



**Review: CamelAudio Cameleon 5000 V1.2
by Rick Paul - 1st June 2004 -**



Question: What do you get if you cross a large mammal that is sometimes called "the ship of the desert" with a tropical lizard noted for its ability to change color to match its surroundings?

Answer 1: You mean to tell me you've read the title of this article and you still don't know?

Answer 2: A really bad joke. (Think about it. There's got to be a double meaning in there somewhere.)

Okay, I couldn't resist, but there is a point. Well, sort of...

[CamelAudio's](#) Cameleon 5000 gives you the ability to do with sound the equivalent of what these traditional "what do you get when you cross..." jokes do, at least in our imaginations, with animals. That is, Cameleon (I'm going to drop the "5000" in the interests of brevity and convenience) allows creating hybrid sounds that contain traits of the source sounds, but which could end up sounding totally new. The results may or may not work, depending on what types of sounds you use and how aspects of those sounds are used in the hybrid sounds. Either way, though, you can achieve some truly exotic possibilities, and have a lot of fun in the process of, er, "sound husbandry".

Basics

At its core, Cameleon 5000 is a software-based additive synthesizer that features morphing, sample analysis, and resynthesis. Cameleon also features a variety of onboard effects, and includes a sound library with on the order of 600 presets. If that sounds like a mouthful, it's because



there is a lot to Cameleon. Let's take a closer look.

Cameleon's basic sounds are built by adding harmonics together to sculpt the overall sound spectrum. Noise elements can also be added into the sound in a similar fashion (we'll get into more details on constructing sounds below). However, if that sounds like too much work, there is another way.

Cameleon can also import samples in WAV or AIFF format, but the raw samples are not used directly within Cameleon. Rather, they are analyzed and resynthesized such that the only difference between a sound based on an imported sample and one a user created from scratch is how closely that sound resembles whatever content was in the sample. Let's put it this way: If you want a reasonably accurate portrayal of an acoustic instrument, it's a lot faster letting Cameleon do the sample analysis and resynthesis than it is to program the sounds from scratch. You still have the ability to tweak those resynthesized sounds later on, just as if you had programmed them from scratch.

Cameleon doesn't stop at providing basic additive synthesis and sample analysis-based resynthesis, though. Cameleon allows morphing between up to four sounds to create sounds that have traits of any or all of the source sounds. These morphed sounds can create hybrid instruments, for example containing the attack of a saxophone with the sustain of a violin (or vice-versa), textures that evolve in simple or complex ways over time, and more.

Once you have a basic or morphed sound put together, Cameleon provides a nice slate of effects, including a sweepable formant filter, distortion, compression, chorus, delay, and reverb. Some of these are borrowed from CamelAudio's SuperCamelPhat multi-effects plug-in ([see product review](#)), and others are new to Cameleon.

Cameleon is surprisingly easy to program. If you're the type who would rather just sit down and make some music (and I'll admit to being in this category much of the time), though, Cameleon also includes a library of approximately six hundred preset "Instruments" (simple or morphed sounds plus modulation and effects settings). Individual Voice Programs (the source sounds used in morphing) are based on sample content from Sonic Reality, a highly respected sample house known for their SampleTank-based SonicSynth, among other products. These presets are categorized by type (e.g. Strings, Brass-Winds, Guitars, etc.), and include most everything you'd expect along with more esoteric and creative content that show Cameleon in its best light.

Cameleon is available for both Windows and Macintosh as a VST instrument plug-in, and also as an AudioUnits plug-in on the Mac. We reviewed the Windows VST plug-in under Cakewalk's SONAR 3.1.1 with the corresponding Cakewalk VST Adapter. Installation under Windows involves extracting the files from a "ZIP" archive to your VST plug-ins directory, then running the Cakewalk VST Adapter's plug-in registration program to update the adapter's configuration files so SONAR and other DirectX-based applications will recognize Cameleon. Software protection consists of a password on the ZIP file and a serial number that must be entered when the plug-in is opened for the first time (or opened by the VST Adapter).

Cameleon 5000 sells for \$199 directly from CamelAudio's web site. In addition to Cameleon itself and the preset library, a 64-page on-line manual in Adobe Acrobat (PDF) format is included. The manual does a very good job of treading the fine line between being comprehensive enough to cover Cameleon's features thoroughly and being tutorial enough to help the first time user get up to speed quickly.

"What Sounds Do Cameleons Make?"

Once you've plugged Cameleon into SONAR, Project5, or another VST host, perhaps the easiest way to get started with it is to try out some of its presets. That will keep you busy for a good long while as you go through the various Instrument groups and the sounds among them that approximate acoustic instruments, create something totally new, or come somewhere in between those two extremes.

To my ears, the preset sounds are a mixed bag. While there are some quite good imitative sounds (e.g. Breathy Sax, Bagpipes), there are also some that don't even come close (e.g. most of the acoustic pianos). The sounds that morph multiple acoustic instruments together, or don't even attempt to be imitative, tend to be much more interesting on

average. This notion highlights one of Cameleon's key strengths, not only in terms of its preset sounds, but also with respect to the types of sounds for which Cameleon is ideally suited. In particular, Cameleon seems best at creating sounds that may evoke the character of other, more recognizable sounds, but which are not strictly imitative sounds. (Note: CamelAudio indicates the next update to Cameleon will remove some of the poorer sounds and add approximately 200 new sounds which play to Cameleon's strengths.)

For example, on one project I was looking for something to evoke an Asian flavor. After trying various samples of Asian instruments in a sample player, and finding that they were a bit too literal for my needs, I decided I'd see what Cameleon had to offer. Very quickly I came across several presets that were right up the alley I had in mind. I ended up picking a preset pad that morphed between sitar, harp, and synth-like elements. No one would mistake the sound for a true sitar, yet it imparted exactly the sort of exotic feel I had in mind.

Morphing Around

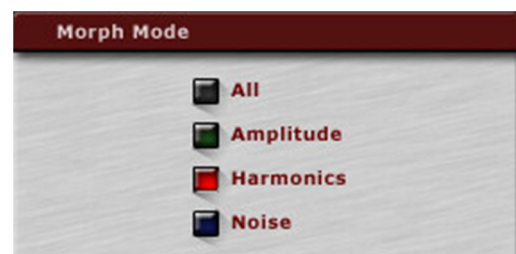
Once you've tired of the presets, or just resolved that there are too many of them to even consider trying in one or two sittings, it's time to get to the heart of Cameleon and do some morphing. You could, of course, go for broke and build your own sounds to morph between, or resynthesize sounds based on samples, but the easiest way to get started is with what the Cameleon manual refers to as "preset morphing."



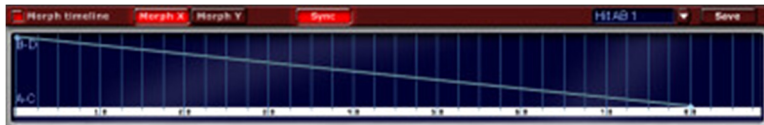
Cameleon is packaged with a large number of "Voice Programs," which are basically preset sounds created using additive synthesis, either directly or through resynthesis of sampled sounds through Cameleon's sample analysis. These sounds are sorted into descriptive categories (e.g. Brass-Winds, Strings, Pianos, Guitars, Effects, etc.) to make it quick to find the type of sound you are seeking. To morph between preset sounds, you must first pick up to four raw ingredients -- i.e. individual Voice Programs -- via the dropdown load buttons near each of the four corners of Cameleon's

"Morph Square" (see the screen shot to the left of this paragraph).

The sounds you will use for morphing will be labeled A-D, with one in each corner of the Morph Square. (You do not have to use all four corners. Any corners into which you do not explicitly load a Voice Program will contain a Voice Program called "Simple" -- i.e. a simple sine wave.) The Morph Square is essentially a controller, where the placement of a dot within the Morph Square determines where on the spectrum between the various Voice Programs the sound, or specific components of the sound, will lie. By default, all components of the sound are equally affected by placement of the dot. However, if you prefer, you can use three separate dots to control the balances of amplitude, harmonics, and noise components of the Voice Programs. (We'll get to more about the components of Cameleon's Voice Programs below.) This is accomplished by selecting a Morph Mode other than All, to place the dot for the respective component of the morphed sound.



Note that the dot that controls the morph position is not necessarily static. You can manipulate it in real time while Cameleon is playing (or even while you're playing Cameleon if you're coordinated enough to play a MIDI controller and move a mouse at the same time -- you can also control it by assigning it to be controlled by a MIDI continuous controller). Perhaps more usefully, you can record the motions of this morph controller via SONAR's parameter automation (or similar facilities in another VST instrument host), so that the dynamic morphing will be played back under sequencer control. Note that to record the automation of the morph control, or any of Cameleon's controls, in SONAR 3 you will need to place Cameleon in an FX bin, rather than in SONAR's Synth Rack. You can, however, create morph automation envelopes by hand with Cameleon in the Synth Rack using MIDI envelopes.



Dynamic morphs can get even more interesting with Cameleon's Morph Timeline. The Morph Timeline provides independent, automated control over the X- and Y-axes of Cameleon's Morph

Square over time. Breakpoints can be defined to shape the timeline's graph (click to add a breakpoint, right click to delete one), and changes in the independent Morph X and Morph Y timeline windows are reflected in the Morph Square (and vice-versa). Note that, in using the Morph Timeline feature, you give up the independent control of morphing balances for amplitude, harmonics, and noise. That is, the Morph Timeline uses the "All" morphing mode.

A Sync button is available to synchronize the morph with your DAW's project tempo, for example, if you would like to arrive at certain positions in the morph process in conjunction with certain beats or note divisions, or if you'd like a rhythmic morph to be synchronized with the beat. When the Sync button is engaged, the Morph Timeline will show beat divisions, and breakpoints snap to the nearest beat division. Drag on the beat divisions timeline to scroll to beat divisions that are outside the Morph Timeline's window for a long morph.

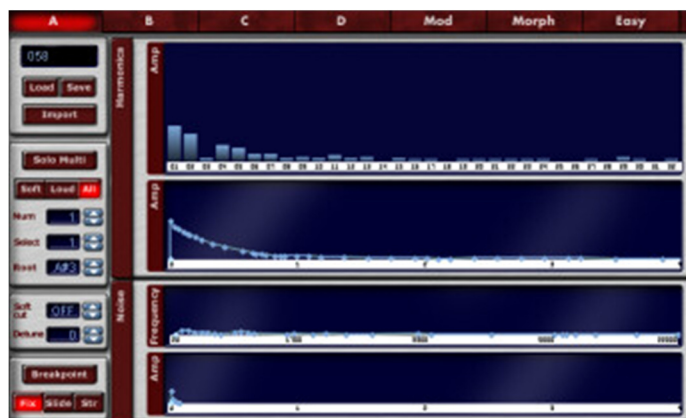
Say you create a complex morph pattern that you know you'll want to use in multiple patches. Wouldn't it be nice if you could somehow save that pattern for reuse in the future? Well, you can. The Morph Timeline window also has a Save button to allow doing just that. This is for saving morph patterns, not entire patches, so it makes it easy to take just the morph pattern created for one patch and apply it to a totally different patch. Cameleon also comes with a number of preset morph patterns, available through a dropdown selection box.

What I find particularly amazing here is that, for all the power Cameleon provides over morphing, from morphing separate elements of the sound to synchronized morph timelines, once you get the hang of a few basic operations, the morphing becomes extremely intuitive. Creating basic morphed sounds is as simple as picking the components and moving a dot around a matrix. Creating advanced, dynamically evolving morphs, even ones synchronized to your song's tempo, isn't that much harder. Some users typically avoid original sound programming like the plague because they'd rather just play music. Even these users may find that an extra few seconds to drag a dot or two around, with the goal of creating a sound that's never been heard before, is pretty unobtrusive to their creative flow.

Synthesize It Again, Cam

Okay, so you've gone through all the presets (or at least as many as you could before needing to get some real work done), morphed between preset Voice Programs to your heart's content, and now you want to do something even more unique. Or maybe you've just got ideas for an original morph but aren't finding the specific source sounds you have in mind in Cameleon's preset Voice Programs. In either case, perhaps you have audio samples, original or otherwise, that could be perfect as source sounds for the morph you have in mind -- hmm, how about morphing the sound of your spouse's voice with that of your dog's bark (don't tell him or her I suggested that!). Enter Cameleon's sample analysis and resynthesis.

Cameleon isn't a sampler or sample player. However, it can import samples, analyze them via CamelAudio's Sonic Blueprint technology, and resynthesize them using Cameleon's additive (a.k.a. "spectral modeling") synthesis. We can only touch on that feature in this review, but the basic process couldn't be simpler. Click on the tab corresponding to Cameleon's Voice Program editing page for the Voice Program you want to base on a resynthesized sample. Optionally, set some parameters to help Cameleon know about the nature of the sound you will be importing. (For example: How many different multi-samples, from 1-8, will you be importing for this voice parameter? Which multi-sample is this one? Will the individual sound file being analyzed represent a loud or soft



sample, or all velocity levels? Etc.) Finally, click the Import button and pick a 16-bit/44.1 kHz mono WAV or AIFF file to analyze. If you are using multi-samples and/or velocity layers, you will repeat the Import process for all the multi-samples and velocity layers (up to 8 multi-samples and 2 velocity layers per multi-sample).

Once you've had Cameleon analyze the sample, or all components of the multi-sample, you will probably want to give your newly created Voice Program a meaningful name then save it so you can use and reuse it, either on its own or in morphs, in any future patches. At this point, your newly resynthesized Voice Program can be used just like any of the preset Voice Programs that came with Cameleon.

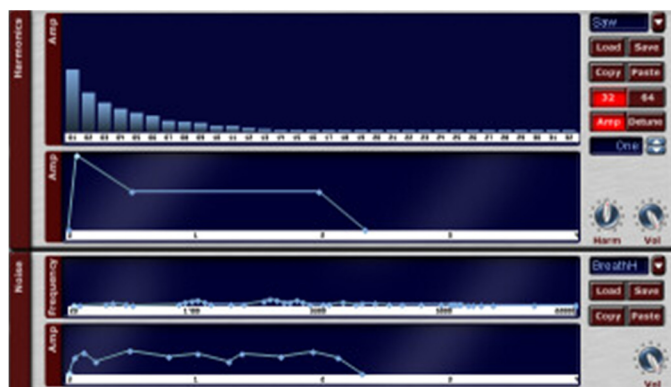
I have pretty near zero experience with sampling -- my style is to acquire quality sample sets for my needs then load them in whatever sample player is most applicable at the time. However, because the sample analysis and resynthesis portion of Cameleon seems to be a key differentiator, I figured I should at least give it a try. I initially imported a single piano sample, taken from a commercial sample library I use that happened to have files in the appropriate WAV format due to a previous sample library conversion. I didn't specify any of the optional parameters Cameleon makes available to help in the sample analysis, but rather just waited for the sample to be converted and resynthesized, then tried playing it directly through Cameleon. The resynthesized sample sounded pretty good near the note that had been sampled, and, as you might expect, not quite so good the farther you got away from the sampled note.

Wondering how multi-sampling might affect the results, I decided to try multi-samples. I imported four separate samples, taken from a range of approximately four octaves, again changing none of the other options from the defaults -- i.e. only telling Cameleon I was going to be using four samples and which sample I was importing in each operation. Predictably, the results were improved. The resynthesized samples certainly wouldn't give GigaPiano a run for the money, but they wouldn't fare so poorly against some of the synthesized pianos I've heard in the past, either. Of course, if I'd had the time and inclination, I could have improved upon my results by taking more samples, using both velocity layers, and potentially tweaking other parameters. For a quick sample analysis and resynthesis exercise, though, I was pretty pleased with the results. Also, it's important to remember that the point was not to get a great piano sound -- I have no intention of switching away from high quality, disk streaming sample libraries for that purpose. Rather, it was to capture the basic nature of that piano sound for use in morphing with other sounds.

Once you have resynthesized a sound from samples, using that resynthesized sound as a Voice Program for morphing isn't the only thing you can do with it. You can also use the full complement of Cameleon's additive synthesis tools to, tweak, or even mangle, the sound as you see fit. We'll get to that area of Cameleon in a moment. As an example, though, if the sample you imported had an undesirable ringing in a part of its frequency spectrum, you could look at the harmonics graphs to see if you could identify where the ringing occurs, then reduce it or draw it out entirely.

Add a Pinch of...

Additive synthesis tends not to be the most intuitive form of synthesis in the world. Unlike subtractive synthesis,



where you start with a wave form that is rich in harmonic information, then sculpt areas of it out to get to the sound you're looking for, additive synthesis starts with harmonically simple wave forms and adds them together to create more harmonically complex wave forms. However, additive synthesis lends itself quite nicely to evolving sounds over time since there are many small components whose levels can be varied independently, for example to morph between one sound and another.

Cameleon provides two different types of basic sound components. First up are the pitched

harmonic waveforms, which are composed of up to sixty-four harmonics or Partials. You can think of these as pitched sounds -- basically simple sine waves -- that add together to create both pitch and timbre, depending on which Partials are included and the amplitude of each included Partial. The second component is noise, in this case from a multi-band noise generator. Think of this second component as taking white noise, which by definition has equal amplitudes at all frequencies, and running it through a big graphic equalizer to shape the frequency content of the noise. Most musical sounds contain both pitched and unpitched components. For example, with a flute, there will be the tone of the flute itself, which is largely pitched, and the breath noises, which are largely unpitched. The sound of these combines to create the overall sound we hear when a flute is played. Varying the levels of the Partials, sculpting the frequencies, and varying the levels of the noise component, all over time, provides the myriad sound possibilities our ears can distinguish.

A tutorial in additive synthesis is beyond the scope of this review. (CamelAudio's web site's [Community page](#) includes links to tutorials on additive synthesis, tutorials on Cameleon itself, and much more.) Suffice it to say that Cameleon's user manual provides a useful introduction, as well as elaboration on the specific controls available within Cameleon for making the process about as friendly as can reasonably be expected given the nature of the beast. For example, if you know you want to make a sound brighter, while you could just figure out which Partials need to be increased and go at it, Cameleon provides Partial grouping functions that allow you to set the "Bright" Partials for being controlled. When you do this, instead of raising the level of each relevant Partial individually, Cameleon pre-selects the Partials that are applicable, and raising the level raises the level for all the applicable Partials. Similarly, there are grouping controls for "Dark" Partials, odd and even harmonics, fifths, and octaves, as well as to equally adjust all Partials.

There are separate amplitude envelopes to adjust the level of both the harmonics and noise components of the overall sound over time. These envelopes are manipulated with breakpoints, with an interface that is nearly identical to that used for setting the Morph Timelines. The Partials can also be detuned by setting a Detune mode and manipulating the frequency of the Partials (the default mode for manipulating Partials is Amplitude mode, which manipulates the relative amplitude of each Partial). This capability can be useful for more accurately emulating acoustic instruments. There is also a Harmonize button to get the Partials back to a normal harmonic series in case you mess up. The Partial grouping functions mentioned above with respect to amplitude adjustments can also be used for detuning adjustments.

The frequency portion of the noise generator is also adjusted with an envelope. In this case, the envelope acts more or less like a graphic equalizer, where breakpoints you set create the various bands.

Just as there are Morph Timeline presets available in the Morph display, the Voice Program editing display has harmonic and noise presets. For example, if you'd like to start creating a sound based on a sawtooth wave, simply call up the "Saw" preset, and go to it. If you'd like to impart upon that sound a breath noise characteristic, call up one of the breath noise-oriented presets, and go from there. This can be a big time saver for getting off to a start in what could otherwise be a much more complex endeavor. Of course, you can always save your own harmonic and noise presets if you create any you'd like to reuse.

I've mentioned various facilities Cameleon provides to simplify sound creation with additive synthesis, but it should be noted that Cameleon also has facilities to provide even more control over the evolving sound, at the expense of extra work in programming the sounds. In particular, while Cameleon's Voice Program editing display defaults to an Overall mode where you are editing envelopes and Partials on an overall basis, there is also a Breakpoint mode that allows you to edit the harmonic Partials and the noise frequency envelope at each breakpoint on the envelope. Thus, for example, if the noise frequency curve of the sound changes over time, you can simply create whatever breakpoints you need on the noise amplitude curves, then select each breakpoint and fine tune the noise frequency envelope at each point. Similarly, if the harmonic balance changes over time, you can create breakpoints on the harmonic amplitude envelope then tweak the Partials at each point. Both the amplitude and frequency of the Partials can be adjusted, the latter via the Detune mode. There is a huge amount of power in this, though thinking about all this tweaking is, frankly, a bit daunting, and starts making resynthesis sound even better. The key, though, is that the power is there if you need it.

Playing Mod

Cameleon provides a heap of power for controlling its parameters via DAW-based automation, such as SONAR's automation envelopes. However, it can be a lot more fun, and often more musical, to just "play the instrument" instead of programming envelopes or even trying to record automation moves via controlling an on-screen softsynth. To this end, Cameleon provides a "Mod" (for "Modulation") page, which allows tying a range of control sources, such as MIDI continuous controllers, to modulation targets within Cameleon. (Note: This is in addition to providing a powerful MIDI Learn facility. The Learn facility allows mapping most any on-screen knob in Cameleon to MIDI continuous controllers of your choice simply by pressing the Learn button, moving the on-screen knob, then moving the control on your MIDI controller you want mapped to that knob.)



The top roughly three quarters of the Mod page is dedicated to control source to modulation target mapping. You pick a control source, pick the parameter you want to map it to, constrain the range of the target parameter via Low and High sliders, if desired, and you're good to go. Control sources that can come from a MIDI controller, such as a keyboard, include MIDI continuous controllers 0 through 118, as well as velocity, aftertouch, and key position. In addition Cameleon itself provides two LFOs as well as a user-defined Envelope, and all three of these can be used as control sources. Modulation targets include pretty much all of Cameleon's global parameters, such as morphing controls and effects settings. There are a total of eight mappings available.

Also on the Mod page are the pitch bend range dial (set in semi-tones), two LFOs (with sine, sawtooth, and square wave options as well as tempo-synchronized LFO rate controls), and an Envelope graph window. As you might expect, any envelope drawn in the Envelope window is manipulated in the same manner as any of the other envelopes in Cameleon.

I might note that the Mod page settings are not global, but rather are specific to the "Instrument" (i.e. the level that stores four Voice Programs plus effects settings and more -- what I might be inclined to call a "patch"). Thus, controller mappings can be optimized for the specific sounds you are playing, so you don't need to be worrying about the needs of other types of sounds when allocating your modulation mappings.

Essentially, the Mod page allows you to customize your "instrument" for optimal playability. This provides the opportunity to make it more expressive based on human interaction. This aspect may not be unique to Cameleon, but it's nice to have there.

Easy Does It

As you might have noted from the screen shots and discussion above, Cameleon has a tabbed interface to multiple pages of parameters, grouped by function. Four of those pages, "A", "B", "C", and "D" are for editing the individual Voice Programs. Another is for controlling the morphing. There are three additional pages, two of which, "Mod" (for "modulation" -- see above) and "Effects" (see below), are pretty self-evidently named. The remaining page, which is the sixth page in Cameleon's tab order, is, somewhat intriguingly, called "Easy".

Cameleon's user manual describes the Easy page as providing, "a selection of tools for making quick and easy adjustments to Cameleon 5000's sound." It is divided up into six separate blocks of controls.



First up is the Timbre block, which contains Brightness, Harmonic, and Noise controls. The Brightness control allows raising or lowering the overall high frequency content of the sound. The other two dials are used to raise or lower the overall harmonic and noise content of the sound, respectively.

A global LFO provides sine, sawtooth, or square wave oscillation. The LFO can be synchronized with the DAW's project tempo via a beat division-oriented LFO Rate dial. There are also dials to adjust

LFO Pitch and Amplitude depth.

The Voices section has controls to adjust polyphony (from 1-12 notes), the number of Partials (up to 64) Cameleon generates, and whether portamento is on or off and, if it is on, the relative speed of the slide from one note to another. Both the Polyphony and Num Partials controls have a bearing on the amount of CPU power Cameleon will consume. Thus, if performance is tight, it can be beneficial to trim polyphony down to the minimum needed, and to trim the number of partials back as far as you can without adversely affecting the sound. Note that turning portamento on will make Cameleon monophonic, independent of how many notes of polyphony you have assigned.

The Amplitude Envelope section is similar to a typical ADSR (Attack/Decay/Sustain/Release) envelope in concept. In particular, the Attack and Release controls work the same way as the similarly named controls in an ADSR envelope. The third envelope control, however, is called "Stretch", and does pretty much what its name implies -- i.e. stretches (or shrinks with lower values) the sound. If that still sounds too "out there", as best I can explain it, the Stretch control appears to be compressing or expanding various envelopes, such as the Morph Timeline, which affect how a sound evolves over time. Use a low value, and the sound unfolds more quickly. Use a higher value, and the sound evolves more slowly. (CamelAudio's Ben Gillett elaborates, "the stretch control is like time-stretching but without any of the usual artifacts, and is also something you manipulate as you play for more expression.") There is also a Loop button. When the loop button is active, two markers will be placed in the harmonics amplitude controls for each Voice Program, and the user can use these to set a region of each Voice Program to be repeated for as long as the note being played is held.

The Output section of the Easy page contains dials for Volume, Stereo Width, and Velocity Sensitivity. The Volume control is Cameleon's master output level control. Stereo Width controls the perceived width of Cameleon's stereo signal, with all the way to the left's yielding dual mono. The Velocity Sensitivity control determines to what degree MIDI velocity affects the volume of Cameleon's output.

The final section on the Easy page contains a single control called the "Random Easy" button. Those who read my SuperCamelPhat review may remember the "Random Camel" button, and the function here is similar. Basically, clicking on the camel assigns random values to most of the controls on the Easy page, thus providing quick and convenient way to experiment with different settings. Just beware that, if you have spent any time adjusting Instrument parameters, you may want to save the Instrument first before pressing the Random Easy camel because there is no "Easy Undo".

Got Effects?

Moving over to the last of Cameleon's editing pages, the Effects page provides five effects modules. They are the Formant Filter, a Distortion module based on CamelAudio's CamelPhat plug-in, a multi-mode resonant Filter based on the one in CamelPhat, Chorus, Stereo Delay, and Reverb. These modules can be individually enabled or disabled. Their order in the signal chain is fixed, however.

In some senses, Cameleon's Formant Filter is like a big graphic equalizer, albeit controlled by envelopes in a manner that is just like the interface to the frequency envelope portion of Cameleon's noise generator. However, it's important to note that you can modulate the Formant Filter's frequency using any of the modulation sources available on the Mod page. This has the effect of sliding all breakpoints up or down by an equal amount. Consider, for example, the possibility of sliding a bandpass filter based on MIDI note number (i.e. Key Position) such that the bandpass filter accentuates the frequencies around the note you are playing. The Formant filter also provides a number of presets for common tasks such as boosting or cutting highs or lows, simulating a cheap speaker, and more.



Cameleon's Distortion module borrows heavily from SuperCamelPhat's Distortion Unit, Magic EQ, and Compressor modules. The Distortion dial provides clean, hard-edged distortion, while the Tube dial simulates tube overdrive. The M-Bass dial enhances low frequencies for a "phatter" sound, and the Comp (for "Compressor") dial compresses the dynamic range while raising perceived loudness levels.

The Filter module, which is based on SuperCamelPhat's filters, provides either a high pass, band pass, or low pass filter with a sweepable cutoff frequency, resonance control (this accents frequencies around the cutoff frequency), and an "envelope follower". The envelope follower modulates the Filter's cutoff frequency based on the level of the signal passing through it. There is also a red button above the resonance dial that turns on a "phat" mode, to yield a warmer sound with traces of harmonic distortion.

The Chorus, Stereo Delay, and Reverb modules are pretty much what you'd expect based on their names. All three modules feature a control for setting the wet/dry mix level (Mix). The Chorus also includes controls to set the rate of the chorus effect (Freq), the length of the delay used to create the effect (Delay), and the depth of the effect (Depth). The Delay also features controls to set the amount of the delayed signal that is cycled back into the delay (Feedback), and to set the tempo-synchronized delay times for the left and right channels (Sync L and Sync R, respectively). The Reverb includes controls to set the size of the perceived space (Size) and the reverb's decay time (Decay).

In general, the effects modules are serviceable, but won't be substitutes for high-end dedicated plug-ins. They are there more for convenience in sound programming. If you don't already have SuperCamelPhat, the Distortion and Filter units will be a nice addition to the sound shaping possibilities for Cameleon, and the Formant Filter is a particularly nice touch with its ability to slide its envelope breakpoints based on whatever control source you might choose to use.

Playing Around

When I get a new softsynth to review, I generally start by installing it, playing around with it a little, reading the documentation, then trying to use it on a real life project. As it happened this time, though, when Cameleon arrived I was running behind on a production project, and was at a point where I needed some type of synth-oriented sound. I didn't quite know what I needed, but thought it would be nice if it had some kind of Asian flavor to it, and I'd been experimenting with both ethnic instrument samples and various softsynths I'd had available at the time, but wasn't quite finding what I needed.

Normally, in this sort of situation, I wouldn't even think about installing a new piece of software and playing with it. However, whether it was just to procrastinate a little, or because some elusive sixth sense was leading me

somewhere, I decided to install Cameleon, plug it into the project, and see what it might have to offer. Honestly, it couldn't have taken more than an hour of elapsed time for me to install Cameleon, find several sounds that were in the general ballpark, pick a pad-like sound with sitar elements, track a part with it, and be ready to mix. Not bad for a first effort!

The cool thing about Cameleon is that, if you have a vague idea of what you're looking for, you can generally get in the ballpark quickly by looking through Cameleon's categories to find the right "food group", then try out individual Instrument presets to get even closer. At that point, where other softsynths might force you to make a decision between getting into some heavy duty tweaking or just looking at additional presets, Cameleon's Morph Square gives you pretty good odds you can take things the rest of the way just by dragging a dot around. Yet, if that isn't enough, there is still a ton of programming power available.

To put it another way, Cameleon might just be the most efficient synth going for getting from vague notions of a sound to final sound, at least if the general character of the sound needed is up Cameleon's alley. I probably wouldn't make Cameleon my first pick if I definitely knew I needed, for example, an FM sound. However, if I wanted something that was reminiscent of an oboe, but "more ethereal" (or some such fuzzy description), Cameleon would likely be my first call.

In further playing around with Cameleon, both within and outside the context of real projects, I found that Cameleon is a whole lot of fun to play with. It isn't just the morphing, which is indeed a lot of fun in and of itself. CamelAudio has added a number of touches that make experimentation quick and easy. I've already mentioned the "Random Easy" camel button above, which makes it easy to experiment with random settings in the Easy tab, but Cameleon also includes similar facilities in other areas. For example, there is one in the Effects tab that generates a random configuration of effects settings. There is also a Random Morph camel button on the Morph page that does a similar thing with morph settings including not only static morphs, but also dynamic ones based on Morph Timelines. The kicker in this category is that there is also a Random button, not a camel button this time, for generating random Instrument settings. It is a somewhat intelligent randomization feature, though, because it takes the category of sounds you are in into account. Thus, if you are in the Basses category, the randomized presets you get will all be based on bass Voice Programs, so you don't have to worry about, for example, getting pads when you're looking for basses.

There is a lot to be said about an instrument that is fun to play with, especially when the sounds it comes up with, even when randomizing, are so good. On the downside, you may want to be prepared to "waste" a bit of time just playing with sounds. On the upside, that "playing", and the sounds it generates, can sometimes be downright inspirational, and you may find these sounds inspiring some new musical creations.

In all my playing with Cameleon, it seemed extremely stable. The only minor issue I encountered was that some of the sounds seemed to have some clicks in them at the starts or ends of notes. I never did quite figure out whether this was just issues with the programming in specific sounds, something performance related (SONAR's CPU usage meter wasn't high when this was happening, so I'm skeptical on that), or something else. (CamelAudio indicates that they have already resolved some issues in this area, with fixes coming in the next update to Cameleon. Additional issues are under investigation, also with the intent of addressing them in the next update.)

Closing Notes

CamelAudio may be very new to the softsynth business, but Cameleon 5000 is a decided winner, and stacks up well even against softsynths from much more established companies. The combination of dynamically evolving sounds, user interface simplicity, powerful sound design and automation capabilities, and just flat out fun makes Cameleon a pleasure to use on pretty much all fronts.

In my book, though, what makes Cameleon stand out most is how efficient it can be for going from vague ideas to polished sounds. That aspect will likely make it my first call synth for situations where I don't have strong

preconceptions on the type of sound needed -- i.e. when, "I don't know what I want, but I'll know it when I hear it." Since that is a pretty common situation, I suspect Cameleon will be getting a lot of use in my studio.

**Rick Paul is a songwriter living in Southern California. You can contact him at <http://www.RickPaul.info>.*

