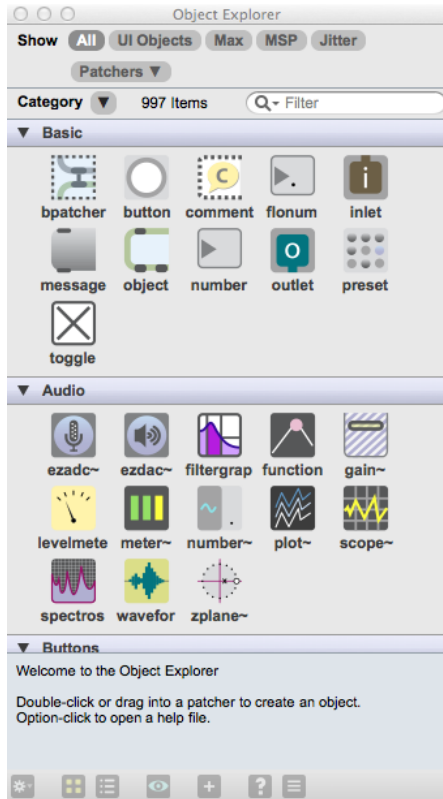


## Composition: Electronic Music II

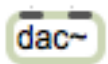
### Basic Sound in Max

Spring 2015

1. Before we talk about sound, we need to quickly discuss creating objects in Max.
  - a. There are different ways to create new objects in Max. For this handout, we will be using the “object explorer”, or what we often call the “palette”.

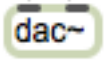


- b. You can access the palette by double-clicking on any empty area of the edit window. This brings up a new window with a scrolling list of objects you can use in Max. You can scroll down to find the object, or if you know the name of the object you want, you can type it into the filter bar to find the object quickly.
    - c. Once you have found the object you want, double click on it, and it will then appear in the edit window. Notice, the object appears where you originally double-clicked in the edit window to open the palette. This allows you to precisely place new objects as you create them.
2. Basic Sound in Max: The dac~ object ([dac~] herein) controls the output of sound from Max/MSP.
  - a. To enable sound output, the [dac~] must be turned on. To accomplish this, two objects are required: the [dac~] object and a toggle.
  - b. Create a [dac~] using the method described above. Once you have done this, the [dac~] will appear in the edit window.



- c. The two black lines at the top of the object are called inlets. Inlets always appear on the top and outlets appear on the bottom of any object you create in Max. The number of inlets and outlets varies depending on the function of each object.

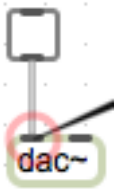
- d. Next, create a toggle object. This can be easily found in the “basic” subwindow of the palette. Of course, you can always search for the toggle in the filter bar.



- e. Now, position the mouse over the outlet of the toggle. A red circle will appear around the outlet when the mouse is properly positioned.

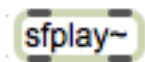


- f. Click the mouse, and drag to the left inlet of the [dac~]. When a similar red circle appears around this inlet, release the mouse button.

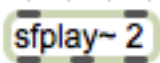


- g. This will create a patch cord connecting these two objects. This allows the toggle to communicate with the [dac~]. Patch cords are one-way paths. Information travels from one object’s outlet to another object’s inlet.  
h. We now have a way of controlling the [dac~]. By switching the toggle on or off, we can turn on and off sound output from Max.

2. [sfplay~] : this object plays back sound files from the hard drive. This object is capable of reading aiff. or wav. files.  
a. Create an [sfplay~] object.

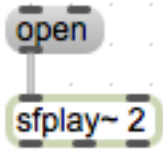


- b. Notice that it has two inlets and two outlets. If you hover your mouse over any of the inlet or outlets, a black box appears with a small clue as to what that inlet or outlet does.  
c. Notice that this [sfplay~] only has one outlet for channel 1. Since the files that we will be playing in this class are in stereo, we need to add another outlet for channel 2.  
d. Double click on the [sfplay~] object. The name of the object is highlighted in purple. Retype the name of the object so it reads “sfplay~ 2” and then hit return. Notice that there is a space between the “~” and the “2”.  
i. We now have three outlets.  
ii. The middle outlet is a new channel 2 outlet.

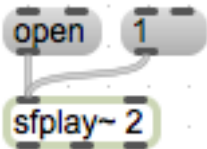


- e. What we just did was create an argument for the object. Many Max objects accept arguments, which are initial parameters you want the object to observe. Since we want [sfplay~] to play stereo files, then we tell it that by giving it a “2” argument. Arguments are always separated from the object’s name by a single space.

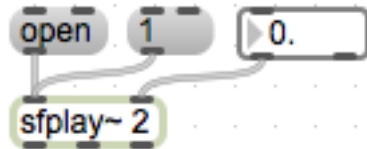
- f. Now, we are interested in the left inlet, because this is where we control [sfplay~] with Open, Play, and Stop commands.
- g. To tell the [sfplay~] object what to do, we will use message objects. Create a message object, and then type into it the word “open”. When you are done typing, click anywhere outside of the message object. Then connect the message outlet to the left [sfplay~] inlet with a patch cord.



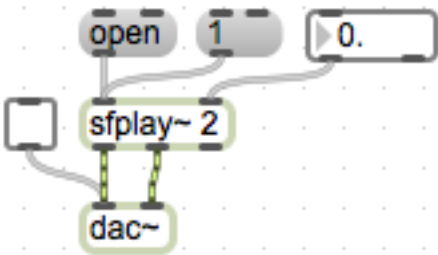
- h. The second message box we will create will contain only the number 1. Connect the outlet of this message box to the left inlet of the [sfplay~].



- i. The right inlet of [sfplay~] is used to provide a speed ratio for playing back the opened audio file. This functions exactly like a varispeed pitch shift operation; sending a value of 0.5 will result in playback twice as long and lowered an octave in pitch, a value of 2 will result in playback half as long and raised an octave.
- j. The simplest way to control the playback speed is by creating a flonum object, and attaching the left outlet of the flonum to the right inlet of [sfplay~], as shown below.



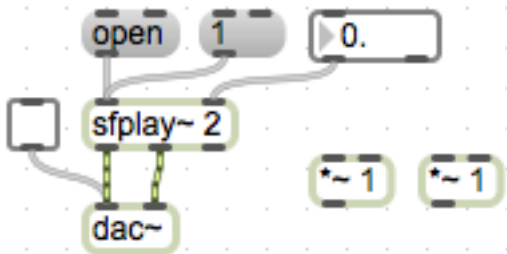
- 3. Connecting the the [dac~]
  - a. Create patch cords connecting the channel 1 outlet of [sfplay~] to the channel one inlet of [dac~], and two to two as shown below:



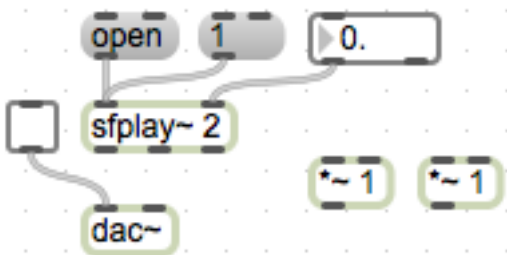
- b. Now that the [sfplay~] is connected to the [dac~] we can hear a sound file.
  - i. Press command E to exit edit mode. This allows you to use the patch without changing it.
  - ii. Click on the “open” message. This will open an open dialog window. Find a sound file you would like to play, and then click “open”. For the purposes of this demo, you can open the “demo1.aiff” found in the Max Demo folder.
  - iii. Click on the toggle to turn the [dac~] on. Now the toggle should have a large “X” inside of it.
  - iv. Click on the “1” message. You will hear the file played back over the speakers.
  - v. Once a file has been opened, [sfplay~] remembers that file and will be ready to play it anytime you click on the “1” message. If you want [sfplay~] to play a different file, click on the “open” message again to find a new one.

4. Controlling Volume

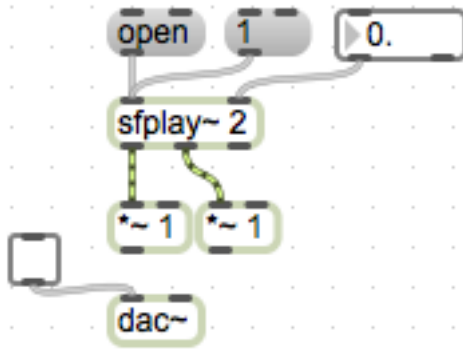
- a. First, some notes about digital audio.
  - i. Each sample (recalling sample rate theory) is expressed within the computer as a value between -1 and 1. These values are translated into electrical voltage, and sent to the speaker, where they correspond to rarefaction and compression accomplished through the speaker’s membrane.
  - ii. To manipulate volume within the computer (i.e., within Max), we can multiply the values of the digital audio by some factor.
  - iii. Multiplying every sample of a file by 0.5 results in the output being half as loud; multiplying by 2 results in it being twice as loud.
- b. To accomplish this operation within Max, the [\*~] object is required. This object performs a multiplication operation on each sample of audio that passes through it.
- c. Press command E to return to edit mode. Create two [\*~] objects, and provide them with arguments of 1. (The argument is optional, and only mentioned so that once they are connected there will be audible output).



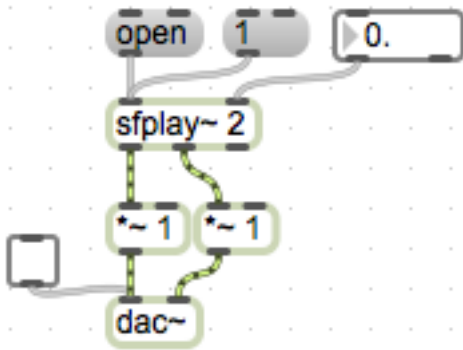
- d. Now delete the patch cords connecting [sfplay~] to [dac~] by clicking on each of the cables and pressing the delete key one at a time.



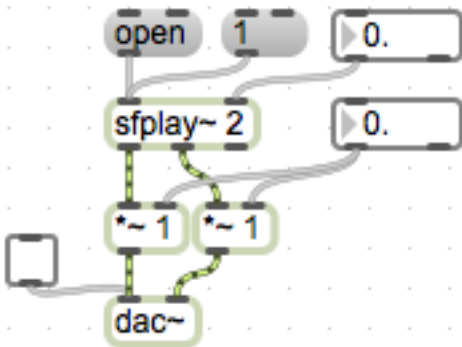
- e. Click and drag the [dac~] and the toggle to make room for our two new [\*~] objects. Connect the channel one outlet of [sfplay~] to the left inlet of one of the [\*~] objects, and the channel two outlet to the other.



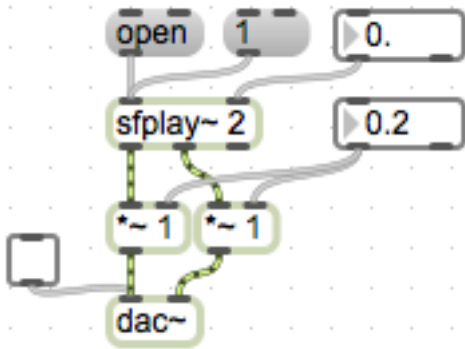
- f. Now connect the outlet of the left [\*~] to the left inlet of the [dac~], and right to right.



- g. For this demonstration we will control the [\*~} objects using flonum objects.  
h. First, create one flonum object and connect the left outlet to the right inlet of each [\*~], as shown.



- i. We can update the math happening in the [\*~] object by sending new numerical values into the objects' right inlet. The value that is set in the flonum object will be sent to each [\*~] concurrently, changing the volume of each channel by the same amount. So, in the following example, each channel will be heard at 20% of their original volume.



- j. To change the volume of the two channels independently, simply create another flonum. Attach the left outlet of one flonum to the right inlet of one [\*~]. In the example below, the left channel will playback at 20% original volume, while the right channel will playback at 90%.

