

Final Performance for FPA 247 at SFU

I prepared this performance using an iPod Touch, mrmr software, MaxMSP and Ableton Live. To get the samples I used, I recorded improvisations of me playing a metal kazoo and hitting and scraping a small metal box with the kazoo. I also used a couple of samples provided for the previous assignments. I then applied various processing on the recordings to achieve the types of sounds I have that vary in timbre, rhythm and other qualities.

I wanted to use the iPod Touch as a multi-function controller device and be able to fire up samples or modify the parameters of different processes or adjust the levels of different channels or sends. To communicate the iPod with the computer, I used the mrmr software. It uses Open Sound Control (OSC) protocol to send and receive data and it provides the application for the iPod, an application for Mac OSX to connect to the device, a template of interfaces in 4 banks (you can build your own interface as well but I thought the provided one does the job for me) and a Max patch that includes a template to get OSC messages into Max. This patch uses the OSC-route external object, which is developed and freely provided by CNMAT.

The provided patch gets OSC messages from the device, connected through an UDP port. These messages are in the form of numbers from 0 to 1000. In order to use the device as a MIDI controller, I needed to convert these messages to MIDI messages, so I have built a Max patch (mrmrTOMidi) for that need. This patch includes the template provided by mrmr with as many outlets added as the number of elements that send OSC messages as a sub-patch. Each element that is represented as a widget (button, slider, pad etc.) is assigned a ctlout object and while it is ok for buttons to have any value triggering the object, for sliders and pads the values between 0 and 1000 needs to be converted to values between 0 and 127. To do that I divide the OSC messages by 7.86 before sending them to ctlouts. Each ctlout has fields to set the controller numbers and MIDI channels. The MIDI channels of all ctlouts corresponding to a bank of widgets can be changed at the same time by 16 presets for each bank. This way each bank of widgets can control 16 controls per widget. To quickly access the presets for each bank I have built another sub-patch that makes it possible to use the computer's keyboard, using the hi object. This sub-patch listens to keys Q, W, A and S to set the active bank and an indicator shows which bank is active at the moment. The position of the rightmost indicator lit is the current active bank. Then MIDI channels from 0 to 10 can be set using the number buttons of the keyboard (I forgot that MIDI channels go from 1 to 16 instead of 0 to 15 so presets need to be re-assigned accordingly). A button (bang) on the patch initializes and activates the keyboard for use and another one deactivates it. The MIDI outputs of all the ctlouts can be set or changed globally from a umenu, labeled MIDI Out. One other feature I partially implemented enables the use of the first bank as a MIDI keyboard with a layout similar to piano. A toggle switches to this mode, directing the messages from the buttons in the first bank to a sub-patch. The problematic part of this mode is the duration of the notes to play. I tried various methods to figure that out but all have some problems so a modification of the sub-patch may be needed before use – or it may be considered as a slightly different type of instrument.

For the performance, I mapped the buttons in the first bank to audio clips I've loaded and setup in Live. I mapped the sliders in the second bank to the levels of different channels and sends and I mapped the movement of accelerometer in X and Y axes to the panning of two

different channels. Different channels are mapped to different controls so with the press of a keyboard button I could control a different set of parameters or fire up a set of different samples. This system allows me to control many things at once without having to use the mouse and this assists my creative process greatly. For example I can (in principle) fire up a sample and introduce it gradually with sliders, then send it to different effects with different amounts, all the while shifting the panning of two channels of sound. These all have to be done one-by-one when using a mouse.

However, I quickly found out that having a system at hand is one thing and mastering it and making use of its full potential is another. One of the things that I found quite hard to do is changing from one bank to another on the iPod. Mrmr application handles this by sliding a finger in a very tight area in the bottom of the view and it is often hard to switch back and forth to different banks without leaving too much time in between. This was a big problem for me since I wanted to control the exact timings of the samples I played because I don't like the way Live warps the samples to make loops. It even changes how the sample sounds and everything needs to be in quantized beats. It would have liked if I could quantize some clips and don't quantize others. For the iPod part of the system, I would have liked to have physical buttons on the side of the device to change the banks. Another thing that requires mastering the system is that nothing is labeled in the view on the iPod. It is possible to add labels if you build your own interface but then my system that changes the MIDI channels to control different things would not make sense and the space on the interface view is valuable. With the patch I have, I need to remember not only a number of controls that correspond to a current view, but up to 10 times that much. And since I can't normally see the patch while I am working on Live, I also need to remember which bank's preset control is active at the moment. While this provides some kind of function wise expanded, view wise compressed version of a monome-like clean slate approach, without perhaps weeks, if not months, of practice, it can get quite complicated very easily. For this reason I used the mouse to control as well. A specially assigned button on my mouse lets me switch back and forth between max and live so I can take quick peeks at the patch, which proved to be quite handy. Another handicap of the mrmr system is that it sends data only in one way – it only sends messages from the iPod to the computer but it doesn't receive anything back. Therefore, if you changed the position of a slider with the mouse, for example, the slider widget on the iPod doesn't get updated to show the current position. This often makes things very confusing. A more suitable use of this system, therefore, would be combined with a separate fader controller (preferably motorized, so that the issue I mentioned would be solved). Also, over all I think that using mrmr with iPod would be more effective if you have a simpler setup using only one bank at a time. In addition to the assumptions embedded in Live and Peak, which I mentioned in the previous assignments, being still in place, I can add that the mrmr-iPod control system and in fact any MIDI based control system, therefore all the DAW and plug-in software have the assumption that electroacoustic music is controlled through changing a bunch of parameters linearly and starting or stopping some processes. Further MIDI system also assumes the equal-tempered tuning system. While the tuning issue is obvious and there are ways to program or configure the systems to use different tunings, the former issue is more related to the underlying paradigms. It is probably natural that some way or the other controlling some parameters will be involved in any system that does processes based on those parameters, but things related to the way those parameters are controlled are probably based on mostly arbitrary choices made through the technological development. Most obviously the control

values of MIDI being between 0 and 127 is not a necessity and even though I don't have a solid alternative solution in mind right now, the linearity of the controls could probably be enhanced or replaced with new notions of control.

To create the audio clips I used as samples, I used a variety of processes, and worked in a similar fashion to the way I worked for the previous assignments. The techniques I mainly used include looping, reversing, delay-based effects including rhythmic delays and reverberation, equalizing, filtering, ring modulation, pitch changing, convolution and panning. Most of the time I worked with the editing program Peak and I also used Soundhack and Logic. Since I did not want to use the looping function of Live, I used the follow actions and asked for the sample to play again once it is done to loop my samples.

I grouped the samples I used into five different categories: long sustained sounds, rhythmic loops, rhythmic phrases without loops, usually high pitched ornamental sounds and percussive sounds. I had some loop samples in separate clips without being looped so that I can make quick references to those whenever I want. To get the long sustained sounds I mostly used pitch shifting several octaves and auto-convolution in different combinations – I convolved first, and then shifted the pitch and vice versa, and sometimes I applied either of the processes again on top. I used equalizing and filtering either to get rid of noise or other artifacts that came out as a result of the processes or make more subtler adjustments to the spectrum. For the looped sounds I again used pitch changing to get rhythmic sounds out of otherwise short and continuous sounds. The recordings of kazoo were very useful for this since the sound is made up of very fast alterations. The long beating loop heard throughout the piece was kazoo shifted 5 octaves down. I also used the same recording, shifted 2 octaves down to make a faster loop. For another loop I had I recorded a rhythmic sample placed in two different tracks but one starting later. The samples were identical so there was no phasing effect, and the rhythmic pattern created remained the same. I also had a deep percussive sound in loop, and a high frequency rattling loop, both of which I got by pitch shifting up or down several octaves. As I said I had some of the samples of the loops in separate clips and in different variations. The variations on these include delay, pitch shifting, equalization, convolution reverb and auto convolution. When I auto convolved a rhythmic or melodic sample, since (in effect) every sample was multiplied by every other sample, the resulting sound had a very interesting complex pattern. The other ornamental and percussive are achieved using similar techniques.

For the pitch shifting processes, reversing and convolution reverb I used Peak's built-in DSP features (change pitch-without preserving duration, reverse, and impulseverb) and I did some of the equalizing and filtering using the BIAS plug-ins that come standard with the software. I also did equalizing using the graphic equalizer plug-in of the GRM Tools and the Channel EQ plug-in in Logic. For convolution I used Soundhack and for the delay effects I used Tom Erbe's +delay plug-in. For the panning I used Live and mapped the panning of some of the sustained samples to one axis of the iPod's accelerometer and the ornamental sounds to the other axis. I planned to do more processes live with the resonator filters of Live's resonator plug-in, granulation and delay using Tom Erbe's +bubbler and +delay plug-ins. I planned to do the processing either by having the effects in return channels and controlling the send levels of each track or adding the effects directly to the track I wanted to process. I also planned to change the parameters of the processes during the performance. (During the performance things did not go exactly as I planned and I could not

do live processing. I am not sure but I think Live got confused when I tried to open up the plug-in windows while constantly receiving control inputs from the iPod.)

The piece turned out to be an improvised Musique Concrète or Acousmatic piece. If it were a fixed piece it would have been, at least in principle, similar to Pierre Schaeffer's pieces. However techniques I used like convolution and the aesthetic and formal aspects of the piece would prevent a direct resemblance. I also found the pitch shifting process I used, especially when with stretching the sound of the kazoo, similar to Hildegard Westerkamp's process in *Beneath the Forest Floor*. The essential difference I found between creating a fixed media piece and performing live is that when performing, everything needs to be done simultaneously. When working with fixed media, I can work on every aspect of the sound – timing, parameters of effects etc. – one by one and as many times as I want. During the performance I only have one chance of doing things right and most of the time doing everything one by one additively would result in a dull performance. Therefore, to perform live, a very careful preparation is necessary, figuring out when to do what or at least what goes with what, even though improvisation is involved in the piece. Creating a score or an outline of the piece would be beneficial for example. Mastering the control mechanisms is also necessary to control multiple things smoothly at once.