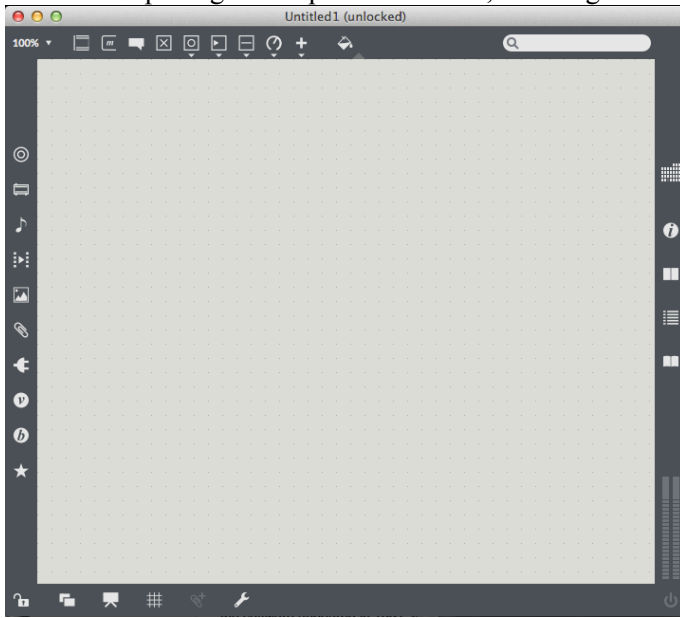


## MUS:4251 COMPOSITION: ELECTRONIC MEDIA II

Spring 2016

### Max Introduction




1. In this handout we will begin our discussion of Max with an examination of the basics. We'll examine the layout of the program, learn how to make connections between objects and build some interfaces.
2. Background: Max was created while Miller Scott Puckette was working at IRCAM in the mid-1980s. French composer Philippe Manoury was the first composer to write a piece with this program, called *Pluton* (1988). Later, Miller Puckette created a signal processing addition to Max when they were work on the IRCAM Signal Processing Workstation (ISPW), which was first called "Audio Max". This later became known as MSP (max signal processing), which was first released in December 1997. Miller Puckette left Max in the 1990s to create a free and redesigned computer program in 1996 known as Pure Data (Pd).
3. When opening a new patcher in Max, we are greeted with a blank workspace.

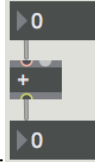


4. Let's examine the different buttons that are present on this workspace. Let's start with the top row. The top row allows you to insert many of the essential items we'll be using in Max, such as objects, messages, comments, buttons, and sliders. The left column has additional items that can be added to the workspace. Perhaps the most useful is the object library, which is an index for the standard objects that come with Max, MSP, and Jitter. You can also use the search filter to find a particular object. Jitter refers to a selection of objects that are used for visual processing. We'll leave those alone for now.
5. The right column has a couple of useful resources. For example, the second button from the top opens the inspector. The third button is a reference source that can be used whenever an object is highlighted. The fourth button is for the Max Console. Errors within the patched are logged here and can be accessed for troubleshooting. You can also send messages to this console using an object called "print".
6. The bottom row on the workspace has a very important button.



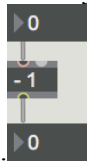
- The lock on the far left can be used to lock and unlock the patch whenever you want to interact with or edit it.
7. Finally, in the lower right corner, there is a meter with a fader and a button.

8. The button turns on and off the dac. The fader adjusts the master volume control.
9. Now, having discussed the basics of the workspace for Max, let's begin by inserting a couple of basic objects into the workspace and stringing them together.
10. First, let's select an object. A blank object when added to a workspace looks like this.
 
11. Click the first button the left at the top row to get an object.
12. A blank object by default contains two small circles on the left side, an inlet and an outlet. The inlet is on top, and the outlet is on the bottom.
13. Now, let's grab a number object. It looks like this:  Go to the 6<sup>th</sup> button from the right on the top row. You'll get a dropdown menu with three options. Click on the first button twice to get two number objects.
14. With objects, we can insert text to give them specific functions. Let's double click on the object and put in a +. Press enter and the object should now look like this. 
15. What we want to do now is to string both number objects to the + object. Take one of the number objects, click on the bottom-left half-circle, and drag to get a patch cord. Drag this patch cord to the left inlet of the + object. Then, take the other number object, and connect the inlet of the number object to the outlet of the + object. If done correctly, our



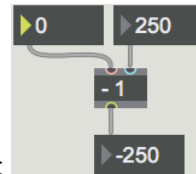
patch should now look something like this.

16. Go to the bottom of the workspace and click the lock to lock the patch. Once you've done that, move the mouse to the top number object, then click and drag up or down. Watch what happens.
17. Let's edit the patch. Unlock the patch. Change the + object to a - 1 object. If you do



this correctly, the patch should now look like this.

18. Repeat step 16. Watch what happens. Notice the bottom number object is higher than the top number object. This happens as a result of giving the - object an argument. We can describe an argument as a parameter that defines how the object is going to operate.
19. Let's experiment now by adding another number object. Let's set this new number to 250. We'll take the outlet of this new number object and connect it to the right inlet of



the - object. If done correctly, it should look like this:

20. Now, repeat step 16. Go up or down on the number object feeding into the left inlet. Watch what happens. Try this with the number object on the right. Watch what happens.
21. Notice that the number object on the right is sending information into a cold inlet. Information going to a cold inlet is stored. The number object on the left is sending information to a hot outlet. A calculation will be performed immediately.
22. Repeat step 20 with a \* object or a / object. Watch what happens. Try some other basic arithmetic functions and see what happens.