



CLAP Grafik auf den Webpages von u-he und Bitwig.

Wer auf der Superbooth im Mai bei u-he und Bitwig genauer hingehört hat, für den war es keine Überraschung mehr, für alle anderen hingegen umso mehr: Die Berliner Unternehmen haben das neue Plug-In Format CLAP („Clever Audio Plug-In API“) entwickelt, welches sie Ende Juni der Öffentlichkeit vorgestellt haben – gemeinsam mit dem Bitwig 4.3 Update und Beta Versionen von u-he und anderen. Das Format verspricht einerseits „offenere Lizenzen“ für DAW- und Plug-In-Programmierung, andererseits neue Optionen für User wie polyphone Modulation und bessere Kommunikation zwischen DAW und Plug-In im Geiste von MIDI 2.0. Autor Heiner Kruse hat hinter die Kulissen geschaut.

VON HEINER KRUSE

► Aufregende Zeiten... Die neuen Möglichkeiten des CLAP-Formats machen Vorfreude auf die Zukunft. So soll es z.B. Performance-Verbesserungen via Multicore-Support geben und es soll möglich sein, dass DAW und CLAP Sampler Plug-Ins über zu verwendende Samples und deren Speicherung kommunizieren, was aktuell aufgrund der Datenmengen nicht geht.

Wer aktuell z.B. Kontakt als VST in der DAW mit eigenen Samples nutzt, sollte diese geordnet separat speichern, wer natürlich DAW-interne Sampler verwendet, kann auch die Verwaltung des Hosts nutzen.

Auch andere CLAP-Details kann man schon ausprobieren oder in Videos sehen. So kann z.B. ein Randomizer in Bitwig 4.3 alle Noten eines Akkordes von U-He's Hive „polyphon“ modulieren. Das bedeutet:

Man spielt den Akkord und jede gespielte Note kann beliebig viele individuelle Modulationen für zuweisbare Parameter bekommen. Das konnte ich im Test allerdings auch mit dem Bitwig-internen Polymer-Synth realisieren. Sicher rücken die CLAP Plug-Ins auch bereits bestehende spezielle Fähigkeiten von Bitwig ins Rampenlicht. So etwas machen zu können, ist stets eine Hauptmotivation für mich gewesen, mir Eurorack-Module zu kaufen. Die Art von

PLUG-IN REVOLUTION AUS BERLIN? **CLAP...** YOUR HANDS IF YOU'RE READY



Urs Heckmann hat uns einige Fragen beantwortet.



CLAP Emblem/
Kennzeichnung
beim Laden
von Hive Beta
in Bitwig.

Sugar Bytes	11	ACE (X64)
Surge Synth Team	1	Bazille (X64)
Swar Systems	1	Diva (X64)
TAL-Togu Audio Line	3	Hive
Taqs.im	2	Hive
u-he	10	Repro-1 (X64)
UVI	2	Repro-5 (X64)
VCV	1	Zebra2 (X64)
Vital Audio	1	ZebraHZ (X64)
Waves	5	Zebralette (X64)

komplexer und subtiler Bewegung, die man damit in Sounds bringen kann, hebt sich deutlich von gängigen „Pads“ ab. So kann man faszinierende Klangwände und „Drones“ erschaffen, aber auch reizvolle Bässe. Geht das demnächst vielleicht auch in jeder DAW? Urs Heckmann sprach im Interview bei Sonic State davon, dass das neue Format besonders auch für andere Hersteller von ansonsten „ausgereiften“ traditionellen DAW's interessant ist, weil auf diese Weise ganz neue Features eingebaut werden können.

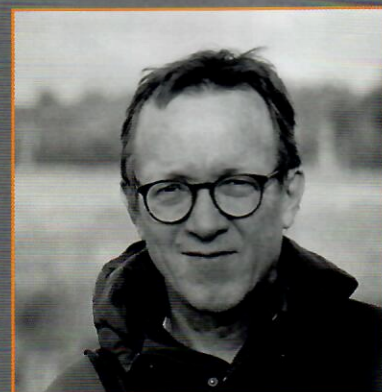
Ist es möglich, Zufallsmodulationen von Tonhöhen auf Skalen zu quantisieren? Das habe ich versucht, mir im Dialog mit Vlogger Polarity zu beantworten, bin aber nur teilweise weitergekommen: Ein polyphoner Quantizer kann in Bitwig 4.3 Ziel polyphoner Zufallsmodulation sein und kann mit Einstellungen, die man sich z.B. mathematisch herleiten kann, auch Tonhöhe quantisieren. Jedoch gibt es kein „Menü“ für Skalen. Der Quantizer kann schließlich auch alles Mögliche verarbeiten und ansteuern.

Auf clapdb.tech gibt es eine Liste verfügbarer CLAP-Betas verschiedener Kategorien (z.B. Plug-Ins, MIDI-FX, Host/DAW) und Betriebssysteme (Mac, Win, Linux). Plug-Ins gibt es z.B. von Herstellern wie Audio Thing, TAL (Tal U-No-LX und TAL DAC), Plogue, Full Bucket Music, nakst, Vital Audio, Audiority und natürlich von U-He selbst (Hive, Diva, ACE und MFM/More Feedback Machine). Andere wie Arturia, Xfer, Cableguys und Valhalla DSP sind laut u-he Website dabei, das Format

auszuprobieren. Bei Redaktionschluss waren allerdings noch keine „offiziellen“ u-he CLAP-Versionen veröffentlicht, lediglich Betas. Im KVR Forum wird über diese diskutiert.

In Bitwig 4.3 werden CLAP-Plug-Ins offiziell unterstützt und sind am Schleifensymbol erkennbar. Neu in Bitwig 4.3 sind auch ein Convolution Reverb- (mit Hüllkurven, Unterstützung von eigenen Audiofiles und „True Stereo“-Konfigurationen) sowie „Delay+“, das verschiedene „Blurring“ Mod bietet, um Delays zu verwaschen. Zudem kann man Dank Bitwigs modularer Struktur auch andere Effekte wie Pitch Shifter, Flanger+ oder Plug-Ins in den Feedback-Loop von Delay+ einspeisen! Neu sind auch zwei Module (z.B. für The Grid oder Polymer) – der „Union“ Osc, der drei Waveshapes mischt, und das Low Pass MG Filter. Die Linux Version von Bitwig lässt sich über „Flatpack“ installieren. Obendrauf gibt es noch das „Perfect Drift“ Sound Package und eine Kommentarfunktion.

Natürlich ist das Ganze für alle Beteiligten ein arbeitsintensives Thema. Hersteller, die mitmachen wollen, müssen Anpassungen vornehmen. Dafür brauchen sie Zeit und Ressourcen. Man darf gespannt sein, in welche Richtung sich die Dinge weiterentwickeln. Es war z.B. in einem Interview bei Sonic State zu hören, dass die Formatvorgaben weniger „Spielraum“ bei der Interpretation lassen, was in der Theorie die Stabilität erhöhen sollte. In den sozialen Medien kommt von vielen Seiten Lob für einen mutigen Schritt, z.B. auch von Plug-In Guru John Lehmkuhl, der sich als Hersteller des „Unity“ Plug-Ins (welches als Plug-In selbst Plug-Ins laden kann) sowohl mit Notwendigkeiten für Plug-Ins auch für Plug-In Hosting auskennen sollte.



HEINER KRUSE

Heiner Kruse ist Künstler, Autor, Dozent, DJ und Labelchef. Als „The Green Man“ (TGM) hat er national und international zahlreich Musik, Videos und Alben veröffentlicht. Seine elektronische Musik baut oft Brücken von Drum & Bass zu anderen Genres (wie bei Releases auf seinem Label Basswerk mit T. Quaeschnig/Tangerine Dream, Skarra Mucci und Gregor Schwellenbach) oder wird als Filmmusik genutzt (CSI Miami, Arte, BBC). Als Autor („Logic Pro“/Rheinwerk Verlag) und Dozent (Akademie Deutsche POP) gibt er sein Wissen praxisnah weiter. (www.basswerk.de, www.heinerkruse.de)



Random Modulation, programmiert in Bitwig 4.3, die sich bewegendem Punkte zeigen die Modulation „per Note“.

Urs Heckmann, Gründer von u-he, hat netterweise die Zeit gefunden, uns einige Fragen zu Entstehung, Details und Zukunft von CLAP zu beantworten, unterstützt von Mea Liedl von Bitwig:

Hallo ihr beiden, danke dass ihr euch Zeit für Professional audio nehmt. So wie ich Entwickler Alexandre Bique in einem Interview verstanden habe, hat er zunächst versucht, u-he Plug-Ins auf LV2 zu portieren, und als das schwierig und die Lizenz für VST3 auch wenig einladend war, kam er auf die Idee, CLAP zu entwickeln. Richtig? Ist u-he dann auf Bitwig zugegangen, um den Stein ins Rollen zu bringen?

URS HECKMANN: Ja, Alexandre hat damals CLAP entwickelt, weil er die Erfahrung mit dem Port unserer Plug-Ins nach Linux suboptimal fand. Das Projekt lag mehrere Jahre einfach ungenutzt herum. Letztes Jahr sprach ich dann Alex, der inzwischen bei Bitwig war, daraufhin an, und er hat es zu Bitwig getragen. Dann haben wir es gemeinsam vorangetrieben.

In einem Interview wird von Lizenzen gesprochen. Was genau haben Plug-In-Entwickler zu leisten, wenn sie ein VST3 Plug-In entwickeln wollen? CLAP ist „Open Source“, d.h. diese Probleme fallen weg, korrekt?

URS HECKMANN: Nicht ganz! VST3 und AU sind ebenfalls Open Source. VST3 hat aber ein duales Lizenzmodell: Für VST3 kann man entweder mit GPLv3 Open Source entwickeln oder aber man muss für „Closed Source“ einen Vertrag unterzeichnen, der es in sich hat. Steinberg behält sich z.B. dann das Recht vor, diesen Vertrag einseitig zu kündigen, und sie haben angedeutet, dass sie ihn nutzen wollen, um einen etwaigen VST2-Lizenzvertrag „durch die Hintertür“ zu kündigen. In der Version von März 2021 ist in §9 eine automatische Kündigung

sämtlicher VST2 Lizenzen enthalten. Diese wurde nach Protesten temporär zurückgenommen. Aktueller Stand der Kommunikation von Steinberg ist aber, dass dieser Passus irgendwann zurückkommen soll, aber da müsstest du Steinberg selber fragen. Das Vertrauen seitens der Entwicklergemeinschaft ist aber arg beschädigt.

Bei CLAP kann sowas gar nicht passieren. Durch die MIT-Lizenz fällt sämtliche Abhängigkeit weg. Eine lokale Kopie der CLAP Sourcen kann jederzeit von jedem wieder veröffentlicht werden, so dass weder Bitwig noch u-he noch sonst wer die Möglichkeit hat, die Entwicklung von CLAP Plug-Ins oder Hosts zu verbieten. Noch eine Anmerkung dazu: Du kannst Dir sicher vorstellen, dass die nachträgliche Aufkündigung des VST2-Abkommens für einige Firmen/Entwickler eine richtiggehend existenzielle Bedrohung wäre, vor allem für solche die ihre gesamte Codebase auf VST2 laufen haben, und dann *alles* refaktorisieren müssten. Eine Grundidee von CLAP ist, dass solche Firmen intern ihre VST2-basierte Codebase relativ einfach in eine CLAP-basierte umwandeln können. Diese können dann mit Wrappern als CLAP-as-VST3 oder CLAP-as-AU laufen. Solche Wrapper sind in Arbeit und laufen teilweise schon recht ordentlich.

CLAP soll Vorgaben genauer „definieren“, damit DAW's weniger Interpretationsspielraum haben und alles besser läuft. Außerdem können Plug-In und DAW miteinander kommunizieren. So können sie sich gegenseitig mitteilen, wann etwas schief läuft. Auch kann die DAW den Sampler fragen, welche Samples er braucht und eventuell auch das Management des Abspeicherns übernehmen. Korrekt?

URS HECKMANN: Ja. Das mit den Samples ist noch WIP, kann aber schon im aktuellen Zustand auf Github eingesehen werden.

Polyphone Modulation – kam die CLAP-Idee auch aus Erfahrungen mit Modularsystemen? Kann man mit CLAP die Flexibilität der Modularsysteme erreichen? Ist das etwas, woran u-he und Bitwig arbeiten?

URS HECKMANN: Jein. Die Idee entstammt eigentlich aus einem Ärgernis mit klassischer Automation. Wenn man klassisch einige Plug-In-Parameter automatisiert, dann kommt man nur schwerlich zurück zum originalen Preset. Die Einstellungen werden ja überschrieben. Mir ist das schon oft passiert, dass ich mitten in einem Song einfach alles wieder auf Anfang drehen wollte, und das war oft mühsam. Parameter-Modulation in CLAP ist nicht-destruktiv und temporär. Wenn die Modulation vorbei ist, ist der Knopf immer noch da, wo er im Preset schon war. Außerdem kann man während der Modulation noch am Knopf drehen. Dass das später auch noch polyphon wurde, ist das Icing on the Cake, weil eine polyphone Automation ja immer nur nicht-destruktiv sein kann. Natürlich wird dadurch alles recht modular. Es ist aber nicht für Auditorate-Modulation gedacht, sondern eher Event-basiert wie eben MIDI oder MPE, aber mit direkten Zielparametern.

Aktuell habe ich nicht ganz verstanden, wie man z.B. polyphone Zufallsmodulation in Bitwig quantisiert. Kann man Modulationen in Bitwig auch hintereinander platzieren, z.B. um eine Zufallsmodulation zur quantisieren?

URS HECKMANN: Das weiß ich auch nicht... Mea?

MEA LIEDL: Ja, Modulatoren in Bitwig Studio können sich gegenseitig modulieren. Du kannst daher den Random Modulator dem Input-Regler des Quantize Modulator zuweisen und dann mit dem Quantize Modulator den gewünschten Device Parameter modulieren. Wenn für



Convolution und Delay+ mit eingebundenem Pitch Shift Effekt im Feedback-Bereich aus Bitwig 4.3.



Union OSC und Low Pass MG in Polymer in Bitwig 4.3.

beide Modulatoren "Per-voice" aktiviert ist, erhältst Du so die gewünschte quantisierte polyphone Modulation. Zum Thema Modulationssystem findest Du in unserem User Guide auch in sehr gutes Kapitel: https://www.bitwig.com/userguide/latest/the_unified_modulation_system/

Ich verstehe es so, als könnte man jedes CLAP-Plug-In mit MPE-Controllern in Bitwig sinnvoll verwenden und extrem flexibel Modulationsziele zuweisen, und zwar schon wegen des Expressions Moduls in Bitwig, mit welchem man die Infos vom MPE Controller Parameter zuweisen kann. Richtig?

URS HECKMANN: Ich weiß nicht genau, was du meinst, aber CLAP-Plug-Ins können selbstverständlich sowieso sagen, dass sie MPE unterstützen.

Bezüglich Microtuning: Ich hab es so verstanden, dass CLAP-Plug-Ins so mit dem MTS-Standard arbeiten, dass Microtuning sehr leicht möglich ist...

URS HECKMANN: Es steht einem CLAP-Plug-In nichts im Wege, MTS zu unterstützen. Z.B. funktioniert auch MTS-ESP von Oddsound (Anmerkung d. Red: Test in professional audio Ausgabe 08-21) mit CLAP-Plug-Ins.

Bezüglich CPU-Last: CLAP kann die CPU Last des Plug-Ins besser an mehrere CPU-Kerne verteilen, richtig?

URS HECKMANN: Ja. Hosts und Plug-Ins müssen das unterstützen. Für Effekte macht dies eher wenig Sinn, aber für Synthesizer bietet sich an, die Stimmen verteilt zu rechnen.

Ich habe gehört, die Game-Industrie/Epic Games habe sich gemeldet, nachdem sie mitbekommen haben, dass CLAP Open Source ist. Weißt du mehr, warum? Man kann sich denken, dass sie mit CLAP-Plug-Ins mehr

jam



dressed to press

Kompakter MIDI Performance Controller mit polyphonem Aftertouch

Seine umfangreichen MIDI-Steuerfunktionen, der programmierbare Fußschalter und seine kompakte Form machen den Xjam zum perfekten Controller und Alltagsbegleiter für Musiker, DJs, VJs, Sounddesigner und Content Creator.

erman design. Performing worldwide.

esi-audio.com
hyperactive.de



Bewegung in Sounds bekommen können. Da geht es vor allem um die Entwicklung von Sounds, oder? Oder geht das noch weiter in die Kontrolle von Videospielen hinein?

URS HECKMANN: Da kann ich nur spekulieren, aber aus ein paar Konversationen habe ich da schon etwas herausgehört: Game Engines haben zunehmend eigene Audio Engines, da die Platzhirsche (Wwise und FMod) sehr teuer und schwer zugänglich sind (Lizenzen sind „Unobtainium“ für kleine Unternehmen). Unreal hat z.B. ein Audio-Authoring-System namens „MetaSounds“. Diese haben alle eigene Plug-In-Schnittstellen und können auch live via MIDI (oder so) Musik erzeugen. Die Idee, diese Plug-In-Schnittstellen zu standardisieren, ist aber naheliegend, weil dann die Game-Composer die dynamischen Inhalte gleich mit den Plug-Ins kreieren können, die dann hinterher bei dem Enduser laufen. Außerdem erhofft man sich wohl, dass es mehr Konkurrenz und damit mehr Auswahl und geringere Kosten gibt.

Alle warten auf MIDI 2.0, aber CLAP „liefert“ ähnliches schon vorher, wenn ich es richtig verstehe und u-he/Bitwig arbeiten nun mit der MIDI Manufacturers Association zusammen...korrekt?

URS HECKMANN: Soweit ich das sehen kann, redet man miteinander. Die MMA ist besorgt, dass CLAP zu voreilig ist, sich MIDI

CLAP Audio Software Database

This is a database of audio production software that supports CLAP (Clever Audio Plugin), an audio plugin ABI which defines a standard for Digital Audio Workstations and audio plugins (synthesizers, audio effects, ...) to work together. CLAP is a more flexible alternative to VST2 and 3, and is a royalty-free standard licensed under the [MIT license](#).

Note: Versions numbers are the first known versions with CLAP support, not necessarily the most current version.

Discussion and submissions may be made at the [KVR Audio topic](#) I created for this purpose.

Your host is Martin Diers, a.k.a. teilo.

Recent Updates

Date	Developer	Software	Version	OS	Category
01-Aug-2022	nakst	Fluctus	0.6.2	Windows	Instruments
29-Jul-2022	AudioThing	Fog Convolver	2.0	Mac, Windows	Audio Effects
18-Jul-2022	nakst	Integrate	1.0	Windows	Instruments
17-Jul-2022	Plague	Bidule	0.9781	Mac, Windows	Meta Plugins
17-Jul-2022	Full Bucket Music	The scrooo	2.0.0	Mac, Windows	Instruments
15-Jul-2022	AudioThing	Speakers	1.2.2	Mac, Windows	Audio Effects
13-Jul-2022	AudioThing	Filterjam	1.3	Mac, Windows	Audio Effects
10-Jul-2022	Full Bucket Music	Fury-800	2.4.1	Mac, Windows	Instruments
10-Jul-2022	Full Bucket Music	FB-7999	1.1.2	Mac, Windows	Instruments
10-Jul-2022	Full Bucket Music	WhispAir	1.1.2	Mac, Windows	Instruments
10-Jul-2022	CLAP Team	CLAP Info		Mac, Windows, Linux	Utilities
08-Jul-2022	Audiority	L12X Solid State Amplifier	1.3.0	Mac, Windows	Audio Effects
07-Jul-2022	Full Bucket Music	The qyooo	1.2.0	Mac, Windows	Instruments
07-Jul-2022	Qtractor	Qtractor	0.9.27	Linux	Hosts/DAWs

clapdb.tech Database.

2.0-kompatibel zu nennen, wenn MIDI 2.0 noch gar nicht vollständig entwickelt ist. Aus den Gesprächen mit der MMA lese ich heraus, dass es aber ein großes Interesse daran gibt, dass CLAP „alles richtig macht“. D.h. wir haben da noch ein paar offene Baustellen, aber auch direkten Zugang zu den MIDI 2.0-Spezifikationen der MMA.

Was genau ist mit „Voice Tracking“ gemeint?

URS HECKMANN: Um polyphone Parameter-Modulation effizient zu ermöglichen, muss der Host wissen, wann die Release-Phase einer Stimme ausgeklungen ist. Das ist in CLAP durch ein Feedback-Protokoll gelöst, indem das Plug-In dem Host mitteilt, wenn eine Note/Stimme „fertig“ ist. Das, und die Art und Weise wie der Host damit umgeht, ist sicher das, was mit Voice Tracking gemeint ist.

Vielen Dank für die interessanten Einblicke!



u-he Hive gibt es als CLAP-Beta.

Zum Schluss noch eine Anmerkung: Die Modulation via CLAP, die Urs beschrieben hat, ist als eine Art „relative Automation“ zu verstehen. So etwas ist auch schon aus anderen DAWs bekannt, allerdings nicht in einer polyphonen Variante, z.B. als „Modulation“ in Ableton Live oder als relative Automation in Logic, die dort aber nur bei sehr wenigen Parametern (z.B. Volume) möglich ist. Gleichwohl gelang es mir im Test, u-he's Hive CLAP-Beta auch mit der „Standard“-Automation in Bitwig ergänzend anzusteuern, so dass die relative Automation nicht nur von Hand, sondern auch via „absoluter“ Automation in andere Wertebereiche gebracht werden kann. Zudem steht in den CLAP-Infos: „CLAP supports per-note automation and modulation in accordance with the MIDI 2.0 specifications“.