

BLEEFMAN

Licence and technical information

Bleepman is virtual synthesizer programmed in Native Instruments Kontakt sampler, you need full version of Kontakt 4.2 or newer to make it work.

You are licensed to use this device and samples which come with it, in the creation of a recorded or live sound performance, free or commercial, without paying any additional license fees or providing source attribution.

Please DO NOT: include provided samples or script in any music library or sample library; sell, repackage or re-distribute the samples or sampler programs.

USE AT YOUR OWN RISK! This device is provided 'as is' and there is no warranty of any kind. It is experimental device, there is no guarantee it will produce any pleasant or useful results.

Bleepman

Bleepman is virtual device designed to bleep. It is based on single-cycle waveforms. The waveforms have been recorded from vintage analog synthesizer, some random old electronic gadgets, or generated digitally to resemble sounds of old video games. There are 40 waveforms to choose from, last two being noise variants.

Filters and effects

Bleepman is rigged with a selection of Kontakt's filters and effects. There is toggle switched filter, 2 pole low pass or Pro53, controlled by knobs: f-mode (toggle filter mode), cutoff and resnance. Then there is one pole hi-pass filter, controlled by hipass knob, which sets cutoff frequency. The effects are delay (delay, time, damping, pan, feedback knobs), custom lo-fi reverb (reverb and size knobs), tube distortion (drive knob), cabinet simulation (cabinet knob – for selecting the cabinet type, when fully counter clockwise, the effect is off; size, air, treble and bass knobs), lo-fi effects: bitrate and sampletate simulator – when set to fully clockwise, those effects are off.

Main instrument controls

There is pitch knob which sets instrument's master tuning. It can be automated to create pitch sequences or sweeps. Snap switch will make pitch controls snap to semitones. Learn switch function will be explained later.

There is standard AHDSR envelope, panorama and volume master knobs. Sensitivity knob controls the amount of keyboard velocity sensitivity, when set to fully counter clockwise, the instrument will ignore note velocity information.

Step sequenced LFO modulators

This is instrument's specific concept. There are four 8-step sequencers, step value applies to modulator range. Modulation can be send to one of instruments parameters, like: volume, pitch, panorama or filter cutoff. Modulation target can be selected from drop-down menu next to sequence graph.

You can pick one of nine modulation shapes. Modulation shapes are being calculated by the script and parameters are being set at defined time period. Modulation engine is controlled by rate, precision and sync knobs. Precision is modulator frequency, that is, how often parameters are being modulated. Setting a low precision will result in noticeable steps, making it one of the kind aliased LFO. E.g. 8 step ladder shape is approximation of linear shape at low engine precision.

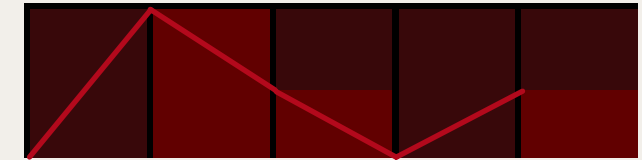
Engine rate is, how many frequency ticks it takes, to go from one step to another, effectively it sets sequencer speed. Sync button, will synchronize rate and precision values with host's tempo, when sync is on, rate knob will snap to note fractions: 1/32 note, 1/16 note, 1/8 note and so on.

You can edit sequence program manually or change it using one of automatic transformations - use trans button next to sequence graph.

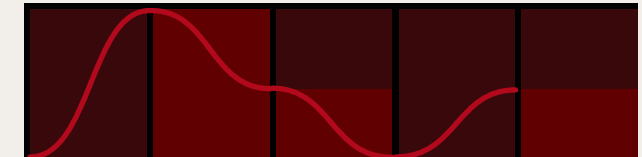
The whole modulation engine is taken from another device named Queuescape, where it was implemented on larger scale and in somewhat more complicated manner. If you find it interesting, check it here: <http://szcz1.bandcamp.com/album/the-sounds-of-szcz-vol-3-queuescape>

Here's the list of available modulator shapes with graphical presentation of how parameters are being changed.

linear



sinusoidal



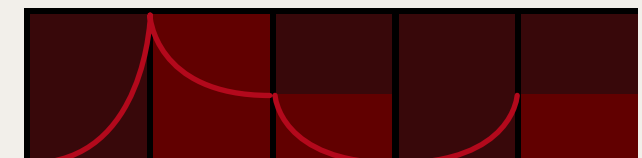
square



parabolic



inverse parabolic



linear x2 + hold



linear x8 + hold



linear x16 + hold



8 step ladder



List of available sequence transformations:

reset – fill sequence with default values,

cast – copy sequence data to other sequences, only unassigned (disabled) sequences will be written.

comb – increase every odd step value, decrease every even step value,

compress – compress sequence, decrease the amplitude of the whole sequence,

cycle – repeat a cycle of values over the sequence, according to cycle knob,

expand – expand sequence, increase amplitude of the whole sequence,

fall – decrease all values by the same amount,

flip – reverse the order of sequence,

flux – modify each value by a small random amount,

invert – invert sequence values,

level – will copy the value of the first step to other steps in sequence,

mirror – will copy first half of the sequence over the second half in reversed order,

multiply two – same as two step cycle, repeat first two steps over whole sequence,

random – fill sequence with random values,

rise – increase all values by the same amount,

slice – set every even step value to zero,

slide – slide selected sequence one step to the right, last step value will move to the first position,

slide by cycle – same as slide, but using many steps, according to cycle knob,

vector – will make transition between first and last value in the sequence.

Random modes and buffer

Each sequencer can be set to random mode with rand switch, in this mode the machine will write a random value to the next sequencer step each time sequencer moves. There is alternative buffered random mode, activated with r-type switch. In this mode the machine will fill a 64 step buffer with random numbers and then use them instead of continuously generated random numbers, creating a looped set of random data. Each sequencer has it's own buffer. You can limit the loop size with cycle knob, additionally you can use mod-wheel to control loop start position. So you can playback different parts of the loop while performing.

Shuffle button will fill all 4 buffers with random data, this is not un-doable, so use it with care. Buffers' content is being saved with patch or within preset, so unlike regular random mode, buffered mode produces repeatable results.

In buffered random mode sequence transformations will not apply to the sequence data, but to the buffer. This way you can manipulate larger sequences. Note that transformations will apply to a part of buffer, according to cycle setting (except for cycle and both slide transformations which always apply to the whole buffer).

Shadow voices (uni)

Bleepman has two shadow voices, labelled as uni 1 and uni 2. When shadow voice is active the instrument will generate additional shadow note on incoming midi note message. You can change additional note properties; assign different waveform, change panorama and pitch.

Shadow voice is being activated by uni knob, which is the shadow note volume [relative to main note velocity]. When uni knob is fully counter clockwise, shadow voice is disabled. Shift and fine knobs control voice pitch, shift will change tuning octave up or down and fine will detune the voice slightly.

Shadow voice properties can be selected as modulator target. Pitch modulation effect will be cumulative with shift and fine settings. Also note that shadow voice properties are relative to main instrument settings, so if you set modulators targets to instrument pitch and uni pitch, shadow voice pitch will be affected by both of them.

Main instrument voice can be disabled, by setting waveform number to 41 [fully clock wise]. This way you can create sounds using only shadow voices.

Monophonic mode and arp

Mono button will activate/deactivate monophonic mode. In monophonic mode, as opposed to polyphonic, when a key is being held, pressing another key will change the pitch of current note, instead of triggering a new note. Tweaking speed knob, will affect the speed of pitch sweep between notes. When releasing a key the pitch will return to the value of newest key – in other words, the machine will remember the order keys were being pressed and track them back.

Monophonic mode is dependant on engine precision. At high precision settings, the transition between notes will be smooth. At low settings, the pitch change steps will be noticeable.

Monophonic mode can be limited to a key range, by manipulating keyboard-range knobs. Changes will reflect on Kontakt's virtual keyboard, keys working in monophonic mode will be coloured cyan, while keys working in polyphonic mode will be in default blue colour.

Bleepman has a simple arpeggiator, which is extension of monophonic mode. If arp is activated and multiply keys in monophonic keyboard range are pressed, the instrument will sweep through all activated notes according to monophonic mode speed and engine rate. It will go to the next note each sequence step.

There are standard arp patterns, up, down, up and down, as played. Also there are several special arp modes. In random arp mode, the instrument will pick a random new note [from active notes] – current note is in the poll, so it is possible to stay on current pitch. In buffered random mode, random arp pattern is derived from first sequence buffer data – it will react to cycle setting and to buffer mod [mod wheel by default]. Uni split mode can be used to create keyboard split, monophonic key range will trigger main note waveform and the remaining keys will only trigger shadow notes. You can e.g. create a setup where you can trigger programmed shadow notes pitch sequence with one hand and play a lead melody in monophonic range with the other hand.

Snap patterns

Snap pattern is a simple kind of scale enforcer and enhancement of snap function. When snap is active and snap pattern is picked, the instrument will snap pitch only to semitones that are set as active in the pattern. Note that this only applies to pitch changes generated by sequencer (or arp in pattern mode) and doesn't affect actual notes being played.

To activate snap pattern, select change snap pattern from drop-down menu. The machine will pick a random preprogrammed pattern. Select disable snap pattern to turn it off – use all semitones for snap. If you'd like to pick a specific pattern, there is a hidden drop-down menu, somewhere under menu button. A pattern is represented by binary string of 12 digits. First digit is referring to C note and "1" means that the note is active in the pattern.

This mechanism is ported from Psychodrumma muzak generator, which you can find here: <http://szcz1.bandcamp.com/album/the-sounds-of-szcz-vol-2-psychodrumma>

Arp patterns

Arpeggiator has special "pattern" modes. In pattern mode arpeggiator modifies the note pitch according to a program, in similar way as sequencer does, taking newest note as root pitch.

Arp pattern cannot be programmed directly, you can generate new set, using menu command, generate arp patterns. Arp pattern size can be controlled with cycle knob (regardless of r-type setting), also it responds to buffer mod (so you can change pattern read position with mod wheel). Arp pattern always snaps to semitones and snap pattern, regardless of snap switch state.

Numeral pattern mark "1" or "2" indicates pattern range setting – for "1" pattern pitch range is 1 octave, for "2" it is two octaves. "S" mark designates special mode. In normal mode, pattern always takes newest "note on" message as root note, while in "S" mode root note is always C. Another difference is that in "S" mode, pitch sequence modulation overrides pattern modulation for shadow notes, while in normal mode modulations are cumulative. Also in "S" mode pitch sequence root note will be rounded to nearest C note and it will stay on the same note until all keys are released. This way, in "S" mode, all pitch modulations stay within current snap pattern. Note that main pitch modulation target is, in fact, the whole instrument tuning modulator, so unlike uni pitch mod, it always affects whole instrument.

Midi learn functions

You can program pitch sequence using midi keyboard. To do so press learn button. You will notice change in colour scheme on Kontakt's virtual keyboard, now you can enter data pressing green coloured keys. This function will apply to first (from the top) sequence with target set to pitch, or to first sequence (in no sequence is assigned to pitch mod). To exit sequence learn function, press learn button again or finish entering data.

As you might have noticed Kontakt 4 doesn't support automation of data tables, which are used in Bleepman to program sequences. If you'd like to program sequences with hardware controller, there's is a script based workaround. Click in the empty space right under desired step controller, there is a drop-down menu, pick midi learn command to assign midi CC message to sequence step value. Learn button will be flashing, until a CC message is received. To cancel learning mode, press learn button. Unlearn command will delete the assignment for selected step.

It is also possible to re-assign buffer mod controller, which is mod wheel by default (this assignment is done via script, so you can't just delete it from automation table). Use buffer mod MIDI learn command from drop-down menu, now fiddle with new controller to assign it to buffer mod, this operation will clear mod wheel functionality, which can now be used for something else.

Reset MIDI automation command will clear all script based automation, that is restore default mod wheel functionality and clear all sequence step assignments. Automation will be saved when saving project in host or when saving Kontakt patch.

If you are using a sustain pedal you can notice that when you use keyboard range with monophonic mode, sustain pedal functionality will only apply to monophonic range. This is not a bug, but a feature. Also it is possible to change sustain pedal polarity on instrument level, using invert sustain pedal polarity command.

Automatic sequence generation

Bleepman has sequence generation function. It is activated by generate sequences command from drop-down menu. This function will set machine in buffered random mode and fill sequence buffers with random data specially profiled according to sequence target. This function will only affect sequences with set targets, so before executing it, be sure to pick modulation targets. If you want to preserve a sequence, you can temporarily set it's target to disabled, before calling sequence generation function.

Preset system

Bleepman has functionality of saving/loading preset files - use save settings / load settings commands. Preset files are stored in "data" folder by default, but they can be saved elsewhere. The instruments comes with several example presets.

Most of instrument parameters are being saved within preset, including sequence random buffers, arp patterns and snap patterns. Only settings that are not saved within presets are engine parameters: rate, precision and sync as well as script automation assignments.

This document ends here. Have fun.

Szcz @ 2014

www.failryconfusing.net