

GRANULAR SAMPLER – PIMP MY GRANNY

SOUND ANALYSIS SYNTHESIS AND PROCESSING

Giovanni Agosti 928366

GOAL:

The goal of this project is to implement a granular sampler that can be used in an actual musical/creative environment.

Most of today's DAWs already implement very sophisticated algorithms for granular synthesis for time stretching or repitching purposes but not so many of them offer an actual granular synthesizer that can be played by the musician.

Because of this, my purpose has been to realize something that can be played in a creative fashion and that could be flexible enough to implement a good (but not great) time/pitch-scaling/warping but that could be used also to produce sounds with a more “experimental”/noisy orientation, even regardless of some of the proper rules of time/frequency analysis.

The whole work is inspired by Curtis Roads “Microsound” and is delivered both in a Max MSP and Max4Live formats. I choose to implement this project in Max because I'm an Ableton Live user and using Max4Live is very straightforward to map parameters to hardware devices and to experiment with the resulting sounds already in a production environment.

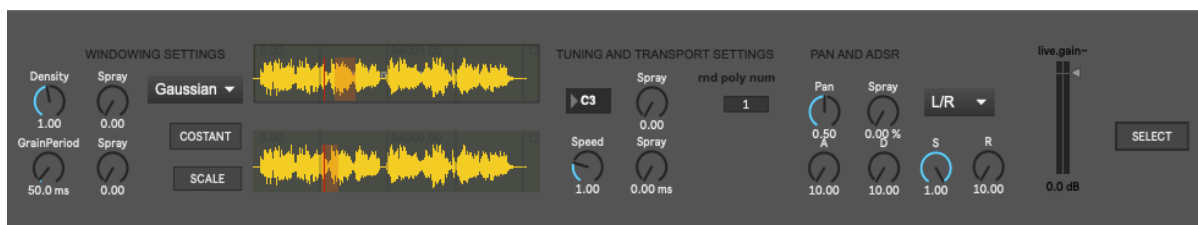
IMPLEMENTATION:

The core of the patch is included in the sub-patch *JOSingleGrainStereo* which exploits the functionality of the Max object *Groove* in order to generate each single grain (i.e. extract a window from the input signal in a specific place and with a specific length and then reproduce it at a specific speed).

This patch is then included into a poly object within the sub-patch *JOGranularStream* which is responsible for generating a monophonic grain stream (i.e. we can have multiple grains at the same time because of the overlapping but all of them are tuned/controlled by a single MIDI note at the time).

This patch is then again included into a poly object within the main patch *PimpMyGrannyFINAL* so that we can play a polyphonic stream of grains (i.e. we can play a triad on the keyboard and 3 contemporary streams of grains are generated).

GUI:



WINDOWING SETTINGS:

- DENSITY: with this parameter we can determine the density of the granular flux which corresponds to the reverse of the hop size. When density is equal to 1 the windows don't overlap, when density is equal to 2 the windows overlap of 50% and so on.

- CONSTANT/VARIABLE: constant keeps a constant density when transposing the note, adapting the length of the transposed window. For instance, windows of 50ms became of 25ms when transposed up an octave. When using constant we ensure to pick a 100ms window if we then have to speed it up, so that the resulting

sounds, when playing a chords, have all the same texture.

- WINDOW SHAPE: simply the shape of the window.
- GRAIN PERIOD: simply the window length (with no transposition).
- SPRAY: produces a random variation of the parameter next to it.

TUNING AND TRANSPORT SETTINGS:

- NOTE REFERENCE: determines the transposition of the whole keyboard in order to adapt it to the pitch contained into the sample.
- SPRAY(Pitch): random transposition.
- SPEED: determines the speed at which the selected portion of audio is scanned.
- SPRAY: random behaviour of the buffer pointer.
- RANDOM GRAINS NUMBER: this is a global parameter and determines how many random grains are generated starting from the same grain.

PAN AND ADSR:

- PAN: simply the overall panning of the outgoing audio.
- STEREO SETTINGS (L/R, L, R etc): determines which portion of the stereo image of the incoming audio signal is selected.
- SPRAY: random panning for each grain.
- ADSR: simply the adsr of the whole synth.

PRESETS:

1. ideal for time/pitch-warping/scaling.
2. similar to the previous setting but with an higher density.
3. rhythmic setting with big windows and with random panning.
4. similar setting but with pitch randomization.
5. setting with zero speed but with randomized buffer pointer.