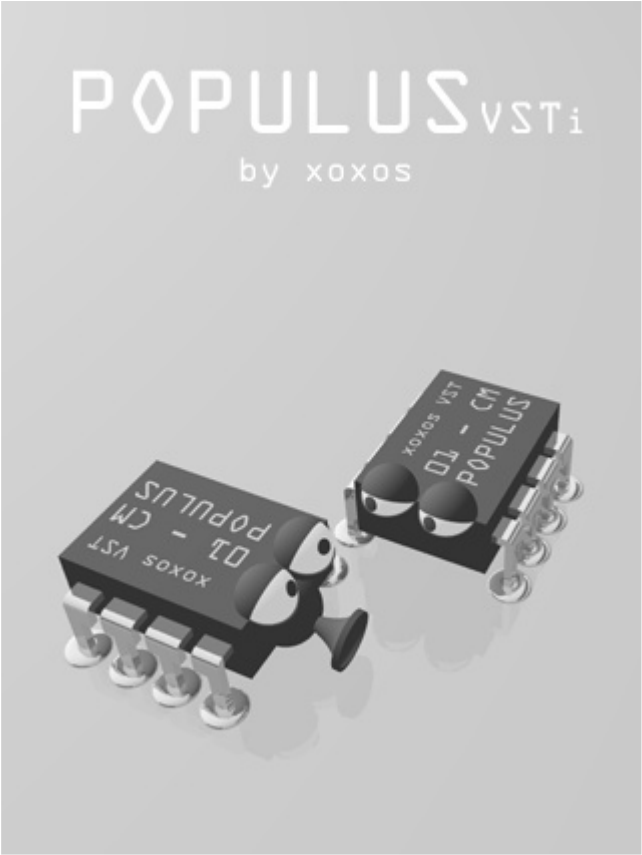
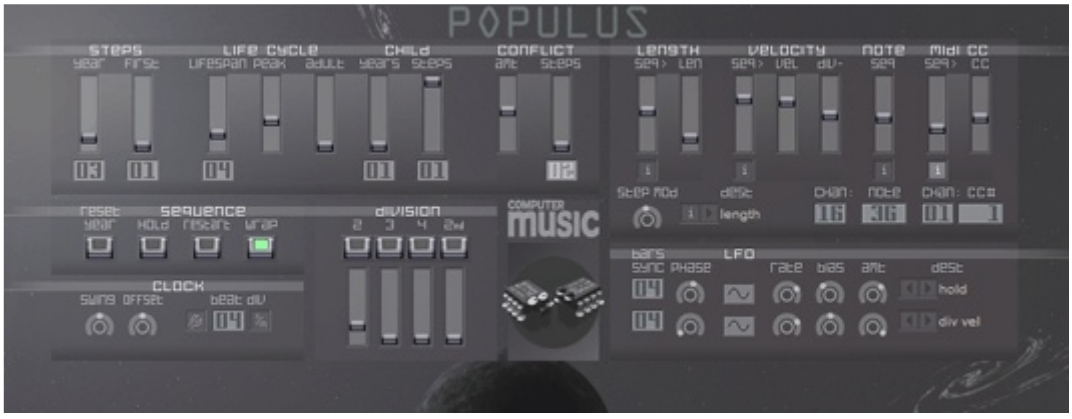




COMPUTER music

What is Populus?

The algorithm used in Populus derives the sequence from its previous output using seven behavioural parameters. The generated sequence is usually patterned, yet rarely repeats exactly, similar to the use of motifs in instrumental music.

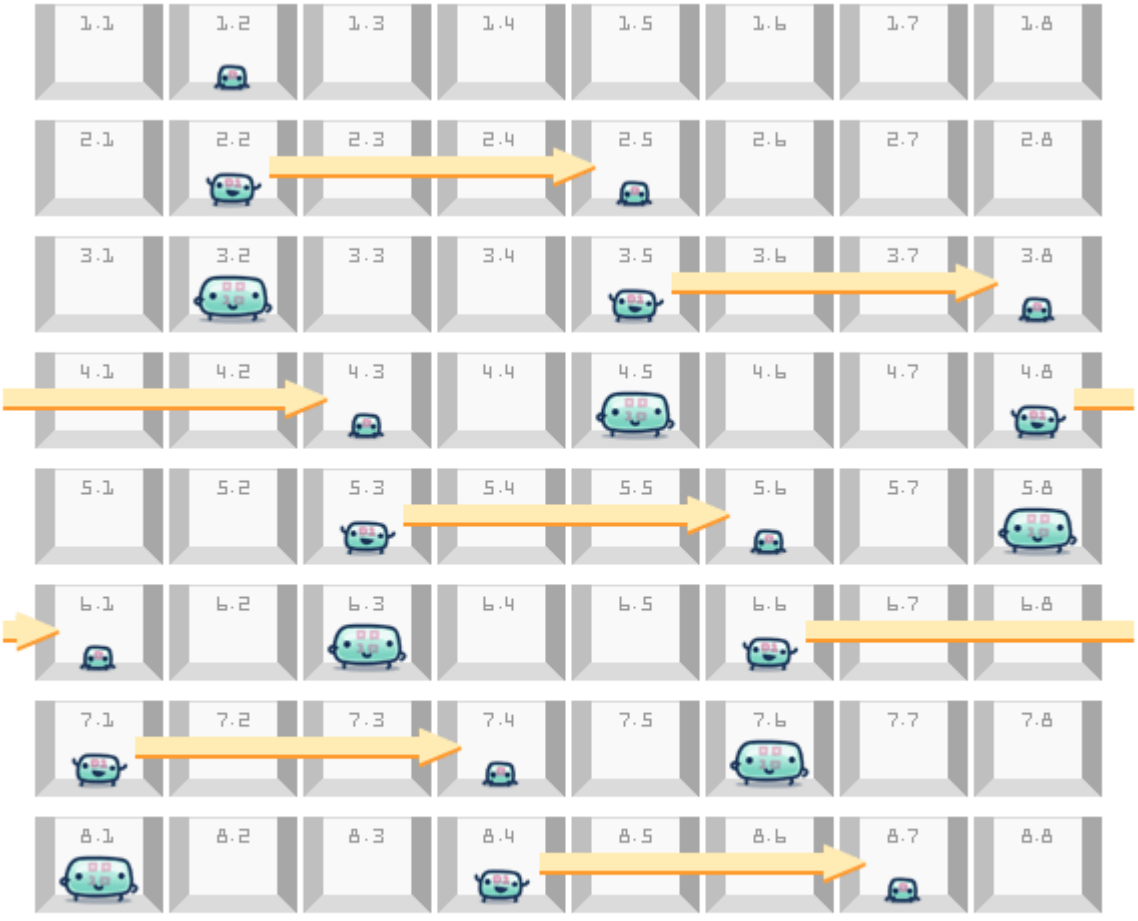
The basic sequence consists of a number of steps, called a "year." The output is calculated at the beginning of each year.

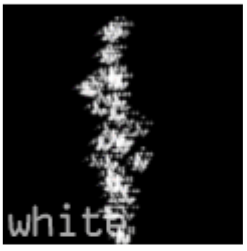
Active, or "living" steps have a lifespan set by the user. Over the course of their lives, steps grow, age, give birth and/or fight with other steps. These parameters describe the "behaviour" of the sequence.

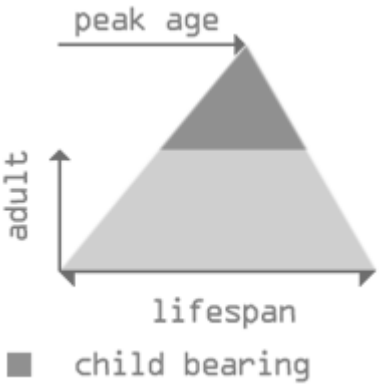
The "growth level" of active steps can be applied to midi note or velocity of the sequence, or other targets.

The top left panel contains the behavioural parameters, starting with the number of steps in the cycle, and which step is active when the sequence starts.

Active steps will create other steps whenever their 'growth' is above adult level. Children are created at the set number of steps before or after the parent.







The peak of growth can be weighted from the beginning to the end of the lifecycle, creating descending, ascending or ascending then descending modulation.

The number of years between children does not come into play when there is no conflict unless settings are changed, as the child will exist longer than the parent.

Conflict is calculated at the set number of steps. The growth of the steps is compared, and the difference is subtracted from the lower and added to the higher in the proportion set by the 'conflict amount' slider.

A victor who is past the peak age will be diminished by conflict, though at a lower amount than the loser. (This was decided by convenience as growth is coded internally as 0 to 2).

With no conflict, sequence modulation is determined only by aging, and will eg. repeat the same pitches if applied to midi note. Conflict increases the complexity of the modulation signal.

Terminal Conflict

midi panel

The midi panel applies the sequence both to midi note output and to a midi cc destination, which can be assigned to a separate midi channel and send messages for pitch bend, aftertouch, or midi controller 0 to 127. Sequence modulation settings may be inverted.

Another step modulation sequence is derived from the step using an algorithmic seed.

The seed is hidden on the gui. Click on the eyes of the singing chip and drag the mouse vertically. The eyes will change colour in response.

Midi notes may be divided into 2, 3 or 4 triggers. The '2nd' option triggers only the second of 2 subdivisions.

Each division has a slider that sets the probability for it to occur.

On the midi panel, the velocity settings include a slider labeled div-, which applies a velocity reduction for division triggering.

I suggest using an lfo to modulate this setting.

clock panel

The clock panel specifies the relation of the sequence to host tempo. Swing and offset knobs work in both directions and have a flat region in the middle to facilitate in resetting them to zero.

The sequence rate is set as a division of the beat. Rates slower than one beat can be selected using the 1/n button.

The button with the wrench icon can be used to hold the beat division setting while making an adjustment.

sequence panel

The sequence panel allows the output to be restarted or reset to the beginning of the current year.

The hold button pauses the sequence for as long as it is held down.

When wrap is deactivated, children or conflicts conducted beyond the sequence boundaries are not processed.

lfo panel

The lfos use conventional sine, saw, ramp and square wave shapes. Bias adds a negative to positive constant to the lfo suited to triggering thresholds and numeric values.

Under most conditions, sequences generated by populus don't repeat at predictable intervals (taking child step settings into account can improve this). Populus is better for creating a rhythmic texture than a discretely iterative pattern. Using synced lfos helps to apply these textures to structured music.

In the panel illustration, the first lfo triggers the hold button, which pauses the progression of the sequence while it is held. Synced every 4 bars, this would create pauses in the same places every measure (determined by phase and rate settings). The sequence will play at the same place each measure.

The second lfo is assigned to 'div vel,' the 'div-' slider on the midi panel. Using an lfo to modulate this parameter adds a rolling feel to multitriggers. Syncing this lfo every 4 measures means that multitriggers will swell at the same place each measure.

programming tips

Using child step values that aren't factors of the year will produce the most dynamic sequences, for instance if the year is 9 steps long, a child steps setting of 3 will only have three generations before births overlap and stop. Using 2, 4, 5 or 7 steps will cycle through all steps when the sequence is wrapped.

Unwrapping quickly depopulates a sequence as new steps exceed the sequence boundaries.

Like Tamagotchi and other electronic pet simuations, appropriate behavioural environments are necessary to keep the sequence populated. A sequence may also expire as a function of its numeric settings. If your sequence dies, press the restart button. If you can hear the sequence depopulating, you can try throwing the peak to the end of the life cycle.

Some people like to use these sorts of things at fast clock rates. These are included.

As with all SynthEdit plugins, holding [ctrl] allows for fine tuning of sliders.

Ableton live users need a wrapper (such Chainer or eXT) to use SynthEdit-created midi plugins.

Many thanks to Haydxn for character illustrations for the manual.

http://haydxn.deviantart.com/

license

Populus VST is distributed with Computer Music magazine and xoxos software licenses. You are not permitted to resell this software.

There is no expression of guarantee due to the use of third party resources.

Populus is made with SynthEdit
http://www.synthedit.com

IMISHT

http://www.xoxos.net

© 2008 Rurik Leffanta · All Rights Reserved
VST is a trademark of Steinberg Soft und Hardware GmbH