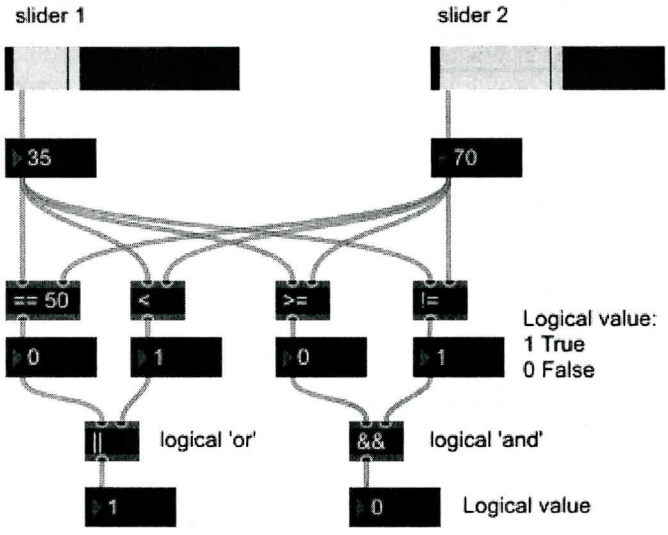
Composition: Electronic Media II
Spring 2015
Max 6: If… then … else …

1. Logical objects in Max
   a. In Max a logical statement that is true is evaluated (output) as 1 and a logical statement that is false is evaluated as 0.
   b. Logical object in Max are:
      i. [==]
      ii. [<]
      iii. [<=]
      iv. [>]
      v. [>=]
      vi. [!=] (not equal)
      vii. [&&] (and)
      viii. [||] (or)
   c. Observer the example below
      i. This patch compares the values of slider 1 to the value of slider two, showing the output of each of the logical objects.
      ii. For the [== 50] object, notice the '50'. Logical objects can take an initial argument for comparison. If the second slider is not set the [== 50] object would check if slider 1's value is 50. When slider 2 becomes active the value of slider 2 overrides the 50.
      iii. Each logical object has two inlets for comparison. The left and right inlet both take values and will be evaluated when a bang or modification occurs on the left inlet.
      iv. For [||] (or) note that it outputs a 1 (True) when either of its inputs are 1.
      v. For [&&] (and) note that it outputs a 1 (True) only when both inputs are 1.
   d. Note: the logical operators ('==', '>', etc.) trigger some objects in Max to output logical values instead of numbers, such as [expr]

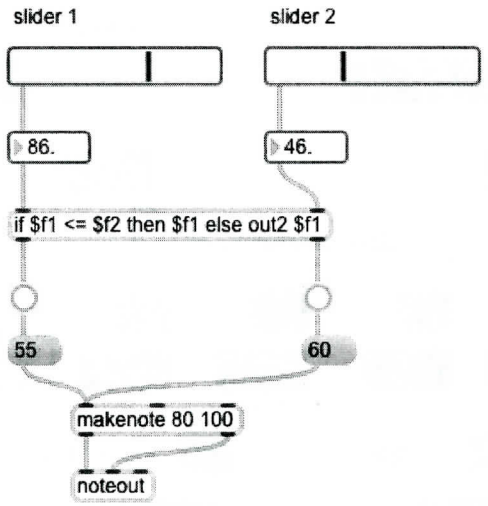


2. Specifying inlets
   a. For some objects you need to indicate how many inlets you need. To do this we use a '$', type of input indicator (i, f, or s), and an input number (1 through 9).
   b. Examples:
      i. $f1 flonum argument will be input to the leftmost inlet
      ii. $i3 integer argument will be input to the 3rd inlet (counting from the left)

3. The object [if …] evaluates logical expressions and can send out different outlets depending on the value of the expression included after the "if".
    a. Format is as follows: [if <expression> then <value> else <value>]
    b. <expression> represents a logical expression such as:
        i. $f1 > 5 (the flonum of inlet 1 is greater than 5)
        ii. $i1 == $i2 (the integer of inlet 1 is equal to the integer of inlet 2)
        iii. Spaces should be included before and after the logical argument (ex: $i3 <space> != <space> $i2)
    c. If the expression is evaluated as true then whatever value follows the "then" in the [if …] object is sent down the outlet
    d. "else" is an optional keyword where if the expression is evaluated as false then whatever value follows "else" will be sent down the outlet

4. The "out2" keyword designates a second outlet from the [if …] object. When placed after "then" or "else" it will send the value following "out2" out the second outlet.

5. Observe the example below.
    a. This patch acts as a comparison between slider 1 and slider 2. Depending on the values of the two sliders a different pitch will be sounded
    b. The [if …] object here has an expression of "$f1 <= $f2" which will evaluate to true if the first flonum inlet is less than or equal to the second flonum inlet.
    c. If the expression evaluates to true then we pass the value of the first inlet out the left outlet, which triggers a bang and we hear MIDI pitch 55 (G-4)
    d. Notice "out2" follows else; meaning that if the expression evaluates to false the then first inlet flonum is sent out the right outlet, triggering a band and we hear MIDI pitch 60.



6. Nested [if …] objects
    a. Often we want to deal with more than one check, such as we want to see if a number is lower than 4, higher than 9, or between 4 and 9. The following example illustrates how we can combine [if ...] objects to achieve just that.
    b. Observe the example on the next page
        i. By pressing the message [1] we generate a random number from 0 to 127 and we compare this number to the values of the two sliders. If the random number is less than the lower bound slider we hear MIDI pitch 55, if the number is greater than the upper slider we hear MIDI pitch 65, and if the number is in the range between the sliders we hear MIDI pitch 60.
        ii. The multislider on the right is used here as a visual of where the random number falls in comparison to the slider range. The top slider indicates the random number

iii. The random number is sent to the first [if …] object where it is compared to the lower bound slider. If the number is less than the lower bound then it is less than the interval and we heard MIDI pitch 55.

iv. If the random number is greater than the lower bound slider the [if …] evaluates to false then we evaluate "else out2 $i1" code which means we are sending "$i1" (the random number) out the second inlet, to the second [if …].

v. The second [if …] provides the comparison of whether or not the random number is between the sliders or higher than the sliders.