MUS 4711 –
Interactive Computer Music

Course Lecture Notes

MODULE 3: Interactivity

PART 11: Mousestate, and Coll

Objects – [mousestate], [screensize], [coll], [translate]

We're looking at alternate ways to get data into MaxMSP this week, focusing on using a mouse, human interface devices (like a joystick), and OSC (open sound control). Once you have data in, it's terrifically easy to map to the values to MIDI or other parameters using [sel] or [scale]. However, some data, like OSC is already normalized as floats from 0 – 1, and may not need that extra step. Always check what you're taking in with [print] or the cold inlet of a message before you start patching any further!

Mousestate – Pretty self-explanatory. [mousestate] is set to poll every 10 milliseconds and report the XY coordinates of the mouse, which drive a cursor in a [pictslider]. 0,0 is the Top Left corner of the screen. But we might not *know* the screen size in advance. To get around this, we can check by using the Mousemeter from the Extras menu, then enter the data in the appropriate [scale] inlets. But people's resolution might differ, so a safer way is to send a [loadbang] to [screensize], which outputs the screen resolution. These are sent to an [unpack], then the X Max is sent to the High Value of the [scale] for X values, while the Y Max is sent to the Low Value of the [scale] for the Y values. This makes Y's values 1080 – 0, which inverts the axis so that the Y coordinates now start from the bottom of the screen and not the top, giving us a 0,0 position of Bottom Left.

Keep this in mind when using [mousestate] or other devices – you may have to invert the X or Y axis to get values in a range that the user will expect!

Analysis of the Max Cookbook's *instatheremin* patch

Coll Twinkle – This may be the most important object of the semester. [coll] is a collection. It takes a single argument, the name you're associating with it. The input is the index number (any integer, I tend to stick with positive values though), and it outputs that associated line. The syntax for a line is:

X1, a b c d e f… z;
X2, a b c d e f… z;
X3, a b c d e f… z;
…

…
XX, a b c d e f… z;

Where X is the index number followed by a comma, the rest of the values are ANY TYPE OF TEXT DATA – INTs, FLOATs, SYMBOLS, etc. in a list that ends each line with a semicolon. If it doesn't have this syntax, you will get errors.

To use a [coll], you can load any text file (.txt extension, and it MUST have that extension, or [coll] will not like it). To enter values, you can either double click the [coll] to view its contents/enter data (follow the formatting, then save as a .txt file with COMMAND-S), or do it in your favorite text editor (I recommend VS Code). This text file must be in the same directory as you save the patch in. Here, I've got a [metro] driving [counter] to step through the values (0-41) of the MIDI notes in *Twinkle* on the left side. On the right, a [random] selects a random value from the [coll] that outputs a symbol in MaxMSP's Note Values format (see the chart). This must pass through the [route] to strip the (symbol) message from the front, before it goes to the [translate] object. The arguments here tell [translate] to, well, translate from MaxMSP's Note Values into MS. This then sets the time of [metro], so we get a slightly drunken rendition of *Twinkle* using the default MIDI synth.

Keep in mind though that [coll]'s can store ANY number of values on a single line that are output as a list. All you have to do is format the [unpack] or [unjoin] message accordingly. We'll take a look at that in my patch for *Needle Point*, where each [coll] is storing a value and a ramp time for every line. You'll also notice that the index numbers are only for cues actually played (there's 119 total!) – if it's not getting an index number that is used, [coll] doesn't output anything. VERY useful as you use it to structure a patch.

Analysis of [coll] use and structuring in *Needle Point*.

Max Puzzle 7 – Using two [coll]'s create a patch that will play a simple children's tune **other than *Twinkle*** (*Hot Cross Buns, Three Blind Mice, Frere Jacques, Go Tell Aunt Rhody*, etc.). One [coll] will control pitch, another will control rhythm. Then, for a slightly harder version, do the same song using only one [coll] (remember to unpack the output!). In both cases, they will send data to a [makenote] and through a [pack] to a [noteout] to play using the default system synth. **HINT – remember to use [translate] to convert notevalues to MS**…