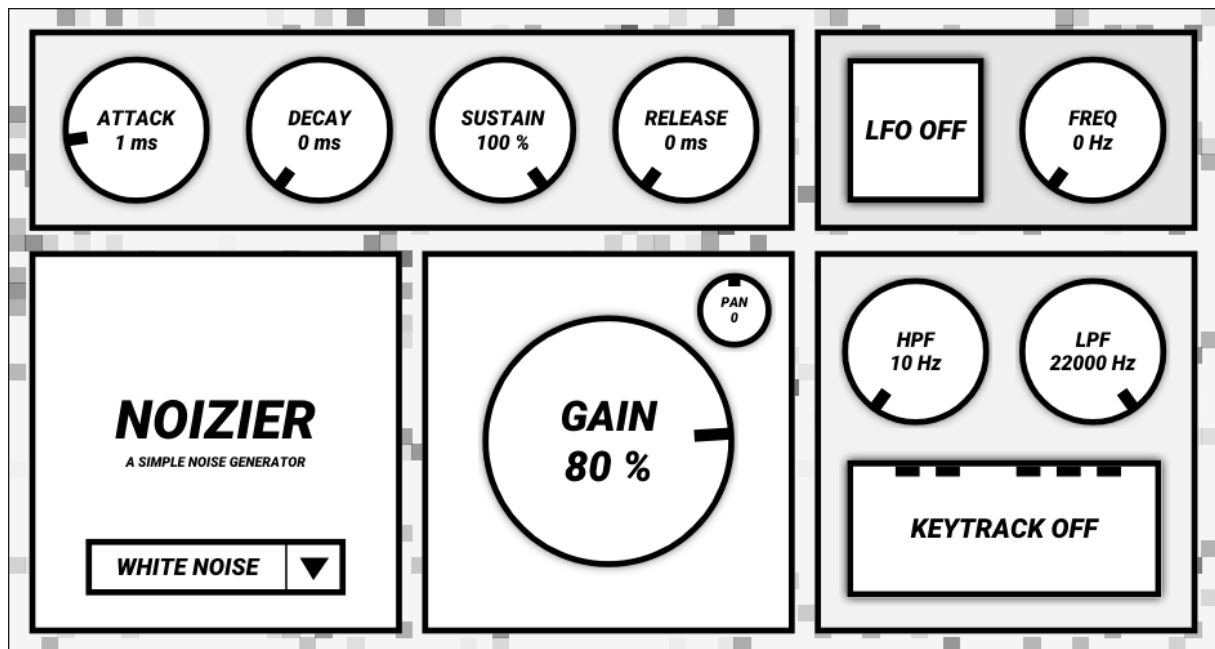


Michael Hurst

# ***NOIZIER***

**A SIMPLE NOISE GENERATOR**



## **Benutzerhandbuch**

Version 1.0  
12. Juni 2023

## Installation

Windows: Besuchen Sie die [GitHub Release-Seite](#) um das Installationsprogramm für Windows herunterzuladen. Öffnen Sie den Installer „Noizier-Installer-x64.exe“ und klicken Sie auf „Install“. Für eine manuelle Installation laden Sie „Noizier.vst3“ herunter und kopieren die Datei in den Standard VST3 Plugin Ordner unter „C:\Program Files\Common Files\VST3“.

Linux: Laden Sie „Noizier.lv2.zip“ herunter und entpacken Sie die Datei. Kopieren Sie nun den Ordner „Noizier.lv2“ in Ihren LV2 Plugins Order. Anmerkung: Die Linux Version dieses Plugins wurde nicht ausgiebig getestet und kann Fehler im Vergleich zur VST3 Version aufweisen.

Vergessen Sie nicht in ihrem Plugin-Host bzw. Ihrer DAW nach neuen Plugins zu scannen. Noizier sollte in der Liste Ihrer virtuellen Instrumente auftauchen und kann nun verwendet werden.

## Deinstallation

Windows: Deinstalliert werden kann das Plugin durch das Löschen von „Noizier.vst3“ im Ordner „C:\Program Files\Common Files\VST3“. Weitere Ordner oder Dateien werden nicht von diesem Plugin erstellt.

Linux: Löschen Sie „Noizier.lv2“ aus Ihrem LV2 Plugins Ordner.

## Credits und Lizenz

Dieses Plugin wurde mit Hilfe der JUCE Dokumentation (<https://docs.juce.com/master/index.html>), sowie den Tutorials von Joshua Hodge alias The Audio Programmer (<https://www.youtube.com/theaudioprogrammer>) unter deren Genehmigung und im Sinne von Open Source Software erstellt. Vielen Dank hierfür!

Noizier ist unter der GNU General Public Licence v3 oder später (GPL-3.0-or-later) veröffentlicht. Die Lizenz finden Sie in der Datei "LICENSE" im Stammverzeichnis des Repositories oder unter <https://www.gnu.org/licenses/gpl-3.0.en.html>

## Intro

Noizier ist ein Audio-Plugin, welches erschaffen wurde, um Rauschen mit einer Reihe von Rausch-Arten und sorgfältig ausgewählten Modulatoren zu erzeugen. Die Motivation hinter der Erstellung dieses Plugins war es ein einfach und vor allem schnell zu bedienendes Plugin für Musizierende, welche gerne mit Rauschen arbeiten, zu erstellen.

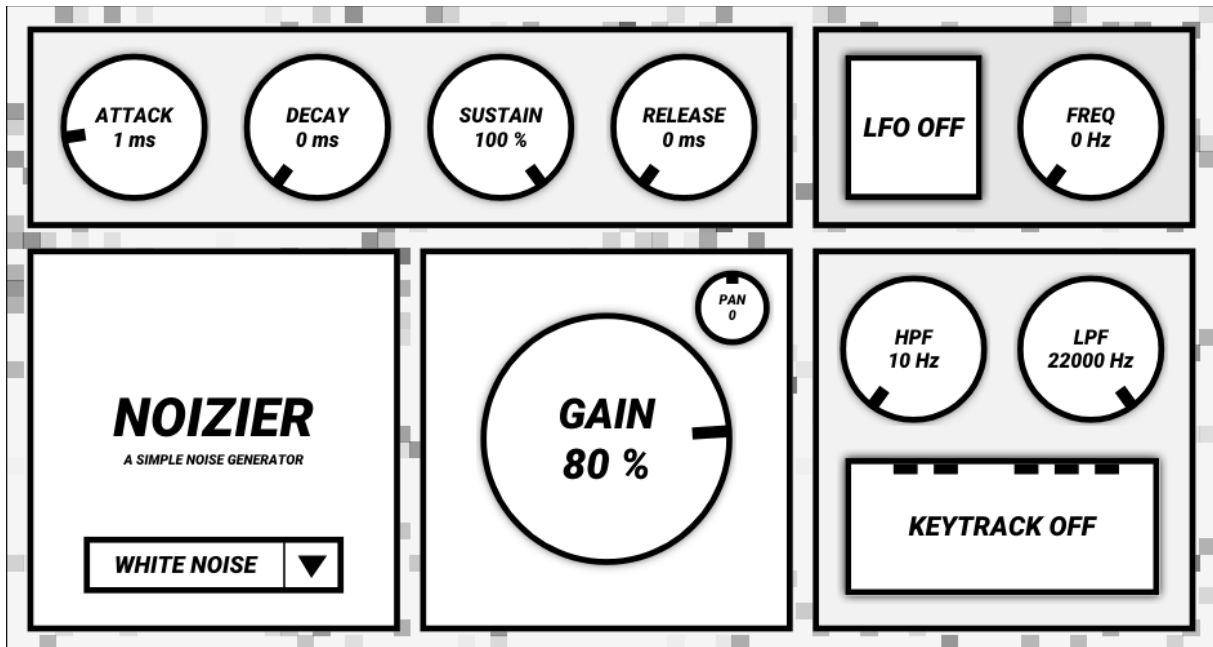
Hiermit ist es nicht mehr nötig, einen gängigen, vollumfangreichen Synthesizer zu laden, nur um dessen Noise Generator zu nutzen. Es entfallen auch die Konfigurationsschritte, um in gängigen Synthesizern ein Rauschen mit Filtern, ADSR-Hüllkurve und einem LFO-Filter einzurichten. Auch müssen etwaige Effekte nicht separat im Mixer Kanal hinzugefügt werden und sind direkt auf Noizier's Nutzeroberfläche vorhanden.

Die CPU- sowie RAM-Auslastung von Noizier fällt sehr niedrig aus, da lediglich ein monophones Rauschen mit einigen Modulatoren berechnet werden muss.

## Starten des Plugins

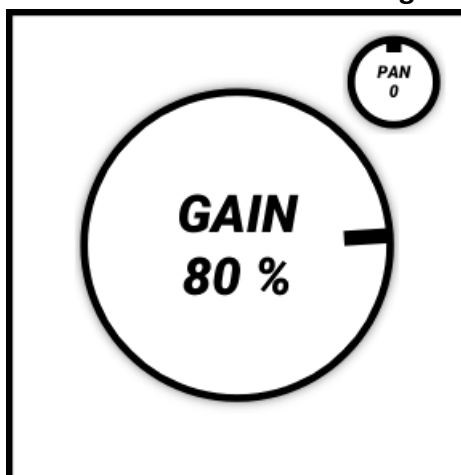
Laden Sie das Plugin in einem Plugin-Host bzw. einer DAW ihrer Wahl. Wählen Sie das Plugin an und spielen Sie einige Noten über die eingebauten MIDI-Funktionen ihres Host-Programms oder über ein externes MIDI-Keyboards. Es müsste ein Rauschen bei Notendruck ertönen.

## Programmaufbau



Noizier's Benutzeroberfläche ist visuell in logische Bereiche seiner Grundfunktionen aufgeteilt. Jeder der einfarbigen Blöcke mit schwarzem Rahmen ist zuständig für eine eigene Funktion.

## Lautstärke und Stereo-Panning



Mittig auf der Benutzeroberfläche des Plugins ist die Hauptausgabe-Lautstärke und der Stereoregler zu finden.

Drehen Sie den „GAIN“-Knopf um die Lautstärke des Plugins zu verringern oder zu erhöhen.

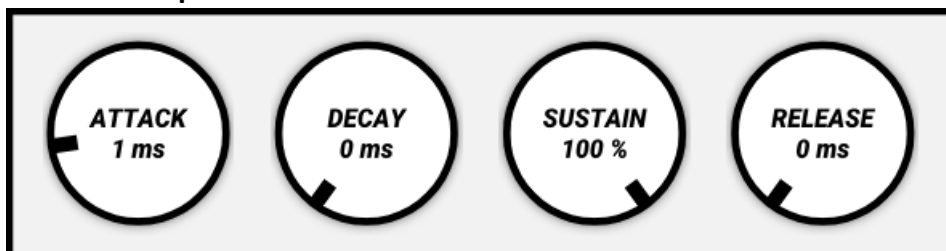
Der Knopf „PAN“ regelt die Stereoausgabe. Ein Wert von 0 gibt an, dass die Ausgabe mittig auf beiden Stereokanälen erfolgt. Ein Wert von -100 bedeutet, dass die Ausgabe hart auf den linken Kanal gepanned ist und nur dort ausgegeben wird. Im Gegensatz dazu, bedeutet ein Wert von 100 eine Ausgabe die hart auf den rechten Kanal gepanned ist.

### Geräuschtyp und Plugin Info



Klicken Sie auf den Pfeil, um ein Dropdownmenü der verfügbaren Geräuschtypen zu öffnen. Standardmäßig ist hier White Noise bzw. weißes Rauschen ausgewählt. Weiterhin ist der Typ „Pink Noise“ verfügbar, welches im Vergleich einen höheren Anteil an niedrigen Frequenzen aufweist. Auch zu finden ist der unkonventionelle Typ „Comb-ish Noise“ welcher experimentell während der Entwicklung entstand und sich ähnlich zu einem Geräusch mit Kammfilter anhört (daher der Name comb).

### ADSR-Envelope



Ein klassischer ADSR-Hüllkurvengenerator zur Verformung der Amplitude.

**Attack:** Gibt die Dauer an, bis die maximale Lautstärke nach dem Anspielen einer Note erreicht ist.

**Decay:** Regelt wie lange es nach der Attack-Phase dauert, bis die Lautstärke auf den Sustain Pegel fällt.

**Sustain:** Pegel der Lautstärke nach der Decay-Phase. Hält so lange an, bis die Note losgelassen wird. Prozentualer Wert der maximalen Lautstärke.

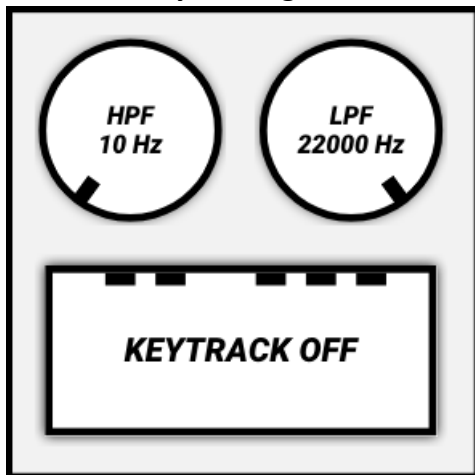
**Release:** Sobald eine Note endet bzw. losgelassen wird, beginnt die Release-Phase. Diese gibt an, wie lange es dauert bis die Lautstärke des Sustain-Plateaus 0 erreicht.

### Amplitude LFO (Low Frequency Oscillator)



Der Button auf der linken Seite mit der Aufschrift „LFO OFF“ schaltet den LFO ein. Über den Regler auf der rechten Seite können Sie nun die Frequenz des LFOs einstellen. Die ausgegebene Lautstärke des Plugins variiert nun in der Form einer Sinus Kurve, wird also lauter und leise. Bei einer Frequenz von 0 bleibt die Lautstärke ebenfalls auf 0 und das Plugin bleibt stumm, egal wie laut der zentrale Gain-Pegel eingestellt ist. Der LFO wird seriell nach dem ADSR-Envelope angewandt, das heißt beide Modulationen schließen sich nicht aus.

### Filter und Keytracking



Die oberen zwei Regler sind jeweils ein High-Pass-Filter und ein Low-Pass-Filter.

Diese haben eine Steilheit von 36 dB/Oktave.

Der High-Pass-Filter schließt alle Frequenzen unterhalb des angegebenen Schwellenwerts in Hertz aus. Der Low-Pass-Filter schließt wiederum alle Frequenzen oberhalb des Schwellenwerts in Hertz aus.

Der Keytracking-Button schaltet einen „Fake-Pitch“ anhand der Noteneingabe ein. Anhand der Tonhöhe in der MIDI-Signal ankommt, wird ein Low-Pass Filter auf das Ausgabesignal angewandt. Somit lässt sich der Filter per Noteneingabe steuern und das Rauschen erklingt in unterschiedlichen „Tonhöhen“. Damit werden Rausch Melodien spielbar.

Mögliche Tonhöhen sind hier von C0 bis C7. Bei C0 ist nur ein sehr tiefes Rauschen zu hören. Ab C7 ist der Low-Pass Filter nicht mehr aktiv und das Rauschen ist auf dem vollen Frequenzspektrum zu hören.

Die Filter Knöpfe und das Keytracking arbeiten unabhängig voneinander. Das heißt sollte ein Low-Pass-Filter bereits per Regler aktiv sein, ist das Keytracking erst ab der Frequenz des Reglers zu hören.

## Tipps und Tricks

### Fake Hi-Hats

Wählen Sie eine kurze Attack-Dauer von 0-10ms und ein Decay von 40-90ms. Stellen Sie den Sustain auf 0% und filtern Sie das Rauschen mit dem HPF ab etwa 1000 Hz. Nun erhalten Sie ein Rauschen, dass ähnlich wie das Hi-Hat eines Schlagzeugs klingt. Geeignet für einen sehr synthetischen Klang Ihrer Drums oder für das Layering mit echten Drums, um so einen artifiziell klingenden Charakter zu erzielen oder die klassischen Hi-Hats ausgefüllter wirken zu lassen.

### Vollere Snares

Wählen Sie die Einstellungen der Fake Hi-Hats und layern Sie eine Note von Noizier über eine bestehende Snare. Ändern Sie nun den Decay entsprechend der Snare Sample Länge. Stellen Sie nun den HPF runter auf 200-400 Hz und limitieren die oberen Frequenzen per LPF nach Geschmack. Dies kann helfen, um eine Snare weniger dünn wirken zu lassen. Testen Sie hier auch das Pink Noise für einen tieferen Charakter.

### Riser/Downlifter

Bekannt aus etlichen elektronischen Musikstücken und vor allem Techno, sowie Trance und Drum and Bass: Der Riser, ein Rauschen welches zum Spannungsaufbau hin zu einem Drop verwendet wird. Stellen Sie hierfür den Sustain auf 100%, die Werte Attack, Decay und Release jedoch auf 0ms. Schalten Sie den LFO ein und verbinden erstellen Sie anhand der LFO-Frequenzen eine Automationskurve in Ihrer DAW (dieser Prozess variiert je nach DAW. Sehen hierfür in der Anleitung Ihrer DAW nach). Zeichnen Sie nun eine MIDI-Note in gewünschter Länge ein und lassen Sie die Automationskurve anhand der MIDI-Notenlänge steigen. Sie erhalten ein zunehmend schneller werdendes Rauschen. Wiederholen Sie diesen Schritt für den HPF, um die tiefen Frequenzen zunehmend aus dem Signal zu entfernen. Testen Sie auch eine Automationskurve auf dem Pan Regler, um den Riser von links nach rechts schwingen zu lassen.

Derselbe Prozess kann mit sinkenden Automationskurven als Downlifter realisiert werden.

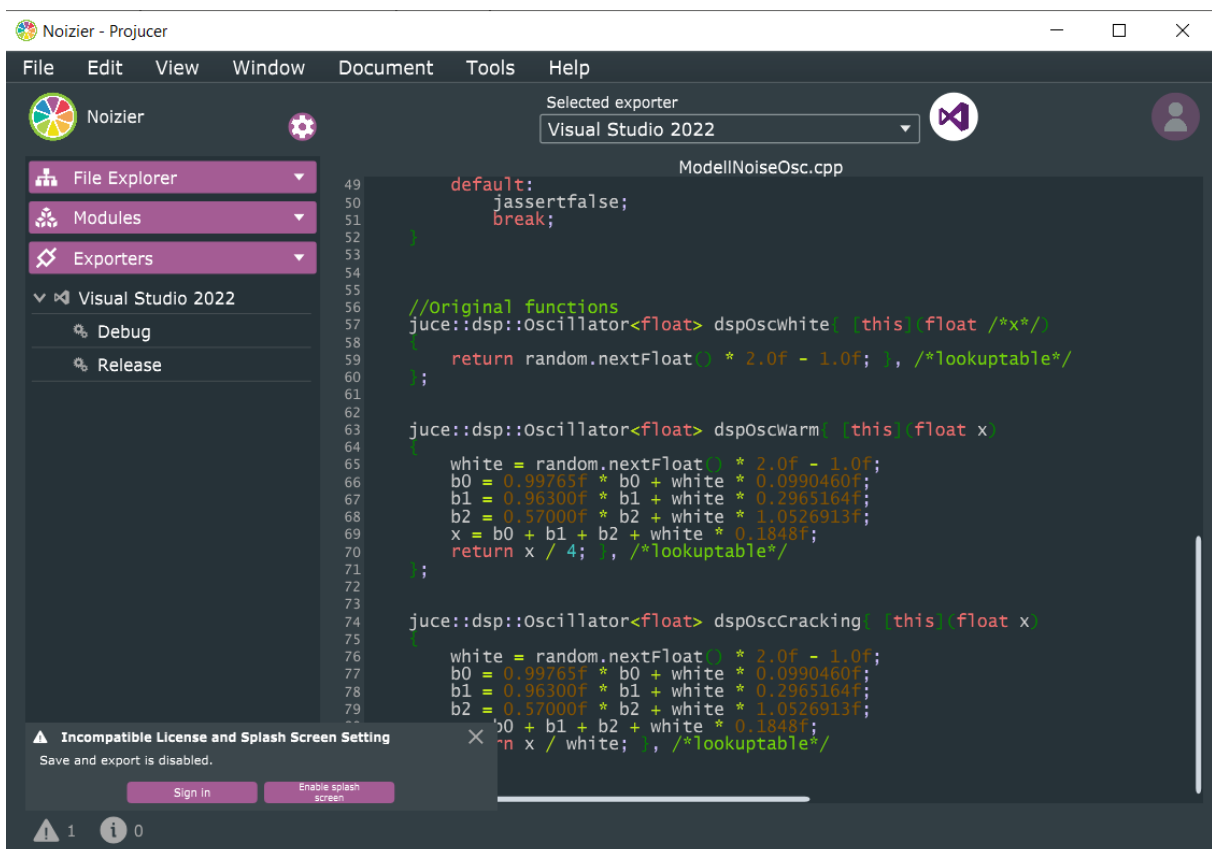
### Groove

Verwenden Sie Noizier als eigenständiges Instrument, um einen Groove in ihrem Musikstück zu etablieren. Spielen Sie hierfür kurze bis mittellange Noten in einem unregelmäßigen Rhythmus ein, ähnlich wie Sie dies auch mit Hi-Hats, Plucks oder normalen Lead-Synths tun würden. Stellen Sie dabei den Attack auf eine kurze Dauer und belassen Sie den Sustain bei 100%. Durch den Einsatz des HPF können Sie die niederfrequenten Anteile limitieren, damit der Rausch-Groove nicht in die Quere ihres Bases kommt. Passen Sie je nachdem den LPF an, um den Groove mehr in der Mitte ihres Mixes zu platzieren. Unterschiedliche Tonhöhen per Keytracking können den Groove interessanter machen und die verschiedenen Noise Typen können ebenfalls getestet werden.

## Für Entwickler: Kompilieren, Verändern und Durchstöbern des Quellcodes

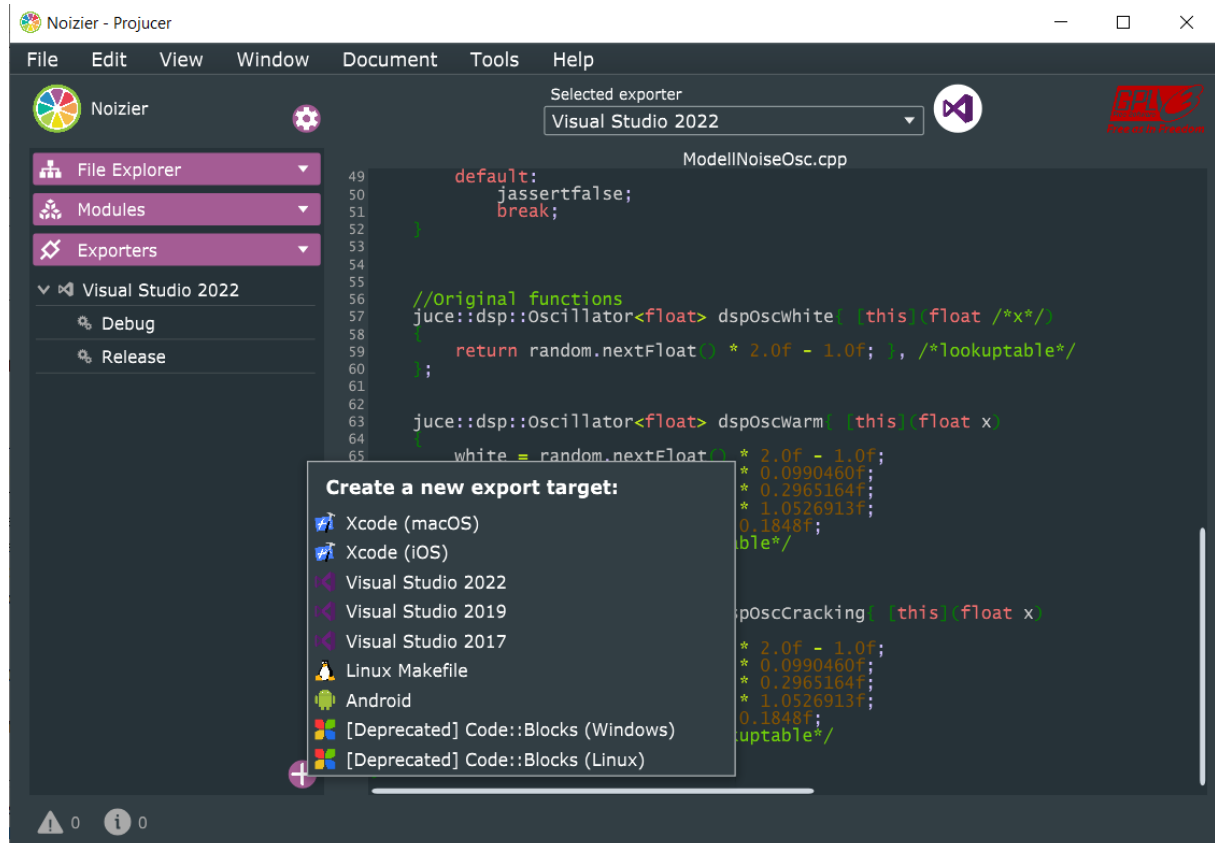
Noizier wurde mit dem JUCE Framework programmiert. Laden Sie daher das [JUCE Framework](#) für die Plattform Ihrer Wahl herunter (auf der verlinkten Website unter „Download from this website“ zu finden) und entpacken Sie den JUCE Ordner in einen beliebigen Ort auf Ihrem Computer. Öffnen Sie den JUCE Ordner, den Sie gerade entpackt haben und starten Sie das darin enthaltene Projucer Programm.

Anschließend laden Sie das [Noizier GitHub Repository](#) als ZIP-Datei herunter oder klonen Sie es über git auf ihren Computer. Entpacken Sie den Ordner. Im Projucer Programm klicken Sie nun in der Menüleiste auf „File“ und dann „Open...“. Navigieren Sie in den Ordner des entpackten Noizier Repositories und wählen Sie die Datei „Noizier.jucer“ aus.

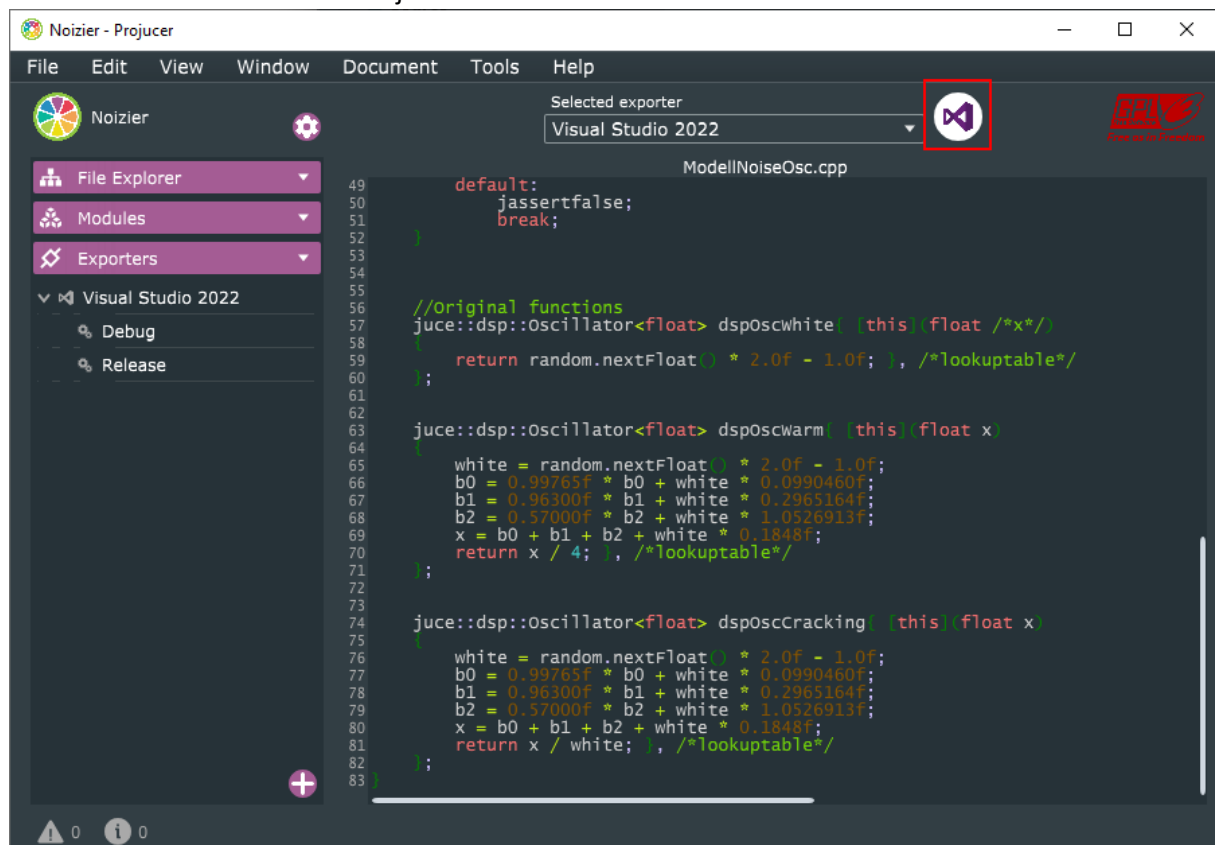


Sie werden eine Fehlermeldung über eine inkompatible Lizenz erhalten. Das liegt daran, dass der Projucer aktuell noch in der kommerziell lizenzierten Version läuft. Da Noizier als Open Source Plugin unter der Berücksichtigung der GPLv3 Lizenz entwickelt wurde, darf in diesem Falle der GPL-Modus des Projucers eingeschaltet werden. Klicken Sie in der Menüleiste auf „File“ und wählen Sie „Sign in...“. Klicken Sie nun auf „Enable GPL Mode“. Der Hinweis sollte verschwinden und das Programm läuft nun im Open Source GPLv3 Modus.

Klicken Sie nun auf das +-Symbol unter dem Reiter „Exporters“, um die Entwicklungsumgebung (IDE) Ihres Betriebssystems als Editor bzw. Compiler hinzuzufügen.



Starten Sie anschließend das Projekt mit einem Klick auf den Button ihrer IDE.





5 Durchstöbern, verändern und kompilieren Sie das Plugin mittels ihrer IDE wie gewohnt.

Hinweis: Sie müssen die entsprechende native IDE bzw. das Build-System auf Ihrem Computer installieren. Falls Sie Visual Studio 2022 unter Windows verwenden, können Sie das Projekt direkt starten, da Noizier in Visual Studio 2022 auf Windows entwickelt wurde und dieser Exporter bereits im Projekt vorhanden ist. In diesem Fall müssen Sie den Exporter kein zweites Mal hinzufügen.

Eine weitere Anleitung auf Englisch ist in der [JUCE Dokumentation](#) zu finden.

Nach dem Kompilieren des Plugins finden Sie die entsprechende Datei im Noizier Projektordner unter „/Builds/\*IDE\*/ x64/Debug/Noizier.vst3“. Der Name ihrer IDE ist dabei zu ersetzen. Suchen Sie im Zweifelsfall nach „Noizier.vst3“ im Builds Ordner. Zum Starten des Plugins kopieren Sie die vst3 Datei anschließend in den Ordner „C:\Program Files\Common Files\VST3“ unter Windows oder in den Ordner „Library/Audio/Plug-ins/VST3“ unter MacOS. Falls Sie das Plugin im LV2 Format kompilieren, dann ebenfalls in ihren LV2 Ordner kopieren. Scannen Sie in ihrer DAW / Plugin Host nach neuen Plugins.