GSinthG Manual

Welcome to GSinthG

GSinthG is a pitch-driven synthesizer, allowing the synth to be played with a monophonic input signal, e.g. a (carefully played) guitar or a voice.

Interface

The user interface has a lot of controls, but many are duplicated across the three oscillators in the synthesizer. There are a few broad areas, which we'll deal with in turn here.

Pitch

These controls relate to the pitch-tracking that drives the synthesizer.

Min Freq, **Max Freq**: These define the lower and upper frequencies that the pitch tracker will look for. Setting these to ignore frequencies below or above those expected for the source intstrument can help improve the accuracy and efficiency of pitch-detection.

Gate: This is similar to the frequency limiting parameters. It this specifies a lower limit for signal level. The pitch detection will ignore the input if it drops below this level. Again, this helps to make the pitch detection more efficient and accurate.

Speed: The Speed parameter sets the number of wave repetitions required for pitch-detection. Lower values for Speed will allow faster pitch-detection but increase the chance of false detection. Higher values will mean pitch-detection is more confident, but slower. The default value should be fine in most cases.

Oscillators

Wave: Select the waveform for each oscillator: sine, triangle, square or sawtooth.

Partial: The frequency of oscillator 2 and 3 can be set as a multiple of the fundamental frequency, allowing a small degree of additive synthesis.

Detune: Allow each oscillator to be detuned by +/- 99 cents. A little detuning can help thicken the synth sound, a lot can just sound dissonant.

Attack, **Release**: These define a simple attack and release envelope for the synthesised notes. Unlike a MIDI synth, there is no keyboard triggering a note, instead notes are triggered by the pitch tracking. When a pitch is detected, that triggers a note with the specified attack. When the incoming note ends, the synthesised note will be released.

Noise: Just adds some white noise to the oscillator to create a "dusty" character.

Level: Controls the level of each oscillator.

Main Level

Main Level: Controls the overall output level.

Auto level: Clicking this button makes GSinthG look at the other settings and then select an optimised overall level that prevents clipping.

LFO

The LFO is synchronised to the host clock and varies the main output level according to the settings. (Not all hosts will support synchronising to the host tempo).

Wave: Select the waveform for the LFO: sine, triangle, square, rising, falling or random.

Frequency: Set the frequency in terms of cycles per bar, according to the host tempo (if supported).

Phase: Adjust the phase of the waveform at the start of each bar.

Depth: Set the depth of the LFO level variation.

Hints and ideas

- GSinthG can be applied to a recorded track, but it can also be fun to play live.
- The effect could be used to create a synth part that mirrors a recorded guitar or vocal melody.

Installation

I've always aimed to ensure that the GVST plug-ins are each a single file and as compact as I could make them.

For simple plug-ins like these, installation usually boils down to copying a file, so I've never created any automated installers. I know some people would prefer an installer, so apologies for the extra hassle, but hopefully it won't be too difficult.

The installation process will vary for different hosts and different operating systems, but I'll try to cover the basics below.

32-bit or 64-bit (Windows and Linux)

The Windows and Linux plug-ins come in 32- and 64-bit versions. Generally speaking you will need the one that matches the host software you're running.

If you're not sure, you can usually tell if you look at the "About" screen, which can usually be found in one of the application menus.

Taking Audacity as an example: at the time of writing you can find the necessary detail in the "Build Information" tab of its "About" screen.

If all else fails, you could try both and see which works. These days 32-bit applications are becoming increasingly rare, so try the 64-bit version first.

General installation

1. All GVST plug-ins come compressed in a `.ZIP` file, so the first step is to extract the files from the `.ZIP` file.
2. Once extracted, you should have a plug-in file - on Windows it will be a `.DLL` file, on Mac a `.VST` file, and on Linux a `.SO` file.
3. You will need to copy the plug-in file to the appropriate folder for your host program and possibly configure the host software to find it.
4. Many hosts will allow you to specify a folder on your computer where it should look for plug-ins. For example, in the Preferences in Audacity for Windows or Mac, you can add extra locations for VST plugins.
5. In most cases, you will need either to restart the host program or re-scan the plug-in folder in order for newly-installed plug-ins to appear.
6. The exact process will depend on the software you're using. You should be able to find specific instructions by searching the Internet, e.g. "How to install a VST plugin in Cubase".

Special/default plug-in locations

On a Linux machine, the convention is to locate VST plug-ins under the `~/.vst` directory. I have all the GVST plug-ins copied into `~/.vst/GVST`.

Similarly, there is a common location for audio plug-ins on a Mac: `~/Library/Audio/Plug-Ins`. I copy all the GVST plug-ins into `~/Library/Audio/Plug-Ins/VST`.

It's usually more convenient to place the plug-ins in a location of your choosing and point your host software to it, if that's supported by the application.

License

1. GVST plug-ins are provided to the user at no cost. While every GVST plug-in is tested to the best of the developer's ability, no warranty or guarantee is offered to the end user.
2. No suggestions made by the developer or his representatives (i.e., freely offered support) are to be taken as an implied warranty or guarantee.
3. These plug-ins may only be distributed by the official GVST website, or by parties explicitly given permission by the developer.
4. GVST plug-ins are to be distributed only in their original form as intended by the developer (i.e., the unaltered archive).
5. GVST plug-ins are freeware, meaning you are never under any obligation to pay for them! However, should you wish to help support continued development of GVST software, please consider donating through the official website.
6. GVST plug-ins can be used freely to create and process audio for private or commercial works.

In a nutshell, the code's all mine, but any music or sounds you create using GVST plug-ins is all yours. Of course, if you hit the big time then do feel free to pop back and donate a little something.

Credits

- Plug-in development, website and graphics by Graham Yeadon.
- A special mention to Rick "grymmjack" Christy and Greg Pettit who helped me with the UI design and documentation early on.
- VST technology copyright Steinberg GmbH.
- The "VST" name and logo are registered trademarks of Steinberg GmbH.
- "Windows" is a registered trademark of Microsoft Corporation.
- "Mac" and "OSX" are trademarks of Apple Inc.