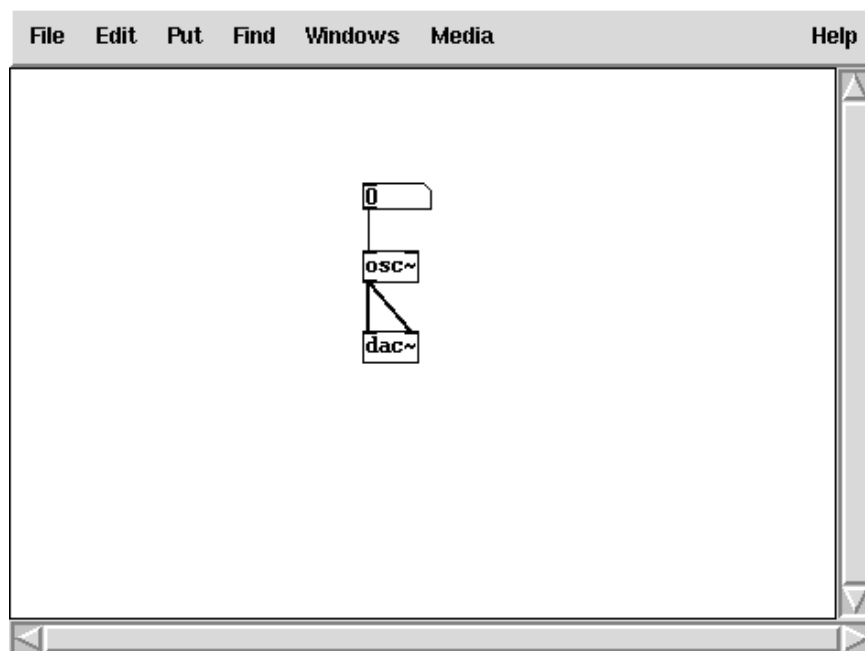
Your tuner patch might want to allow different frequencies to be selected. Send numbers to it with message objects (**Put - Message**, or **ctrl+ 2**) and type the number of the frequency you want inside these boxes. Remember that creating an object or rewiring things you will have to return to the Edit Mode. Make the changes that you see here and, if you want a frequency an octave lower than the previous one, type 220 in the message object. To hear it, get out of Edit Mode and click over the message box. By clicking over the message, the `osc~` object receives the new frequency number. Now you should hear it an octave lower.



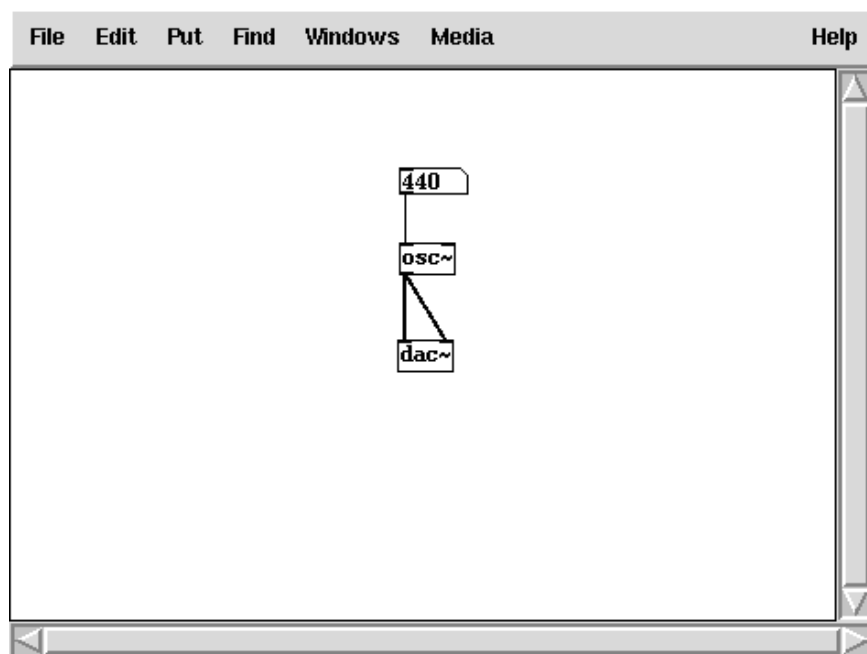
Multiple copies can function like presets for your tuner.



For continuous control use a number box object (**Put - Number**, or **ctrl+ 3**) and connect it to the left input of the **osc~** object. We need now to change the value of the number. To achieve that you will have to get out of the Edit Mode and drag-n-drop the mouse over the number box scrolling it for a Theremin-like glissando.



(or type in an exact number – this is allowed in “Performance Mode” in this case, whereas the previous message object method only allowed that in Edit Mode)



Again, to get more information about the different objects see "Help" by right-clicking on the object and choosing "Help". A complete on-line manual about Pd can be found at http://www.crcs.ucsd.edu/~msp/Pd_documentation/index.htm