MUS 4711 –
Interactive Computer Music

Course Lecture Notes

MODULE 3: Interactivity

PART 13: Sequencing, VSTs, OSC and IanniX

Objects – [live.step], [vst~]

Sequencing with [live.step~] and [vst~] – A lot of people want to create a step sequencer in MaxMSP – it's pretty easy with [coll], but it's even easier to use the [live.step] step sequencer. [live.step] can support multiple sequences of up to 64 steps with different velocities, pitches, durations, and two "extra" categories that vary depending on how you use them. To display/edit, you use the (display_seq $1, target_seq $1) message, sending it an integer to specify the sequence. (nstep $1) sets the number of steps, and the all-important message to drive it is (target_seq 0, next), which sends tells ALL sequences to export their next step in sequence.

We'll run that data from the left outlet of [live.step] through a [zl.slice 1] to strip the step number, then a [zl.slice 3] to strip all but pitch, velocity, and duration, then [unjoin] to format those in a [makenote] then [join] again before hitting [midiformat] where it's joined by the data from the rightmost outlet of [live.step]. This sequence index is a great proxy for MIDI channel data, and that's exactly how we're using it in [midiformat] to prep a message for [vst~], which hosts a VST plugin (use the plugin list from the left toolbar to find one, then drop it on the [vst~] object).

The right outlet of [midiformat] (MIDI event messages) goes into the hot inlet of [vst~]. THIS IS THE ONLY WAY TO USE [vst~]. The arguments of [vst~] specify the number of audio channel returns to MaxMSP (yes, some are more than stereo). Feed those to a [gain~] and [ezdac~] as normal, or to a [vst~] hosting some effect processors.

IanniX (Uses patches 02 and 03) – IanniX is a graphical sequencer based on the work of Iannis Xenakis (and funded by the French government) that allows you to define and trace paths (called curves) in three-dimensional space with cursors that output their XYZ position on the curve, triggers that are triggered by a cursor, and the ability to loop cursors in a number of ways. All of this data is sent out over OSC (and can be configured in a number of ways) from the score file you create in IanniX. Here's how to use it.

Open CONFIG and go to NETWORK. Make sure that OSC output is enabled, and that the DEFAULT IP (ALIAS IP_OUT) is 127.0.0.1 with a port of 57120. This is the default and *should* be there.

Turn on Snap to X and Snap to Y. Create a curve with cursor at 0.0 for a desired length/shape. Select the cursor (a full box will start up around the curve as well), and select INFOS. Under Time, set it to loop – either forward (1 – Loop on Curve), or roundtrip (1 -1 – Loop Roundtrip on Curve). Note the ID for the cursor… that will be important. Using the IanniX explanation demo patch, make sure that you're getting the correct data from the right cursor. In IanniX, select a cursor and hover your mouse over it to see the ID number. You can also select OBJECTS and click individual items in the list. If you're only creating curves/cursor pairs, cursors should always be on the even channel. Finally, to get a perfectly scalable range of 0-1 with no gaps, select the cursor and under INFOS -> MESSAGES, change it to "Mapped on Curves Bounding Rectangle." If you want a slight gap, leave it on the default "Mapped on Curves Bounding Rectangle Including Cursor Size." Test this in the IanniX explanation patch, setting the cursor ID with the number box as needed.

In MaxMSP, we're going to have a [udpreceive 57120]-[route /cursor] then route by the number of the cursor and [unpack f f f] to get the cursor group ID (which we'll discard), then the X and Y coordinates, ignoring the rest. But what to use it with? How about a [vst~]?

This time, don't create a [vst~] object. Just grab it from the plugin menu on the left toolbar and drag it into a patch. This will look a little different in that there is now a list of parameters with names/sliders/values that you can tweak. Click the wrench icon to open the instrument and interact with it. Start the IanniX sequencer, and send some values to the VST by sending a (X $1) message, where X is the number of the parameter you want to send the data to. In this format, it's all normalized 0-1, so there's no other conversion necessary. Go nuts! You can also send MIDI notes to the VST from MaxMSP by creating a [midiin]-[midiparse]-[midiformat] chain (hot outlet to inlet in each case), then taking the right MIDI Event output of [midiformat] and sending that to the left inlet of the VST. You could also use the [live.step]---[midiformat] chain the way we did before…

Max Puzzle 9 – Here's a doozy. Create four forward/backwards looping curve/cursor pairs in IanniX. Round trip for the cursors should be between 45-60 seconds. Send their data to MaxMSP via [udpreceive 57120]. Route accordingly to a [vst~] using the parameters viewable version (drag and drop it, don't create the [vst~] object). These should control 4-8 parameters of your VST plugin. Next, add a sequencer to control the MIDI going into the VST. No more than two channels of MIDI are needed. Include a [gate] and toggle so that the IanniX automation can be triggered on or off as desired.

Please limit yourself to Dexed, Helm, or Surge, so we all have access to them. Make sure you include the IanniX score file (.iannix) with your MaxMSP patch.