**Sound Library Technical Log**

**Concept and End-Users**

The basic concept for the ‘Drone Soundscape Tool-Kit’ sound library is to act as a library of samples for composition, video soundtracks or soundscapes and offer applied examples of the sampler application designed to work alongside the library. The library is to contain four different forms of sonic soundscape material: samples of acoustic instruments (mostly playing held notes and arpeggios focused on the key of G major), field recordings of ambient environments, synthesized pitched and unpitched drone sounds and examples of the sample playback and effects from the associated sampler application.

The application, created in MaxMSP, is designed to be versatile in its functions, utilizing the QWERTY keyboard as the sample trigger input: its most basic function is acting as a simple sample playback tool, with each sound triggered by their individual keys on the keyboard, but the application allows for a deeper and more generative playback tool due to the ability to change the amount of keys pressed before another sample triggers. This allows for the user of the application to be mostly unaware of what samples will be triggered. As an example: when set to small numbers this means that several chosen keys (with corresponding selected samples) can be repeatedly pressed to offer a random sample from the selection.

Alternatively, with higher numbers, an even more chance-based playback system comes to fruition: the application can work as an unconventional generative tool taking the generative data from the keys pressed. Due to the application’s ability to work as a background process, this allows for the possibility of using the device as a tool for creating a gradually changing soundscape whilst typing in any other software.

Furthermore, implemented within the application is a global pitch and speed effect: explained in more detail in the ‘Application – Patch Design’ section of this paper, its basic function is to take the numeric keys of a QWERTY keyboard and apply varying global speed changes to the playing samples, with ‘0’ triggering the regular playback speed and ‘1’ triggering the slowest playback rate available. Due to relationship between playback speed and frequency, this speed change effects the pitch of the samples and this function allows for both the distortion of the initial samples and the ability to manipulate the samples in more pitched musical context.

The sound library was created with three particular end-users in mind. The first are musicians of various calibers requiring soundscapes in a variety of forms. Possibilities include; drone artists requiring additional sounds, electronic (dance or otherwise) artists requiring an eclectic mix of samples for compositions, composers requiring background sound beds and so on. The versatility of the application, especially the global speed effect, allow for even greater sonic control over the samples.

The second potential end user, with focus on the application, is a general music consumer (pertaining to the drone/ambient niche). The application’s generative ability in sample playback can create ever-evolving soundscapes with hundreds of polyphonic possibilities, and with the QWERTY keyboard operating as controller the playback method is practically universal.

The library’s sounds, being entirely loop-able, have the potential to act as a sound bed for various video based projects, creating an atmospheric noise. This particular library’s use of looping drones allows for the visual mediums to have a potentially musical or atmospheric background without the need for in-depth musical composition, which can help reduce overall cost and time for projects or, at the very least, work to create a more stimulating product.

**Sample List**

Below is a list of the sound library’s samples, labeled with letters according to the application’s trigger. The design is such that the sample name indicates all necessary information about the sample without having to hear it, and the samples are broken into the four categories in order to further allow for easy browsing. Each sample is an uncompressed aiff, sampled at 44.1kHz with a 16-bit depth for maximum fidelity. Although it was considered to add duplicate samples at 48kHz for the possibility of video authoring, this was decided against as the supporting application would process the samples though the audio output accordingly, allowing for recording at the desired sample rate and preventing the sound library’s file size being too large.

Not included in the list below is the application examples, following the same encoding and compression as the other samples, included to display the functionality of the sampler and the effect of its sonic manipulation.

**A** – Violin Drone 1 - G and D – Effected

**B** – Lo-Fi Piano 1 – G major - Arpeggios

**C** – Acoustic Guitar – G major – Arpeggio – Effected with Delay

**D** – Field Recording 1 - Heavily Effected - Drone

**E** – Field Recording 2 – Train station - Reverb

**F** – Field Recording 3 - Thunderstorm

**G** – Field Recording 4 – Stream of Running Water

**H** – Synthesised Drone 1 – Sine Wave Synth – Random Pitches

**I** – Field Recording 5 – Street Ambience 1

**J** – Lo-Fi Piano 2 – Improvisation in G major – Heavily Effected

**K** – Acoustic Guitar 2 – Ebow Drone in G

**L** – Field Recording 6 - Street Ambience 2 – Slightly Pitch Shifted

**M** – Field Recording 7 - Machine hum – Slight Delay

**N** – Synthesised Drone 6 – Noise Synth with Distorted Notes

**O** – Field Recording 8 – Cars Creating Stereo Effect

**P** – Field Recording 10 – Heavily Effected Soundscape

**Q** – Synthesised Drone 2 – Layered Sounds

**R** – Field Recording 9 –London Ambience with Reverb

**S** – Synthesised Drone 3 – Emulating Rushing Wind

**T** – Violin Drone 2 – G and D – Heavily Effected

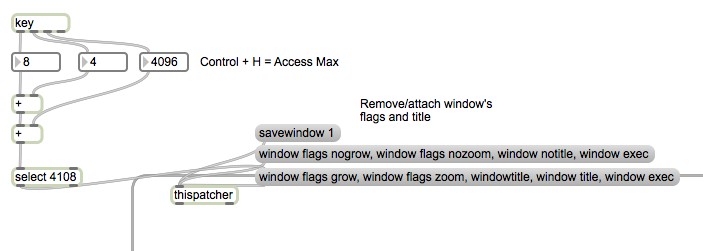
**U** – Vocal Choir Drone 1 – Heavy Reverb

**V** – Vocal Choir Drone 2 – Upper Octave – Pitch Shift plus Reverb

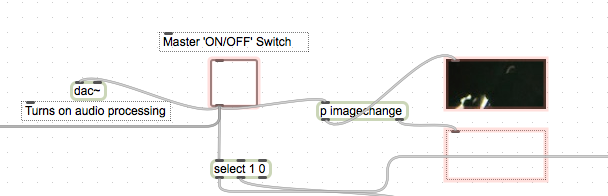
**W** – Synthesised Drone 4 – ‘Binaural Beat’ in G

**X** – Synthesised Drone 5 - Heavily Automated with Ringmod

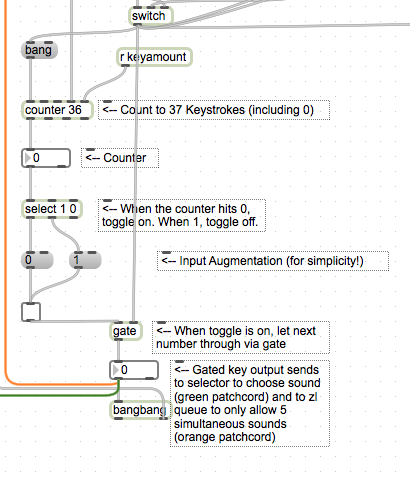
**Application – Annotated Patch Design**

****

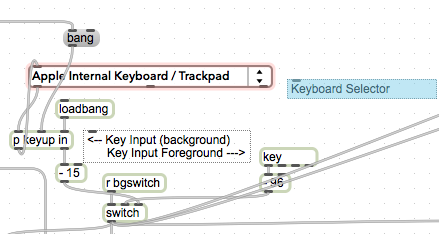
The above section of the patch simply allows for the application to have a more professional user interface; the ‘window flags nozoom/nogro/notitle’ message is designed to prevent the application window from being resized. As a debugging tool, pressing ctrl+h reverts these changes. This is also in the ‘Settings’ subpatch, to allow for a similar aesthetic.

****

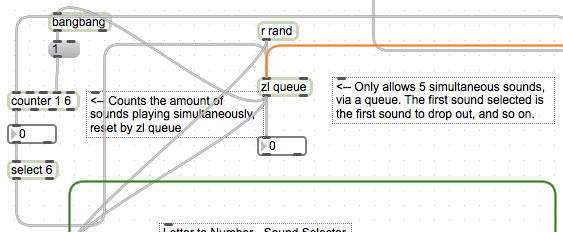
This section, the master on/off switch, works as an invisible button in front of the GUI graphic within the application. When on, the switch turns on audio processing, selects the inverted image and changes the status text to ‘On’ to graphically display the functioning application. The switch also turns on polling of the ‘human interface’ object, which receives the keyboard’s input.

****

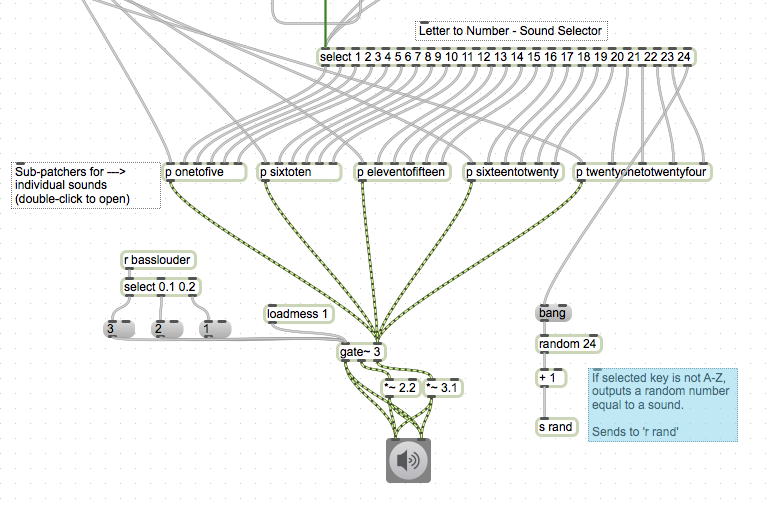
This area of the patch counts the keyboard input. The ‘r keyamount’ object receives the specific amount of keys required to press before triggering and is received from the settings page. When the counter reaches its maximum (numerically as 0), it triggers a gate which allows the ASCII number of the key pressed to pass through to the sound selection and to the queue – designed to only allow 5 samples to play simultaneously.

****

The keyboard selector here is visible in presentation mode and is required to allow the application to run in the background. If, however, ‘r bgswitch’ is turned on, this allows the patch to read the ASCII numbers from the ‘key’ object, bypassing background functionality. The ‘-15’ and ‘-98’ are required to standardize the numbers (A=1 etc.). The switches output sends the ASCII number from whichever keyboard option is selected.

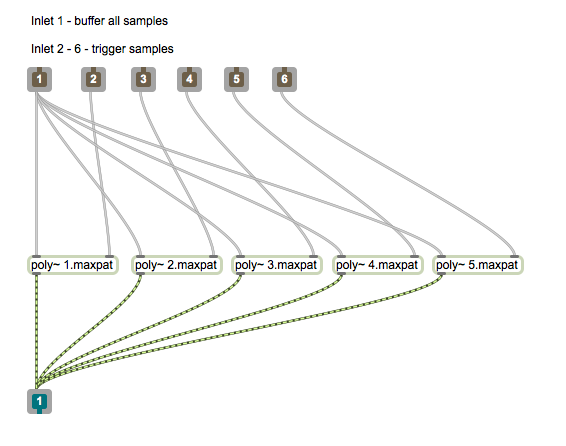
****

The orange patch cord sends each triggered key to the ‘zl queue’ object, which is of a ‘first-in-first-out’ system; the counter object sending a bang every 5 sounds and making the ‘zl queue’ output the first sound triggered.

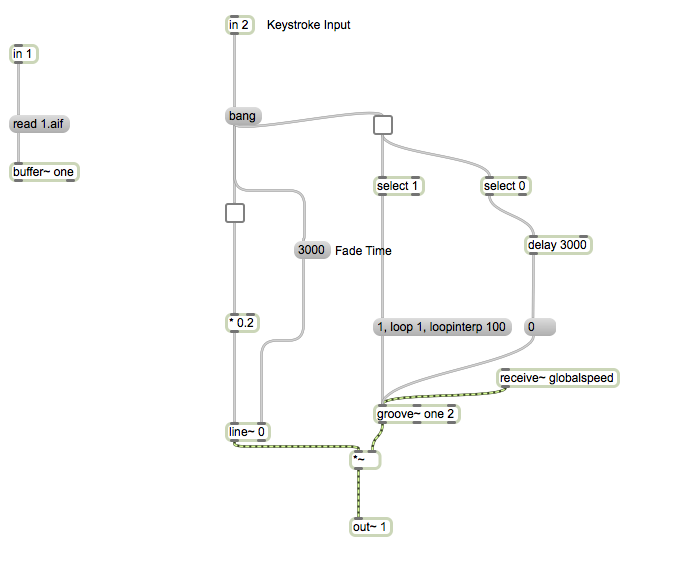
****

The green patch cord sends the information to the ‘select’ object, which routes the ASCII number to the individual sounds via grouped patchers (see ‘p onetofive’, ‘p sixtoten’ etc.). To turn a sound off, the ASCII number simply requires to be repeated: triggering a sample works as a ‘play/stop’ switch. This allows the ‘zl queue’ object to repeat the ASCII number sent through 5 numbers ago and switch the appropriate sample off.

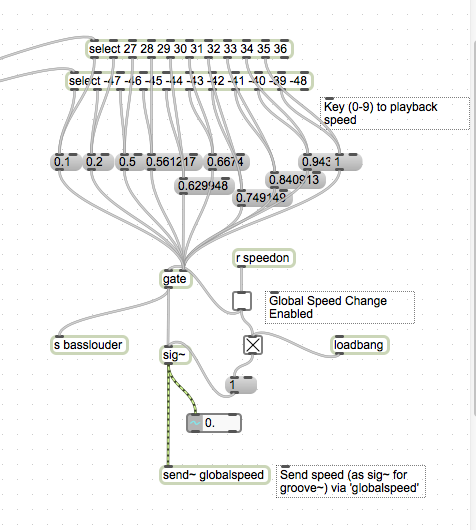
The ‘r basslouder’ object receives a message from the global speed control and brings to volume up accordingly when a low pitch sound is selected to make it more audible.

****

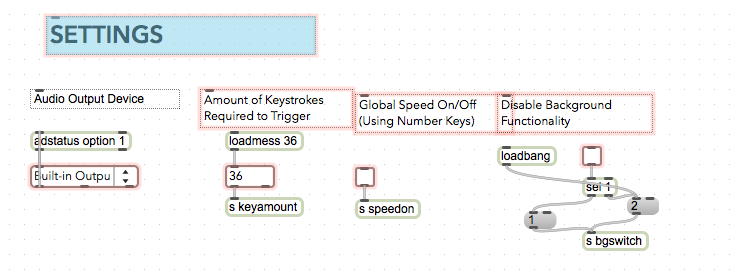
This is the internal design of ‘p onetofive’. Each one of the grouped numerical patchers is identical, save the number and the samples held within. Each ‘Poly~’ object is also identical, save the sample played.

****

This is the design for each ‘Poly~’ subpatch. Each patch has an individual ‘buffer~’ object, storing the required sample in the computer’s RAM, and the ‘groove~’ object is used for playback due to its diverse sample playback abilities. ‘Receive~ globalspeed’ allows the application to change the speed of all samples, and the ‘line~’ object creates a fade at the start and end of the sample when it is retriggered and turned off. The toggle, above both ‘select’ objects, simply toggles the sample on or off, and the ‘select’ objects turn the sample on, turn looping on, and delay the turning off of the sample to allow for the fade out.

****

The global speed section simply sends varying rate of playback to each ‘Poly~’ subpatcher, with 0-9 selecting the specific speeds. The speed change in question is dictated by the ratio difference between notes in the G major scale, essentially tuning the entire device to a key. The ‘Global … Enabled’ toggle is received from the settings page, and turns this device on or off.

****

The settings page consists of little else other than pertinent outputs to specified other areas of the patcher. The ‘adstatus option’ object allows for an audio device to be selected, and the subpatch is made visually appealing using the Max’s presentation mode, similar to the main patch.

**Word Count:** 1534