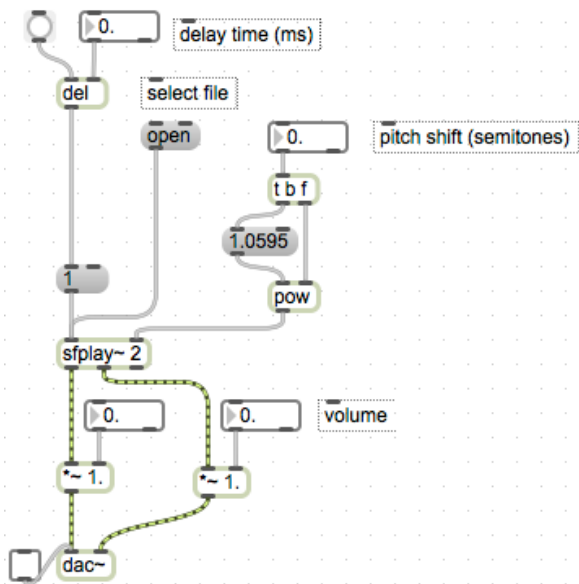
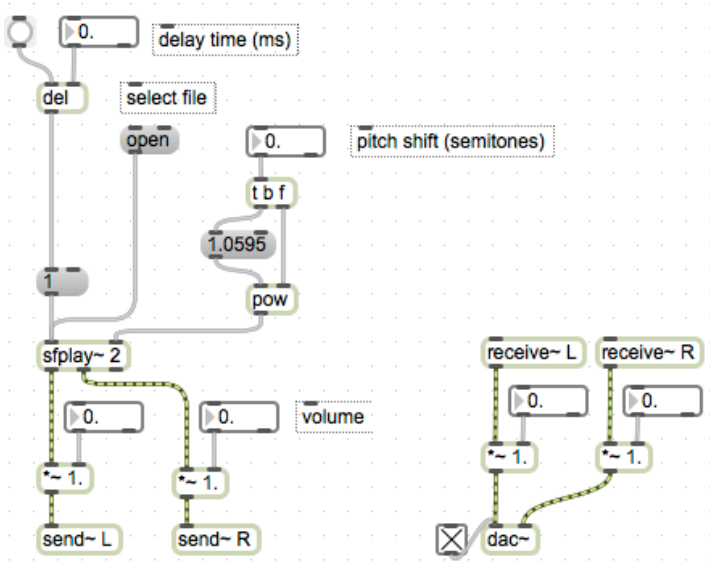


Shading in Max
Electronic Music II
Spring 2013

1. Concept of shading
 - a. Shading involves the combination of a main sound with one or more auxiliary sounds. The goal is to make subtle alterations to the attack/release of the main sound.
 - b. We can accomplish this with a basic `sfplay~` setup.
2. Since we want to have as much control over the parameters of each `sfplay~` as possible, we will use no randomized values.
 - a. The parameters we want to control are: delay before triggering playback, pitch shifting, and panning.
 - b. A single instance of this setup is pictured below.



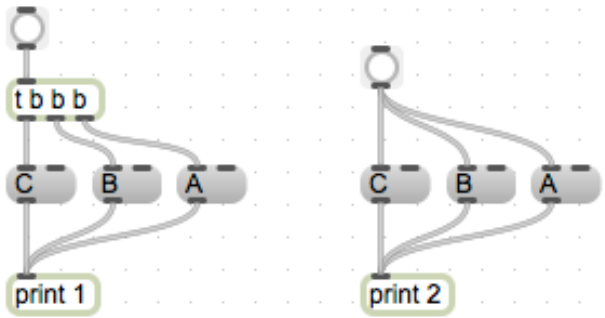
- c. Replace the `dac~` pictured above with `send~` and `receive~` objects, as detailed in past handouts.



- d. To achieve a minimum of shading, we need at least one more sound. As we add instances of `sfplay~`, remember that we need to control the total amplitude with `/~`.

3. Additionally, as we add instances of `sfplay~`, we need a way to trigger them simultaneously. At this point, an obscure aspect of Max's behavior should be pointed out.

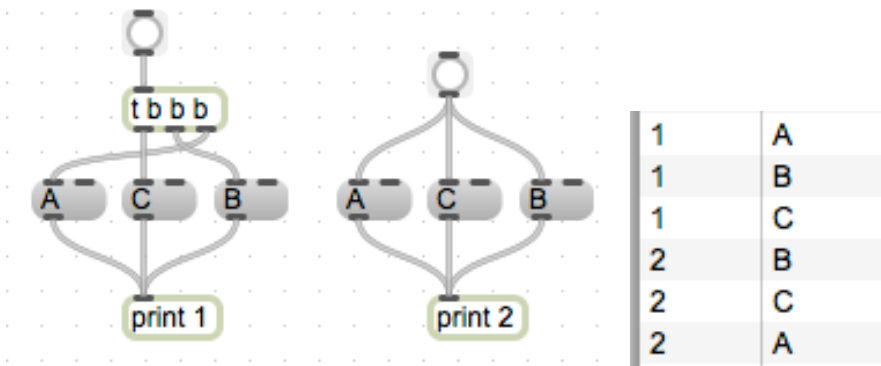
a. Consider the following patch:



b. If we click both bang buttons, we see that they give the same output (labeled 1 and 2 to distinguish between them).

Object	Message
1	A
1	B
1	C
2	A
2	B
2	C

c. In the process of patching, objects are frequently moved from their original orientation. Consider the patch now, after moving each "A" message left of the "C" messages, and its output pictured below:



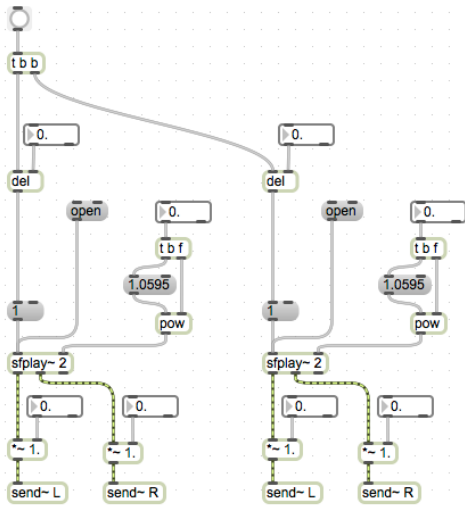
d. The output of series 2 is different, while series 1 remains the same.

e. This is due to the way that Max manages objects in a patch, in terms of their execution order.

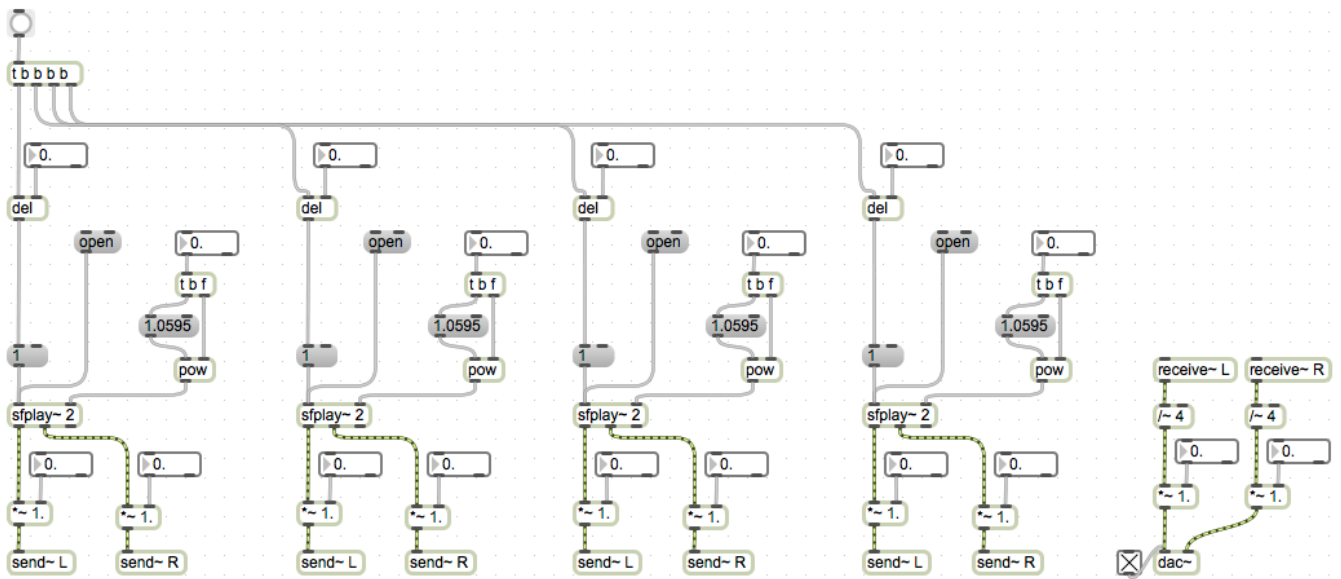
Execution moves from right to left. So, when I moved the series 2 "A" message left of the "C" message, the execution order changed from A-B-C to B-C-A, as reflected in the output.

f. However, the trigger object in series 1 preserved the original execution order. This is because the sequence of bangs coming from `t b b b` did not change, even though the geographical location of the objects **being** banged did.

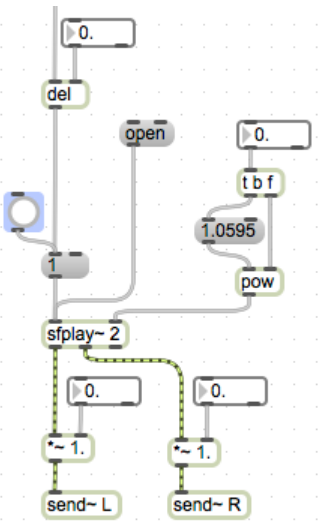
g. For this reason, I strongly recommend using a `trigger` object to trigger our multiple `sfplay~` setups, rather than connecting one bang button to each one in the series.



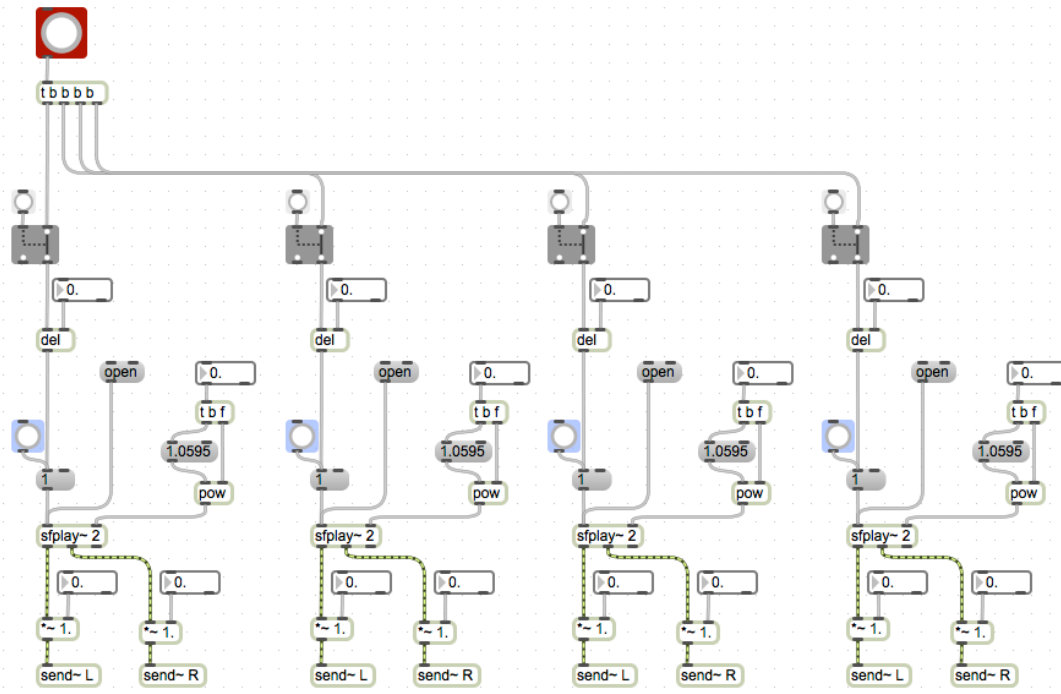
- h. Now that we have a basic shading setup, consider some of the relations between a main and an auxiliary sound;
- i. The auxiliary sound will typically occur just before the main sound. So, the delay setting of the main sound will be longer than the delay of such an auxiliary sound.
 - ii. The amplitude of an auxiliary sound is typically much lower than that of a main sound.
4. We can expand our basic shading setup in several ways.
- a. Adding more `sfplay~` setups; remember that this impacts the value of the `/~` objects in our output setup, as well as the number of bangs sent from our trigger.



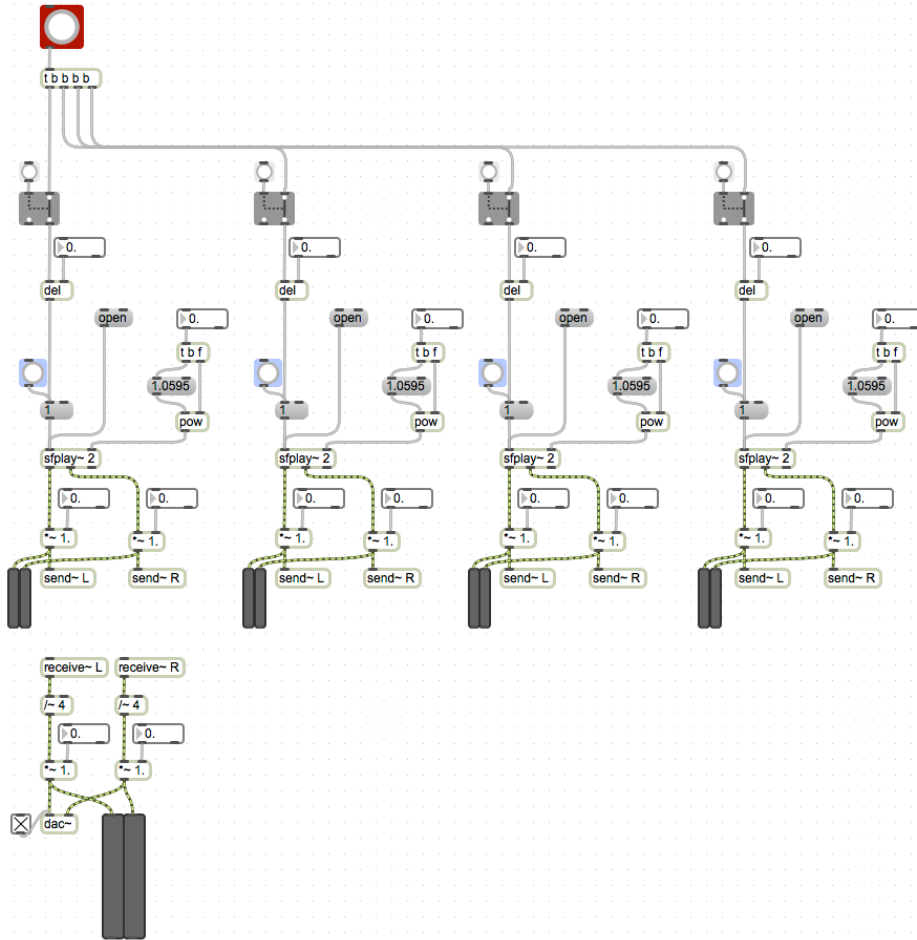
- b. To check each individual `sfplay~`'s output, I recommend adding a bang button connected to the "1" message in each setup. Notice that I change the size and color from a default bang button.



- c. In the event that we would like to use fewer than 4 sounds, we can avoid having to disconnect an `sfplay~` unit by using `ggate`. Notice again that I changed the color and size of the main triggering button.



- d. A convenient visual cue that audio is in fact coming from our objects can be added with the `meter~` object. This creates an LED-style VU meter that reflects the volume of signal sent to its inlet.



- e. Finally, adding an `sfrecord~` setup will allow us to use and edit the now-shaded output in other software (or in Max).

