## Composition: Electronic Media II Spring 2009 Building a Patch in Max/MSP

- 1. Power up the studio and bring up the faders to the Digi and Mac.
- 2. Launch Max/MSP by clicking on the Max icon in the dock, as shown below:



3. Select File>New Patcher to create a new patch. A window like the one below will appear:

| 0 0     | Untitled3 (unlocked)  |  |  |  |
|---------|---|--|--|--|
|         |   |  |  |  |
|         |   |  |  |  |
|         |   |  |  |  |
|         |   |  |  |  |
| Double- | lick to make a new object                                   |  |  |  |
|         |   |  |  |  |
|         |   |  |  |  |
|         |   |  |  |  |
|         |   |  |  |  |
| £ + +   | <b>7</b> I I <b>1</b> I II |  |  |  |

4. Double-click in the lower middle area of the window. A target like the one below will appear:



5. In the object palette, click on the object icon. An empty object object like the one below will appear:



6. Type "dac~", as shown below:



Notice the following:

- 1. "dac" stands for digital to analog converter. It converts digital sounds into analog audio sounds.
- 2. The  $\sim$  (tilde) sign indicates that the object is an audio or MSP object.
- 3. The dac $\sim$  object has a left audio inlet and a right audio inlet.
- 4. You can make the object box longer by grabbing the right side and resizing.
- 7. Using a similar procedure, insert a toggle object and patch it into the left inlet of the dac~, as shown below:



Notice the following:

- 1. The dac~ needs to be turned on and off. The toggle will accomplish this.
- 2. This is a segmented patch cord (see the **options** menu).
- 3. The patch cord is a plain, black line, which indicate that it is a control signal, not audio.
- 8. Using a similar procedure, insert an **object** object directly above the **dac**~ and patch it into its left inlet of the **dac**~, as shown below:



As shown above, type " $\sim 0.5$ " into the box. This is now called a "signal multiplier" object.

9. Patch the signal multiplier object into the right inlet of the **dac**~, as shown below:



## Notice the following:

- 1. The Signal Multiplier will be used to control the volume of the left and right channels of the dac~.
- 2. The patch cords are dashed, yellow and black lines, which indicate that these are audio signals, not control signals.

10. Using a similar procedure, insert a **float number box** above and to the right of the signal multiplier, as shown below:



As shown above, patch it into the right inlet of the signal multiplier.

Notice the following:

- 1. The Float Number Box will be used to supply a decimal number to the Signal Multiplier.
- 2. This decimal number will function as a volume control for the dac~.
- 11. Using a similar procedure, insert a **message box** directly above the **signal mulitplier** and type "cycle~ 440", as shown below:



Notice the following:

- 1. The cycle~ object is a sine wave.
- 2. The patch cord is an audio signal.
- 12. Play the patch as follows:
  - a) Lock the **Patcher**.
  - b) Click on the toggle.
  - c) Adjust the volume from the Float Box.
  - d) Press "stop" to stop the sound.

13. To learn more about the cycle~ object, unlock the patcher, highlight cycle~ and type cmd-shift-h. A window like the one below will appear:

| 900   |   | o cycle~.m   | axhelp  |  |
|---|---|--|---|--|
| <b>cycle~</b><br>Table lookup osci  | illator   |  | open cycle~ referenc  | e  |
| of 512 samples. Its<br>from a buffer~ obje<br>sample. For repeat<br>be no discontinuity<br>cycle~ assumes a s | default waveform is one<br>ct. The 513th sample in th<br>ing waves, it's usually des<br>when the waveform wrap<br>513th sample equal to the | cycle of a cosine wa<br>he wavetable source<br>sirable for the 513th<br>s around from the e<br>1st sample. This is | edly through one cycle of a w<br>we. It can use other wavefore<br>6 (the buffer~) is used for int<br>sample to be the same as th<br>nd to the beginning. If only 5<br>the case for the cycle~ objec<br>e 513th sample the same as | ns by accessing samples<br>erpolation beyond the 512th<br>e first sample, so there will<br>12 samples are available,<br>t's default cosine waveform. |
| 1. argument<br>sets frequency<br>(default 0)  | <ol> <li>Driven by phase only<br/>frequency input<br/>(float or signal)</li> <li>cycle~</li> </ol>  | y (frequency is 0):<br>phasor~ 660<br>phase<br>input (float  | 3. Using frequency input<br>437<br>\$1 400<br>line  | 4. Using your function<br>p generate-buffer<br>buffer~ buffsky 11.61<br>loadmess 22.5<br>cycle~ buffsky  |
| ~ 0.2   | ~ 0.2   | or signal)   | cycle~  | ▶0. listen?  |
| Symbol argument<br>buffer~ to play fro<br>argument after the<br>offset into the buf                           | at sets sample  | start<br>audio   | *****   | See Also:  |
| <b>•</b>  | 🖗 i 🖪 🗄   |  |   |  |

Note the following:

- a) Most objects have similar help windows.
- b) Items from a help window can be unlocked, copied, and pasted.
- c) Each item from a help window also has its own help window, accessed by typing cmd-shift-h as above.
- d) Beware of running multiple dac~s.
- 14. How about controlling the frequency from a number box?
- 15. Save, name, and store the patch, then quit Max/MSP.