

Modular モデル 日本語テクニックガイド version 2021.1

本マニュアルでは、GameSynth に搭載されている**ビジュアルパッチ式効果音作成モデル (Modular モデル)**について、実例とともに操作テクニックを紹介していきます。

1. 押さえておきたい機能

1.1	パラメーター群の同時表示	P3
1.2	Ctrl + P によるランダムパッチ合成	P3
1.3	シグナルアナライザー	P4
1.4	Space キー再生	P4
1.5	オートコネクト機能	P4
1.6	付箋機能	P5
1.7	クイックアクセス	P5
1.8	リアルタイムビジュアライザー	P5
1.9	モジュール解説マニュアル	P6
1.10	モジュールの置換	P6
1.11	グラフィカルオブジェクトのコピー	P7
1.12	モジュール群のインポート・エクスポート	P7
1.13	モジュールボックスのコマンドパネル	P8

2. サウンドメイキングの10個のテクニック

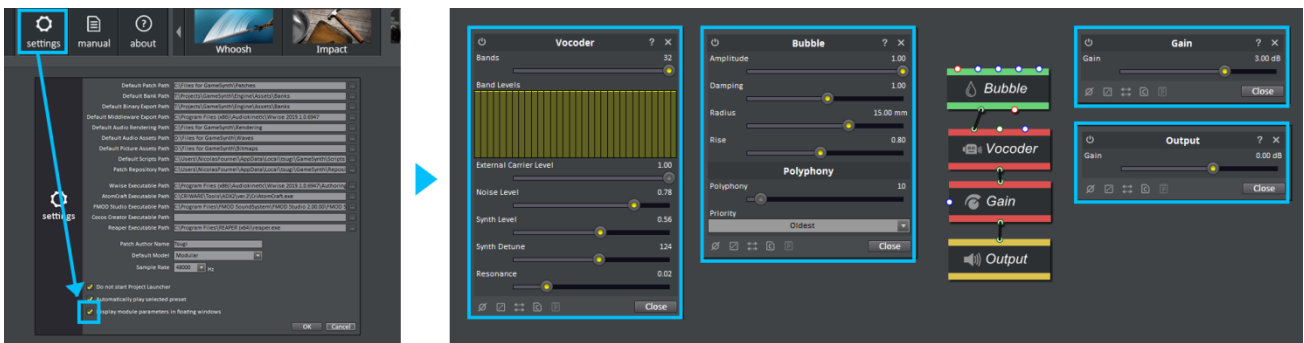
2.1	僅かに違う音を重ねよう	P9
2.2	分岐させてエフェクトをかけ、また戻そう	P10
2.3	シグナルモジュールを使って音を出してみよう	P11
2.4	パッチプレイヤーモジュールを連結させよう	P12
2.5	L O D (レベルオブドメイン) を意識しよう	P13
2.6	サンプル分析機能の結果を用いてみよう	P16
2.7	タイムシフターで音のタイミングを合わせよう	P17
2.8	ストップドシグナルで音を順次再生させよう	P18

2.9 コントロールシグナル同士を重ね合わせてみよう	P19
2.10 イベントモジュールでイベントを送信しよう	P20
3. 搭載モジュール一覧	P21
4. ショートカットキー	P27

1. 押さえておきたい7つの機能活用

1.1 パラメーター一群の同時表示 複数のパラメーターを一挙に表示できます

設定ウィンドウにて、**複数のモジュール内を同時にウィンドウ表示するモード**に切り替えておくと、サウンドパラメーターの微調整を行う際に操作が行いやすくなります。

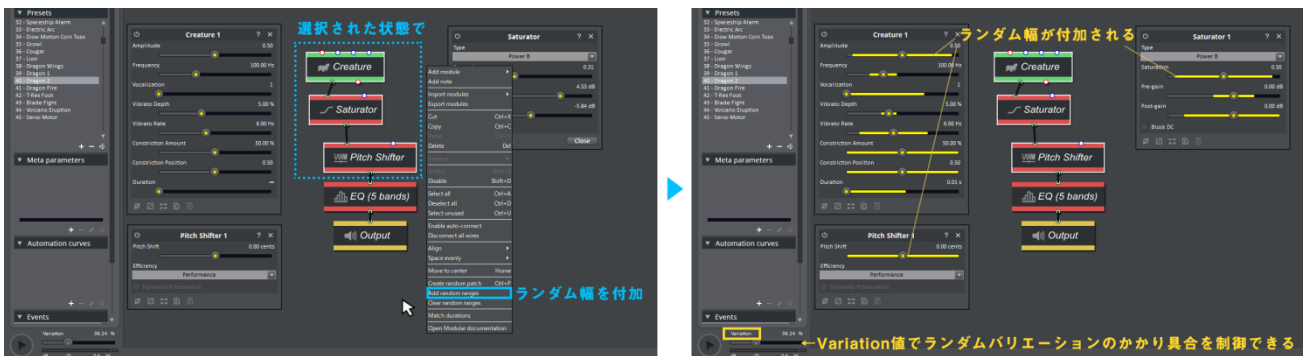


1.2 Ctrl+P によるランダムパッチ合成 ユニークな効果音やアイデアがどんどん得られます

Modular モデルのキャンバス上で **Ctrl+P** を押すたびに、使用モジュール・接続方法・パラメーターなどが**ランダムに選択された新しいサウンドパッチ**がどんどん作り出されます。

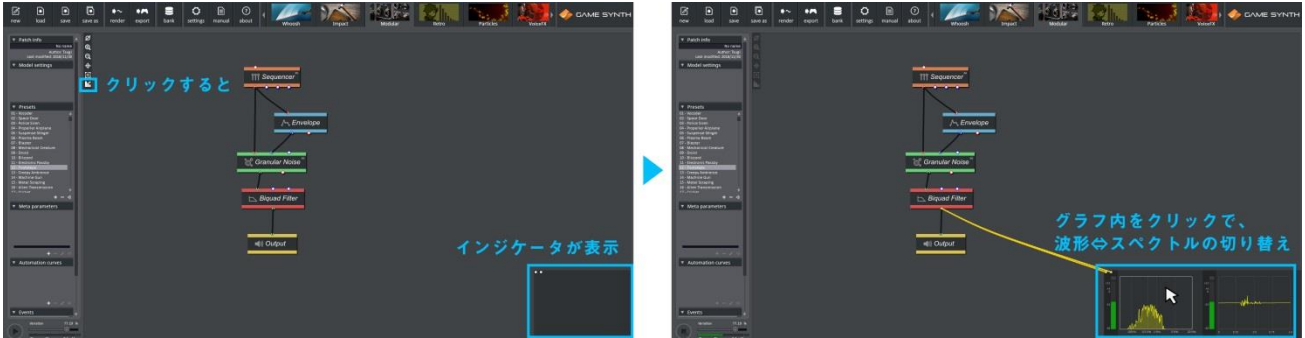
さらに、マウス左ドラッグで表示上の複数のパッチを範囲選択→キャンバス上で右クリック→Add random ranges ボタンを押すことで、パラメーターにランダムレンジが付与されます。この状態で再生ボタンを押すごとに、**違った聞こえ方の効果音がどんどん自動生成されていきます。**

うまく利用すれば、**想像以上の大量の効果音アセット**を作り出し、サウンドメイキングの幅が大きく広がることでしょう。



1.3 シグナルアナライザー 波形・周波数・クリッピングのモニタリングができます

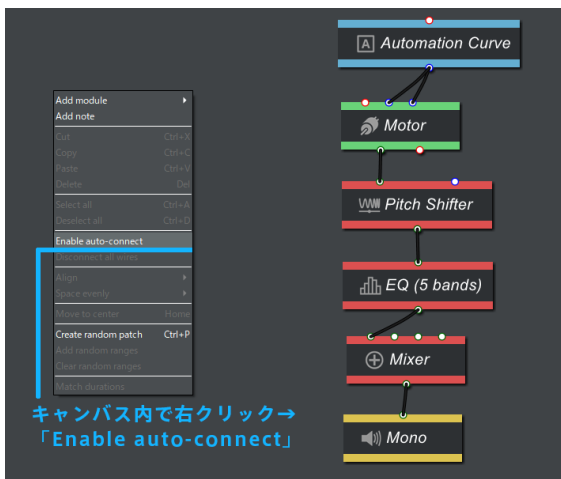
波形・周波数・レベルをモニタリングする2つのウィンドウが備わっています。任意のモジュールと接続すれば、**シグナルのチェック**が可能です。緑のVuメーターをクリックすれば、スケールの切り替えが可能です。



1.4 Space キー再生 マウスカーソルを移動させずにプレビューできるため便利です

キーボードの **Space キー** を押すことで、編集中のパッチをプレビュー再生できます。パラメーターを微調整するシーンにおいて、GameSynth 画面左下の再生ボタンをいちいち押すより便利に使えるでしょう。

1.5 オートコネクト機能 ケーブルが自動的につながります



左図の通り「**Enable auto-connect**」を ON にすることで、新規にモジュールを追加する際に、**自動的に最適なピンからピンへとケーブルが接続される**ようになります（またモジュールを消すと、その穴を埋めるようにケーブルが前後で接続されます）。ユーザー履歴やモジュール同士の距離、関連などが考慮され、ケーブルの補填が行われていく仕組みです。

1.6 付箋機能 好きな場所にメモを残せます



左図の通り「Add note」によって、キャンパス内にテキストメモを残せます。複雑になったパッチの解説などに利用していただけます。

1.7 クイックアクセス 欲しいモジュールをすぐに取り出せるショートカットキーです

本モデルに搭載されている全てのモジュールは、以下の頭文字に相当する5つに分類されています。

- G**：ジェネレーション | 発音装置のこと
- P**：プロセッシング | エフェクトのこと
- C**：コントロール | カーブなどを使った変調のこと
- L**：ロジック | 論理的な制御を行う
- O**：アウトプット | 出力

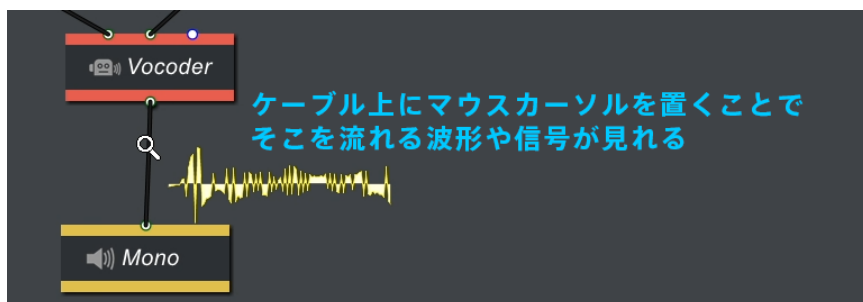
これを利用し、名前の**頭文字を使ったショートカットキー**でモジュールを即座に呼び出すことができます。

例えばジェネレーター（Generator）モジュールであるオシレーター（Oscillator）を呼び出したい場合、「G O」とキーボードで入力することで、すぐにキャンパス上に呼び出すことができます。

1.8 リアルタイムビジュアライザー モジュール内の「どこでどう鳴っているか」が分かります

マウスポインタを、モジュールとモジュールを繋ぐケーブル上に置くことで、**そのケーブルに流れている「波形」や「信号」をリアルタイムで見**ることができます。

モジュールの効果を学びたいとき、複雑なモジュールを組むときに、このビジュアライザーを活用していくと便利です。

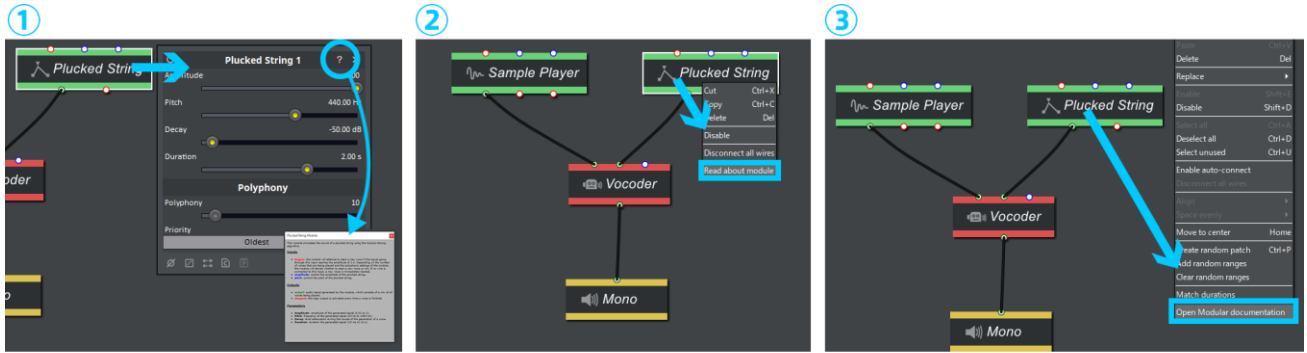


1.9 モジュール解説マニュアル 知らないモジュールの機能をすぐに調べることができます

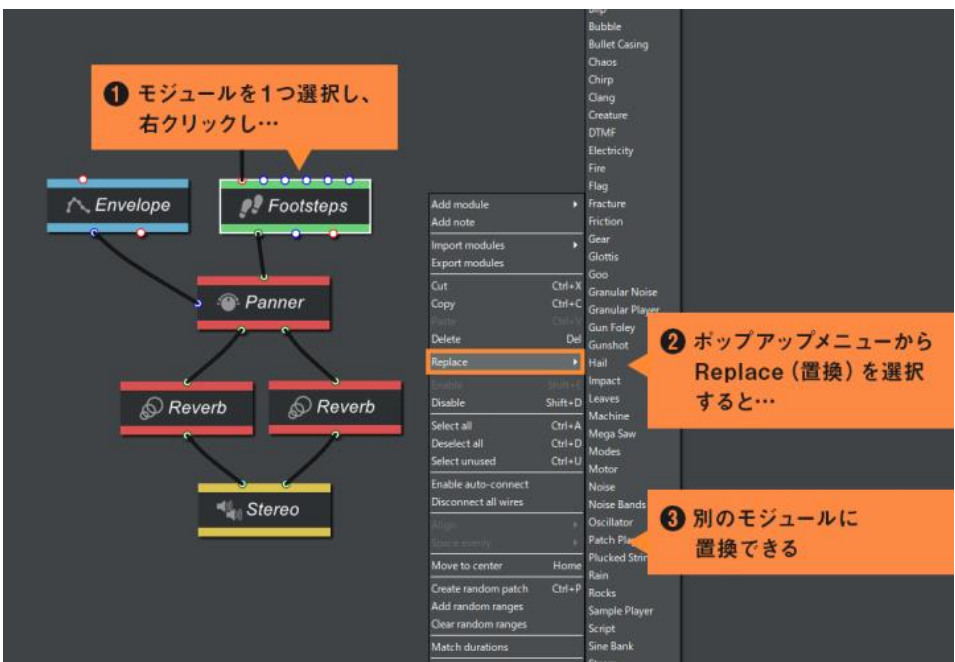
下図①：注目しているモジュールのパラメーターボックスの右上にある？アイコンから、そのモジュールの機能解説を見ることができます。※GameSynth Ver. 2021.1 時点では、解説は英語表記のみとなっています。

下図②：また、モジュールを右クリック→「Read about module」からも解説が見れます。

下図③：キャンバスを右クリック→「Open Modular documentation」から、本マニュアルが立ち上がります。



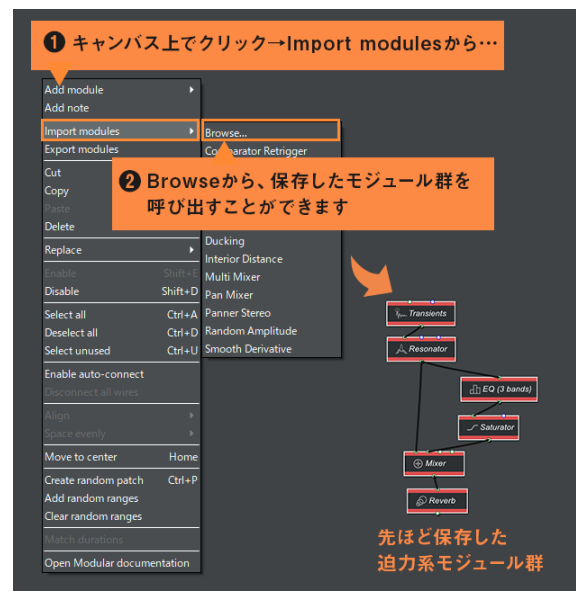
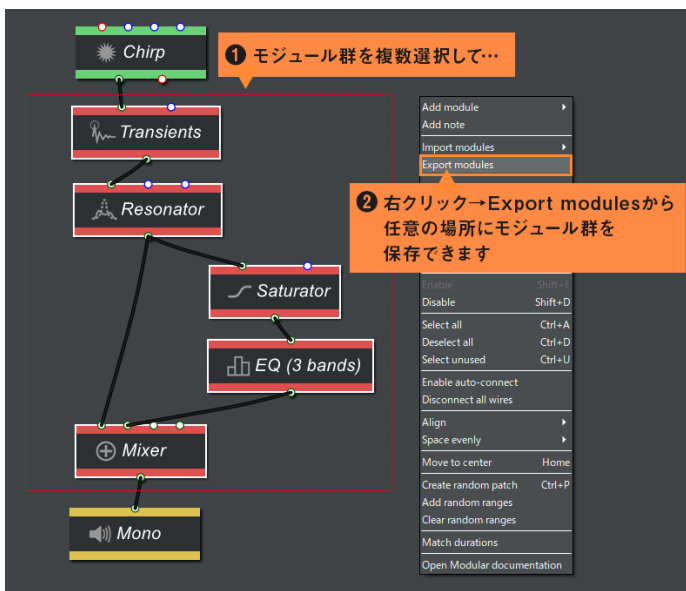
1.10 モジュールの置換 モジュールを切らずに差し換えることができます



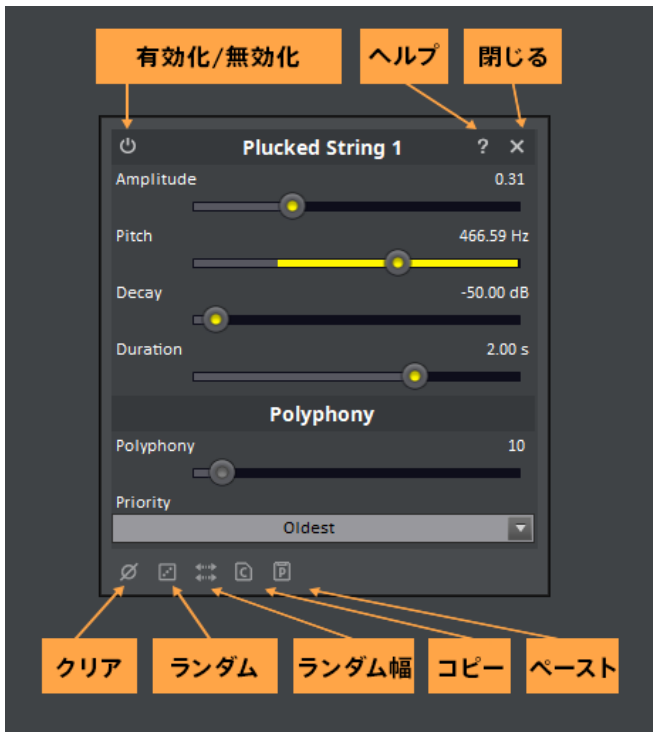
1.11 グラフィカルオブジェクトのコピー Ctrl+ドラッグで同タイプモジュールにコピーできます



1.12 モジュール群のインポート・エクスポート 機能を使いまわすことができます



1.13 モジュールボックスの**コマンドパネル** パラメーターの操作・コピー&ペースト等を行えます



2. サウンドメイキングの 10 個のテクニック

2.1

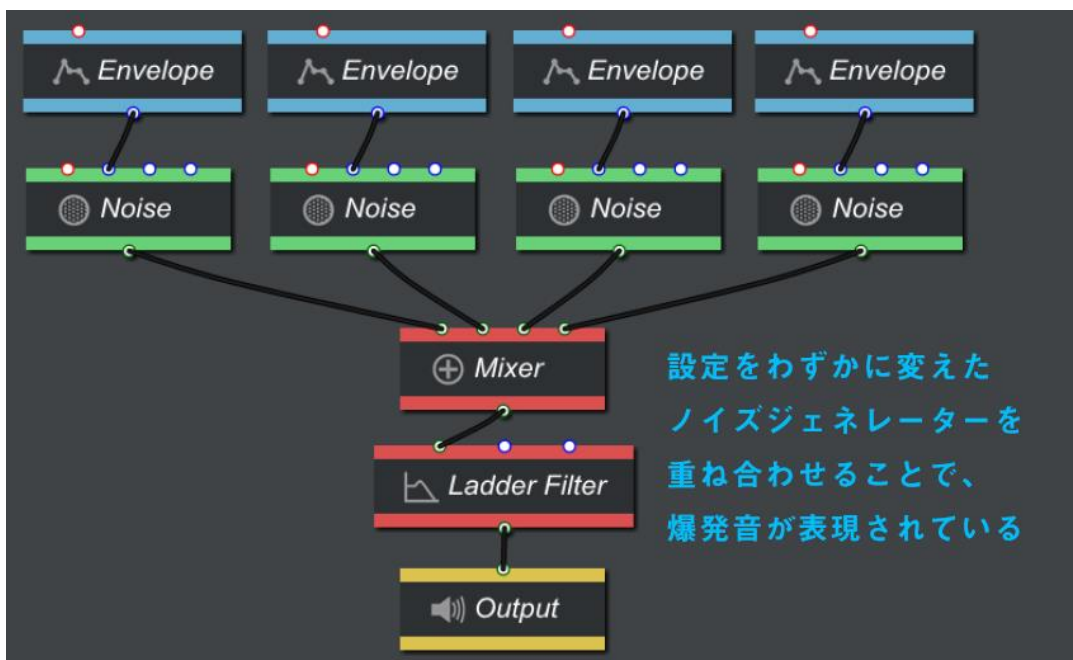
作った効果音が
どうも単調に聞こえてしまう…

▶ 僅かに違う音を重ねてみよう

サウンドデザインの世界では古典的な方法ですが、GameSynth では、下記のように**僅かに異なる設定のジェネレーターを Mixer モジュールで繋げる**ことで、同じ効果を得られます。レーザービームやアラームなども、この方法で音に厚みをもたせることができます。



たとえば搭載プリセットである「44 - Volcano Eruption」も、下図のような似た考えで爆発音部分を設計しています。



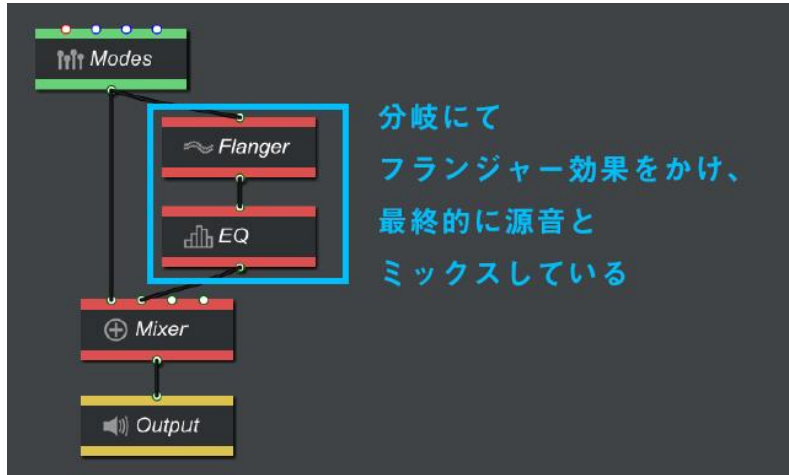
2.2

音にもっと
迫力をもちたい…

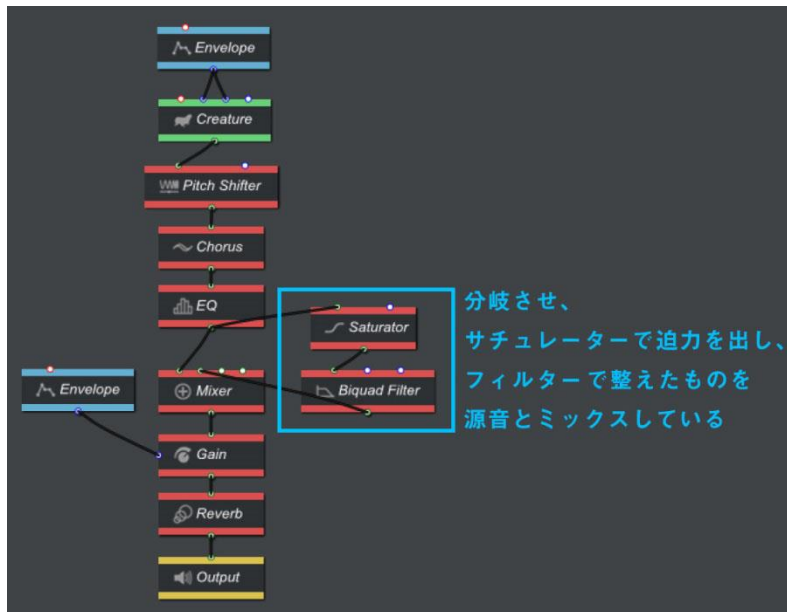


分岐させてエフェクトをかけ、
また戻そう

以下のように源音を分岐させ、分岐させた道でエフェクトをかけて、最終的にミックスする手法です。



GameSynth の搭載プリセットである「39- Dragon1」「42-TRex foot」なども、この考えで音に迫力を持たせています。
下図の青枠でくくっている部分によって、音圧のある低音の鳴き声が変わっています。



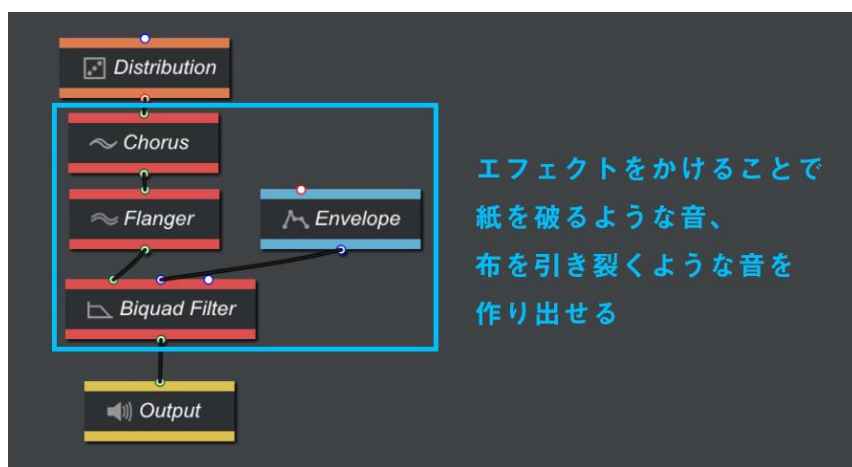
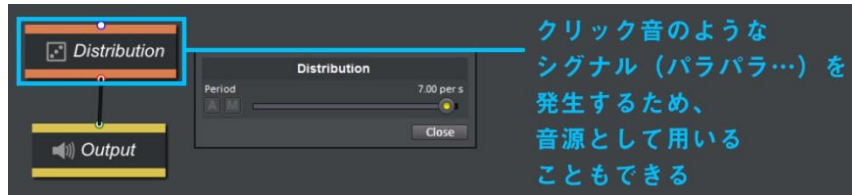
※上記のプリセットは、今後のアップデートにて設計が変わる可能性があります。

2.3

ジェネレーターモジュール以外に
音を出せないの？

シグナルモジュールを使って音を出してみよう

以下のように、Distribution モジュールから発せられるパルスを音源として利用し、サウンドを作成する方法もあります。



また Envelope モジュールも同様に、音源として扱うことができます。以下の例では、Envelope モジュールを短いループ設定にすることで、「ツツツ…」というパルスを発生させ、フィルターに通すことで、工場の機械音のような音を作成しています。エンベロープの形を変えることで、音のテクスチャも変わります。



2.4

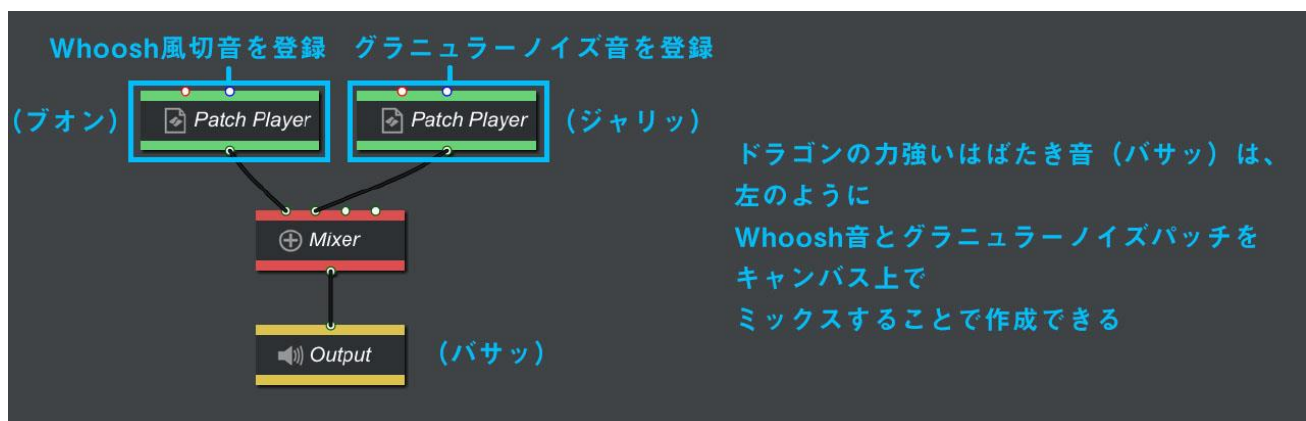
作った音同士を
ミックスさせたい



パッチプレイヤーモジュールを
連結させよう

以下のように Patch Player モジュールを複数使い、異なるパッチを組み合わせて音をミックスすると便利です。

[こちら](#)のドラゴン効果音のチュートリアルでは、「はばたき音」「尻尾で攻撃音」の作成にあたり、このパッチプレイヤーモジュールを使ってミキシングするテクニックが解説されています。



2.5

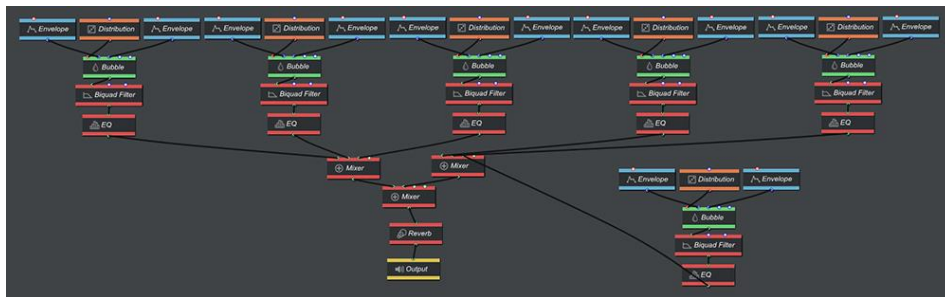
複雑な音をデザインするときの
コツってないの？



LOD (レベルオブディティール) を
意識しよう

以下の内容は、[こちらのブログ](#)にてサンプル音を交えて解説しています。

GameSynth でプロシージャルサウンドバッチを設計するにあたって、「音のディティールごとに分けて考える (=Level of detail)」ことは大事です。例えば、ザアア…ポタポタという「雨音」を作るとしましょう。雨粒がたくさん滴るので、下図のような「Bubble ジェネレーター (水音モジュール) をたくさん繋げて個別に調整する」というアイデアもおそらく可能ですが・・・

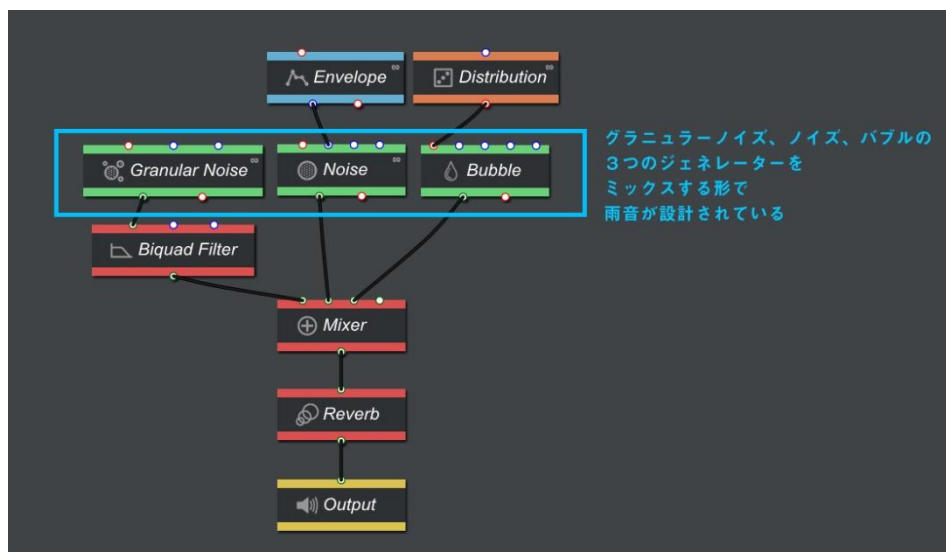


しかしこの設計では、

- 膨大なパラメーターを設定する必要が出てしまう
- わかりにくく、編集しづらい
- CPU 負荷がかかりすぎる

といった問題も出てきます。

一方 GameSynth 標準プリセットでは、下図のような 3 本のラインで構成されたシンプルな設計で表現しています。



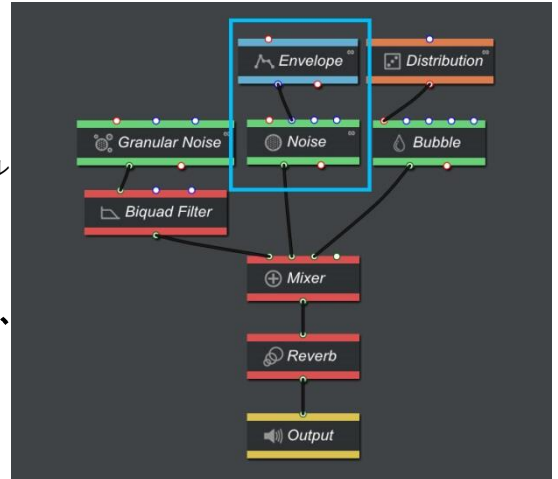
上図のバッチは、3本のライン（左から順に**グラニューラーノイズ・ノイズ・そして水音**）で構成され、それらが最終的にミキサーで音量調整されています。それぞれのラインについて見ていきましょう。

① 中央のノイズモジュール

まず「ザー」という遠く離れた場所で鳴っている雨音を出すために、低負荷なノイズジェネレーター(=バンドパスフィルターに従った、ホワイトノイズを生成する仕組み)を使います。

さらにこの緑のノイズジェネレーターは、すぐ上の Envelope モジュール（青色）によって、カットオフ周波数の上限値がゆっくりと変化するようにになっています。

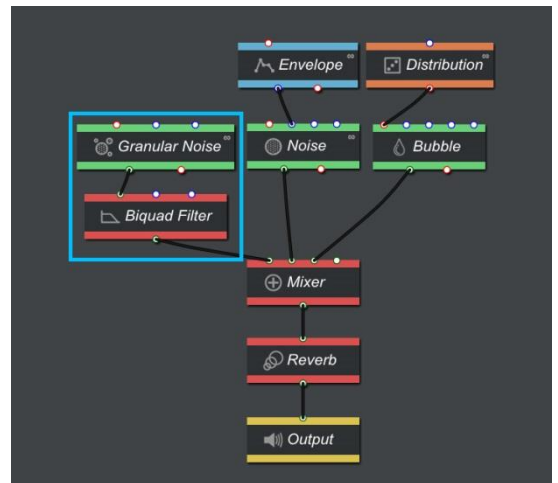
つまり、「ただ単調なノイズの鳴りっぱなし」に聞こえさせないために、**時間ごとに少しずつノイズの音色が変わるように調整している**という事です。



② 左のグラニューラーノイズモジュール

第二に、グラニューラーノイズモジュールを使います。これは、「粒っぽいノイズ」を出すモジュールで、前述のノイズモジュール同様に低 CPU 負荷というメリットがあります。

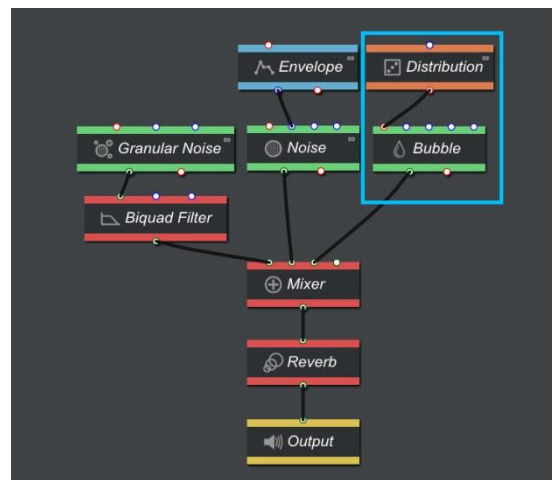
グラニューラーノイズモジュールのすぐ下に接続したバイカッドフィルターで低音のみを通すよう調整してあり、これによって、「天井や傘にあたるパラパラという雨粒」を表現できます。



③ 右のバブルモジュール

第三のバブルモジュールによって「近い距離でポチャポチャ滴っている水音」を表現します。

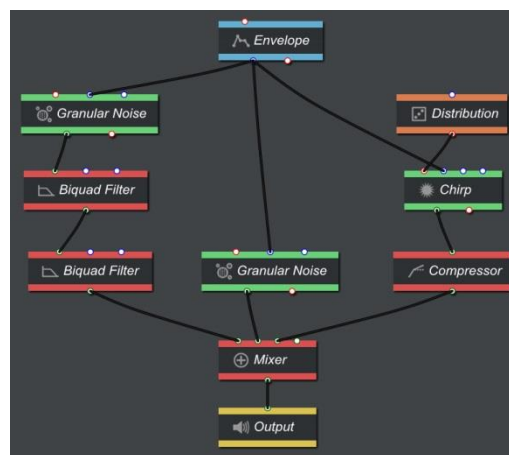
バブルモジュールの上部にあるディストリビューション（分配）モジュールによって、ランダムに次々と水音がトリガーされます。以上に紹介した3つのラインをミキサーで音量調整し、リバーブをかけて、雨音が完成します。



この雨音サウンドパッチは、「クオリティも十分」「わかりやすく個々の調整がしやすい」「低 CPU 負荷」で、一般的な雨の効果音を再生する手法と比べると以下の利点があります。

- ・プロシージャルで常に音が生成されていくため、**単なる繰り返し再生ではない**
- ・ゲームの変数に沿った、**音要素のリアルタイムな制御**が可能（たとえば水音のポチャポチャ滴る頻度、ザーという雨音のトーン、それぞれのミックス具合を、ゲーム内の状況によってコントロールできる）
- ・**380 バイトという、圧倒的なメモリの軽さ**

この LOD テクニックは、いろいろ応用がききます。例えば、GameSynth 標準プリセットである以下の「キャンプファイア」も、同じ手法で設計されています。



グラニューラーノイズジェネレーターで「ポオォ」という火の音を、チャープジェネレーターで「パチパチッ」という木が割れる音を表現しています。

2.6

音にリアル感を与えたい

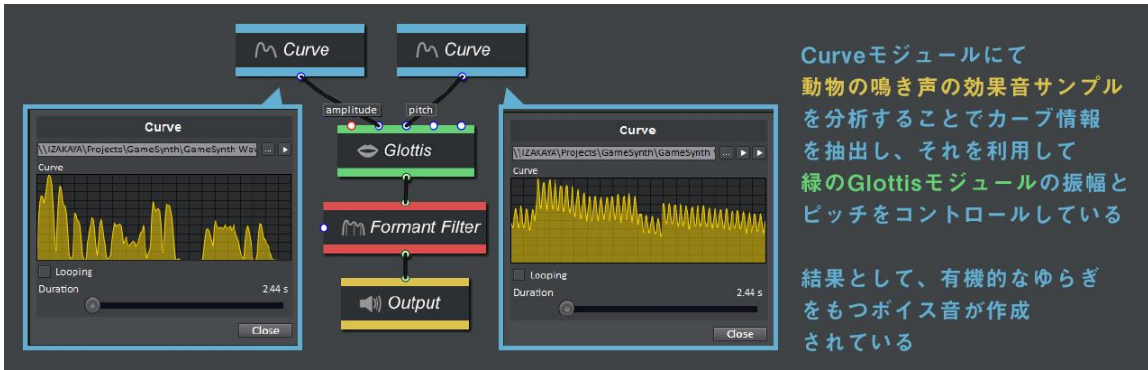


サンプル分析機能の
結果を用いてみよう

[こちらのブログ](#)で、各種サンプル分析機能を用いて、分析結果をサウンドメイキングに活用する手法を紹介しています。

プロシージャルサウンドは、録音した効果音素材に比べ、音のリアルさという点において劣ることがよくあります。そのため GameSynth には、**手持ちの効果音サンプルのパラメーターを分析、抽出し、そのデータをサウンド設計に役立てる機能**があります。

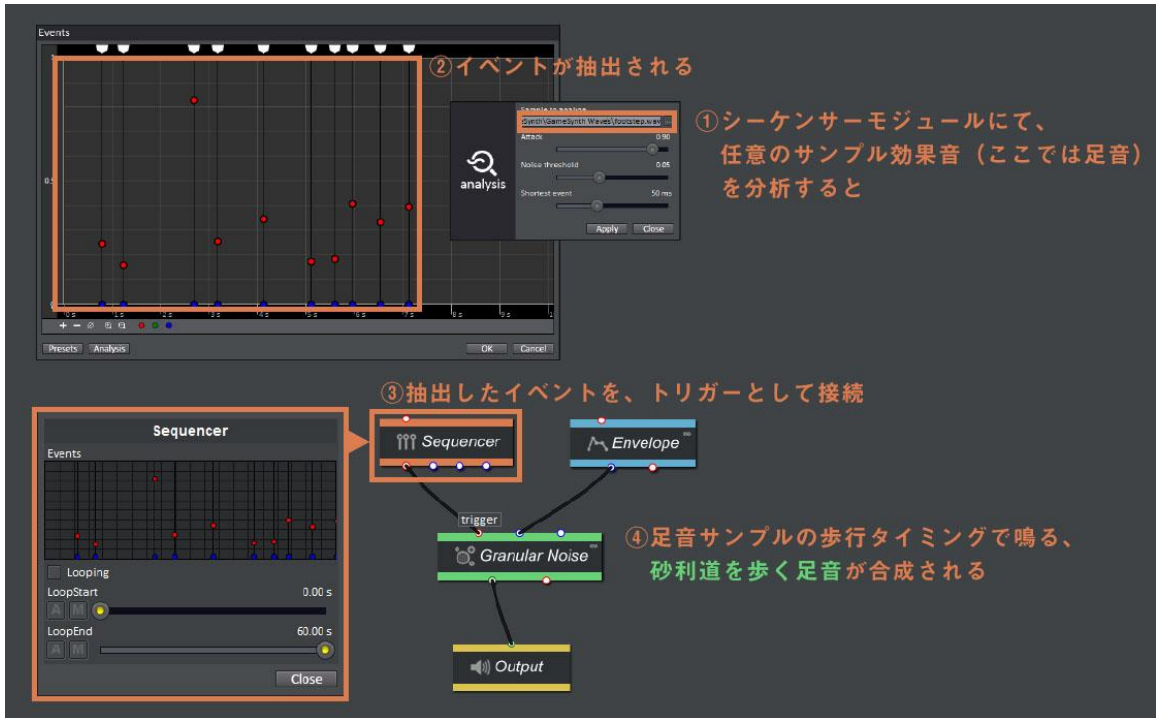
例えば下図のように、Modular モデルに搭載されている Curve モジュールは、サンプル効果音からの振幅、ピッチ、雑音エンベロープの抽出が可能であり、結果として得られたエンベロープを使用して、他のモジュールのパラメーターを制御することができます。



Curveモジュールにて動物の鳴き声の効果音サンプルを分析することでカーブ情報を抽出し、それを利用して緑のGlottisモジュールの振幅とピッチをコントロールしている

結果として、有機的なゆらぎをもつボイス音を作成されている

また以下の例では、シーケンサーモジュールでサンプル効果音からイベント（音の発生タイミング）を抽出し、その結果を用いて**砂利を歩くような足音**を作成しています。



①シーケンサーモジュールにて、任意のサンプル効果音（ここでは足音）を分析すると

②イベントが抽出される

③抽出したイベントを、トリガーとして接続

④足音サンプルの歩行タイミングで鳴る、砂利道を歩く足音が合成される

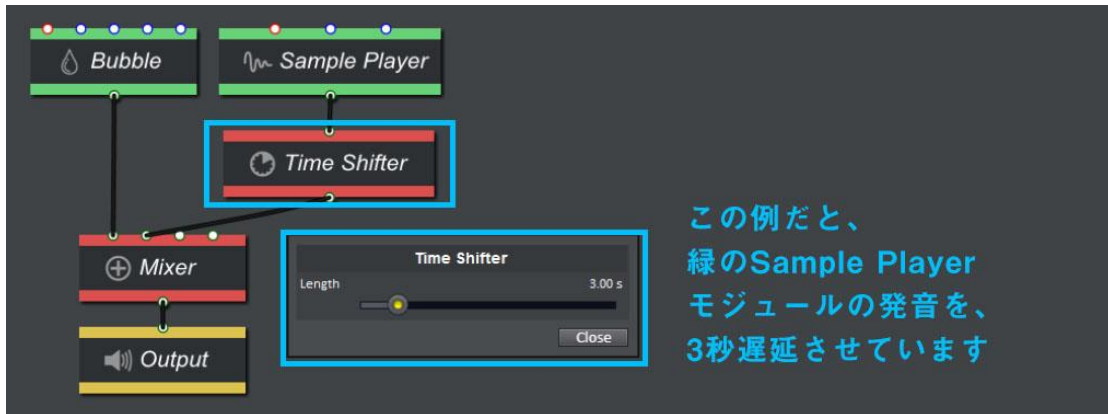
2.7

複数の音のタイミングを
うまく合わせたい

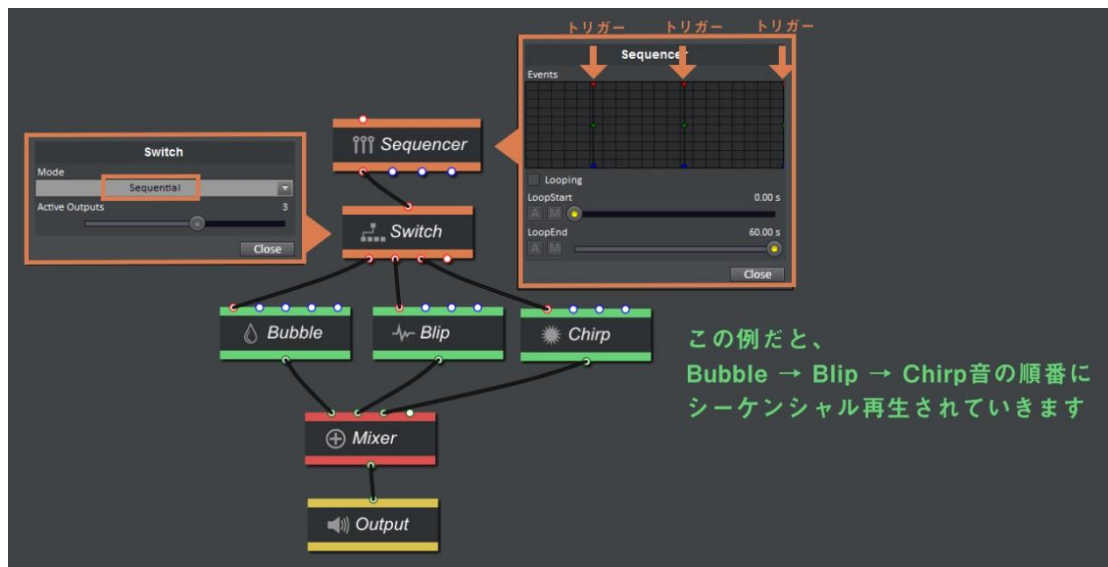


タイムシフターで、
音のタイミングを合わせよう

複数のジェネレーターを組み合わせる場合、タイムシフターモジュールを使うと**タイミングを合わせたり、遅らせること**ができます。



またシーケンサーとスイッチを使い合わせることで、複数の効果音のシーケンス再生も可能です（スイッチモジュールはランダム・ランダムノーリピート・シャッフル設定も可能です）。



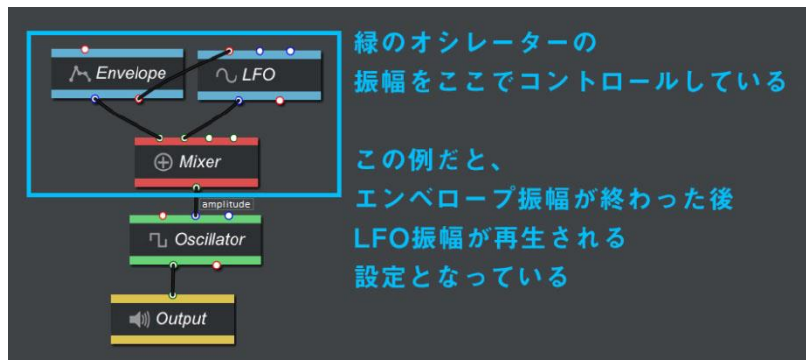
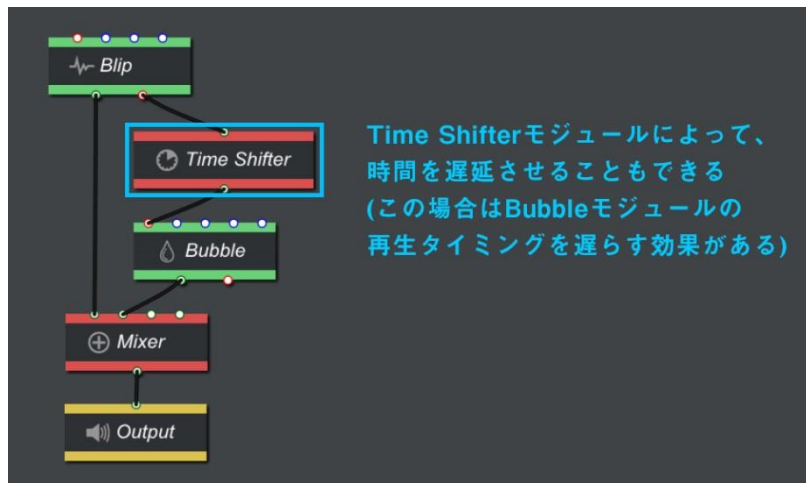
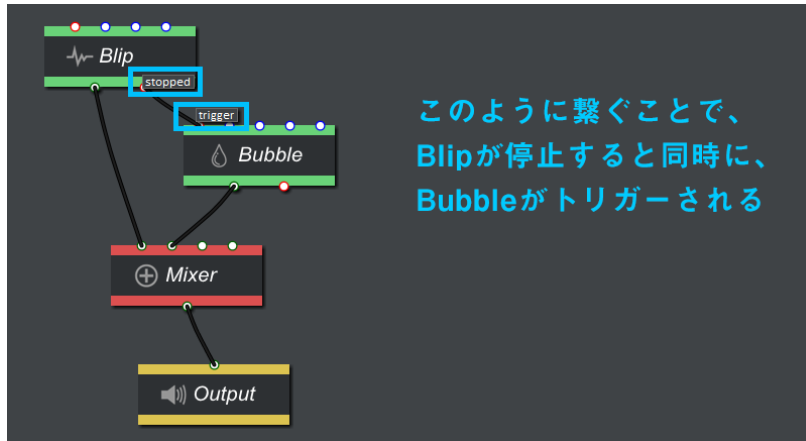
2.8

複数の音要素をシーケンス
制御させたい



ストップドシグナルで
音を順次再生させよう

これまで複数の音のタイミングを合わせる説明をしてきましたが、各モジュールに備わっている**ストップドシグナル**を使ってモジュールを結びつけると、「**Aが終わった後にBを再生する**」といった**シーケンス**を作ることができます。



2.9

音の鳴りをさらに複雑に
したい

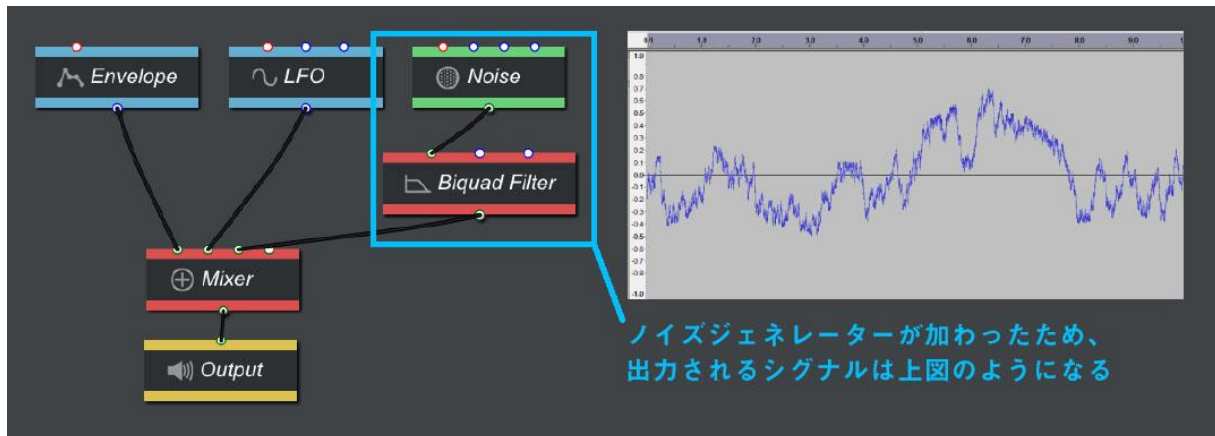


コントロールシグナル同士を
重ね合わせてみよう

以下のように複数の制御信号を組み合わせて、複雑なシグナルを作ることができます。



エンベロープとLFOシグナルの重ね合わせを行っているため、出力されるシグナルは ← のようになる



ノイズジェネレーターが加わったため、出力されるシグナルは上図のようになる

2.10

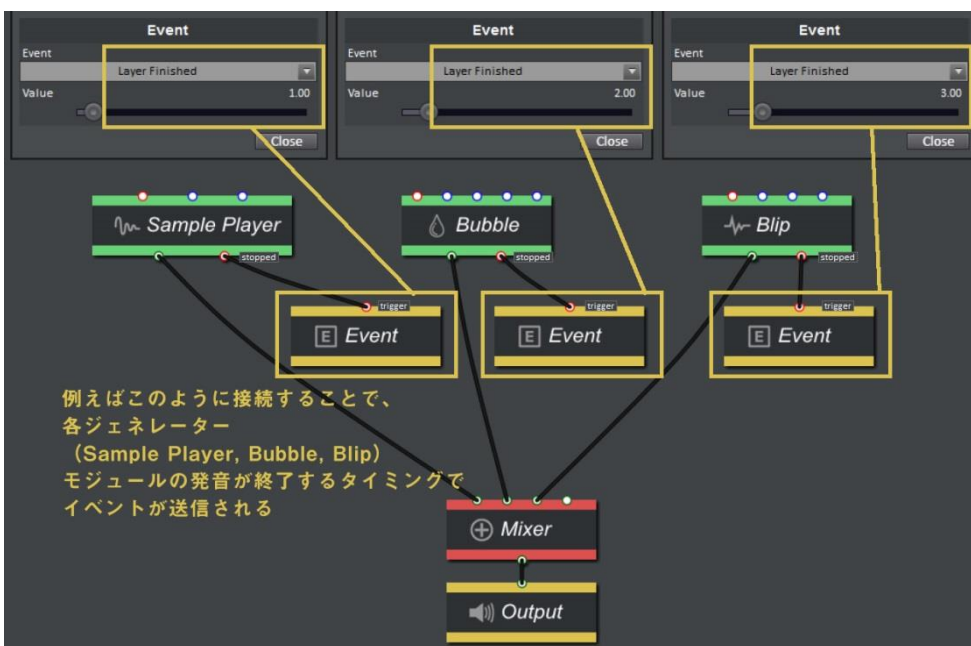
ランタイムエンジンで
トリガーを送信したい



イベントモジュールで
イベントを送信しよう

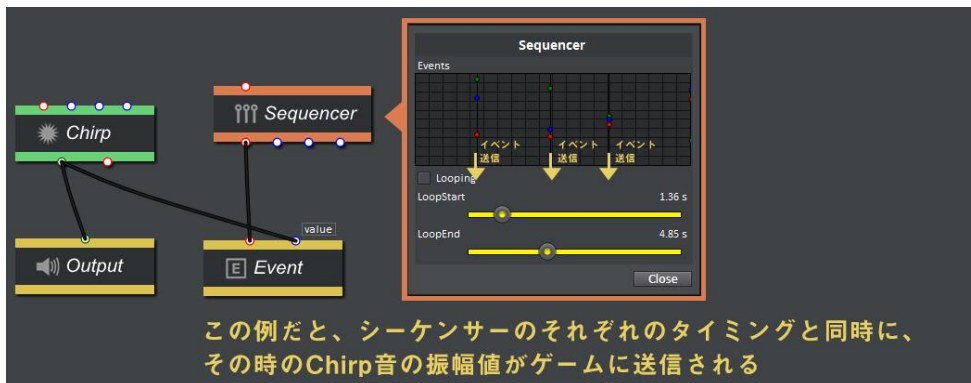
以下のテクニックは、GameSynth **ランタイムエンジン**を用いる際の際にのみ有効です。

下図に示す Event モジュールを使えば、**サウンド合成に伴うイベントをゲーム側に送信**することができます。



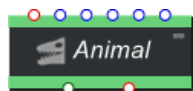
黄色のイベントモジュールは、「**値 1 のシグナルが送られてきた時、イベントをゲーム側に送信する仕組み**」になっています。また、**イベントとともに何らかの値をゲーム側に送る**ことも可能です。

たとえば以下のようにシーケンサーモジュールと接続することで、Chirp 音の振幅情報を決まったタイミングでゲームに送るといった使い方もできます。



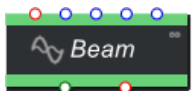
3. 搭載モジュール一覧

ジェネレーター



動物

様々な動物の鳴き声を発音
(グルル ウォーン ピュイー)



ビーム

2つのノコギリ波の加算による、
豊かなビーム音を発音
(ピーツ ピーツ ジーツ)



ブリップ

FM 合成による短い電子音を発音
(ピロツ ビュイン)



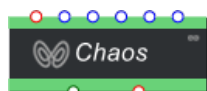
泡

泡音モデルのポリフォニック
モジュール (Distribution などの
イベントジェネレーターとの
併用を想定)
(ポコポコ ホワンホワン)



薬莢

銃から排出される薬莢音を発音
(チャリチャリン)



カオス

FM ライクな音からノイズまで、
幅広いカオスサウンドを発音



パチン

ノイズ成分の多い短い破裂音を
発音
(パチン パーン カーン)



金属板の衝突

金属の板の衝突音を発音
(ガコン ガン)



クリーチャー

クリーチャーの声帯の基本音を発音
(ウェー メー)



電話

電話で使用されるデュアルトーン合
成信号音を発音
(ピツ ポツ)



電気

アーク放電のような音を発音
(ジジジ)



炎

炎の音を発音
(ポオオ)



旗

旗や布のはためく音を発音
(バサバサ)



足音

様々な靴、地面の足音を発音
(ザッ コツツ ジャリッ トツ)



摩擦

岩、木、金属、ガラスの摩擦音を作成
(ガラガラ ズルズル ゴゴゴ)



きしみ

木、線維、ゴムなどのきしみ、ねじれ
音を発音
(ギギギ ギチギチ)



歯車

大小さまざまな歯車の音を発音
(ガタンガタン ガタタタ)



声紋

哺乳類の声紋のような音を発音
(ブー ウー ツー)



液体

水、油、スライムなどの液体音を発音
(ピシャピシャ ピチャピチャ)



グラニューラーノイズ
粒感のあるノイズ音を発音
(ザラザラ)



グラニューラープレイヤー
複数のサンプル効果音を登録し、
ランダムに粒子再生



ガンフォーリー
トリガー、マガジンなどの銃の
操作音を発音
(カチャッ ジャキッ パチン)



ガンショット
様々な銃の発砲音を発音
(バン パン)



雹
雹が地面にあたる音を発音
(パラパラ ポチャポチャ)



衝突
共鳴モードによる物体の
衝突音を発音(カン コン)



草木のそよぎ
木の葉や茂みのそよぎ音を発音
(ザワザワ サラサラ)



装置
工業機械の動作音を発音
(ビーン ウィーン)



メガソー
デチューンされた複数の
のこぎり波を発音(ビー)



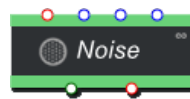
モード (共鳴)
モーダル合成による共鳴音を発音
(カターン ポン ピーン)



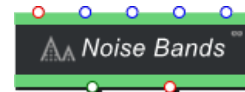
モーター
小型モーターの音を発音
(ウィーン)



ウェーブテーブル
位置・スピードをコントロール
できるサンプル再生機能



ノイズ
ホワイトノイズ&フィルターによる
周波数バンドノイズを発音
(ザアア)



ノイズバンド
変動するノイズバンドを発音
(キューーン シュイーン)



オシレーター
正弦波、ノコギリ波等の音を発音
(ピッ ピーッ ツー ブー)



パッチプレイヤー
他のパッチを再生



弦をはじく音
Karplus-Strong アルゴリズムによる
弦の音を発音(ピーン パチッ)



雨
様々なタイプの雨音を発音
(ザアア)



岩
小石から大石まで、大小さまざまな岩
石音を発音(ゴロゴロパチパチ)



サンプルプレイヤー
サンプル効果音を再生



スクリプト
スクリプトによる発音



サインバンク
64 のオシレーターの複合音を発音
(ビー)



スチーム
蒸気音(プシューッ)



雷
遠近さまざまな雷鳴を発音
(ゴロゴロ ズガーン)



風
穏やかな風から台風まで、あらゆる
風音を発音(ヒュオオ…)

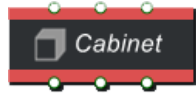
プロセッシング



オールパスフィルター
位相変化（聴覚上変化はなし）



バイカッドフィルター
周波数帯をカットする



キャビネット
箱の中で鳴っている効果を出す



コーラス
コーラス効果を出す



クリッパー
振幅の高低を切って音をつぶす



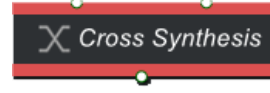
コムフィルター
自分自身で干渉効果を生む



コンプレッサー
音圧を上げる



コンボルバー
2つの音をコンボリューション



クロスシンセシス
2つの音を混ぜる



デグレーダー
音質を劣化させる



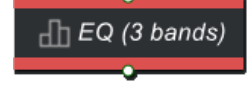
ディレイ
やまびこ効果



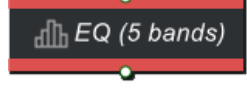
ドップラー
ドップラー効果を出す



エンベロープフォロワー
振幅からカーブを得る



EQ3 バンド
3バンドイコライザー



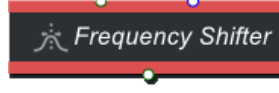
EQ5 バンド
5バンドイコライザー



フランジャー
フランジャー効果



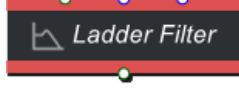
フォルマントフィルター
発音を変える



周波数シフター
周波数を変える



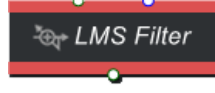
ゲイン
音量調整



ラダーフィルター
ローパスレゾナントフィルター



リミッター
音のレベルを抑える



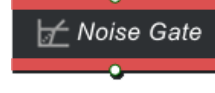
LMS フィルター
可変フィルター



ルーパー
音を繰り返す



ミキサー
音をミックスする



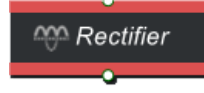
ノイズゲート
小さな音を排除する



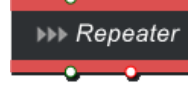
パナー
音の左右の大きさ



ピッチシフター
音の高さを変える



整流器
音の位相の正負を制限する



リピーター
波形を繰り返す



レゾネーター
特定周波数帯に共鳴部位を設ける



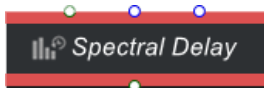
EQ フィルター
シンプルな EQ フィルター



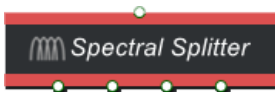
リングモジュレーター
音を金属的にする



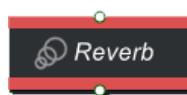
サチュレーター
迫力を増す



スペクトラルディレイ
周波数ごとにディレイを設ける



スペクトラルスプリッター
周波数帯ごとにラインを分割する



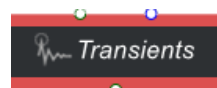
リバーブ
洞窟のようなリバーブ効果



スレッシュホールド
一定レベル到達時にトリガーを送る



タイムシフター
音を遅延させる



トランジェント
音の立ち上がりを強調する



チューブ
管の中で鳴っている効果を出す



ボコーダー
ロボット声効果

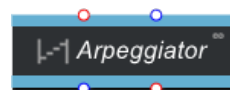
コントローラー



ADSR
ランダム化可能な ADSR
エンベロープ



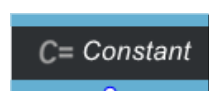
算術
二つの値を加減乗除



アルペジエーター
音階を作成



オートメーションカーブ
カーブを作成



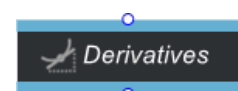
定数
定数を与える



CSV データ
CSV ファイルから
コントロールを作成



カーブ
サンプル音を分析してカーブを抽出



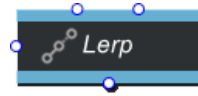
微分
カーブから
速度・加速度を求める



エンベロープ
セグメントカーブを作成



進化
自然現象のように
つねに変化していくカーブ



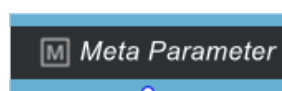
線形補間
2つのシグナルの
線形補間値を求める



LFO
低周波オシレーター



マッパー
カーブ定義に従って、
入力シグナルを変換する



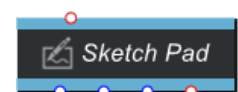
メタパラメータ
メタパラメータを使用



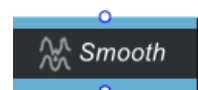
パーリンノイズ
パーリンノイズを発する



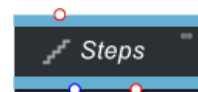
スケールオフセット
一定値を乗除、
一定値を加減



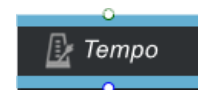
スケッチパッド
X、Y、筆圧で
音をコントロール



スムーズ
入力シグナルをなめらかな
カーブに変える



ステップ
階段状のコントロールを
作成

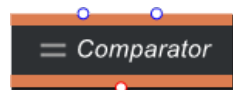


テンポ
BPM を分析

ロジック



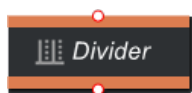
クロック
連続してトリガーする
(ランダムさを調整可能)



比較
比較して条件を
満たしたらトリガーする



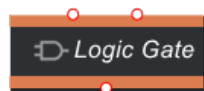
分配
連続してトリガーする
(ランダムな時間で)



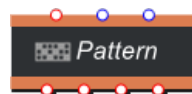
分割
トリガー群の一部を
カットする



歩行
2～10足の動物の
足音をトリガーする



ロジックゲート
論理に従ってトリガーする



パターン
ビートを作る



リトリガー
トリガーを受けたら
増やして再トリガーする



セレクター
複数トリガーから
ひとつのラインへ



シーケンサー
トリガー群の一部を
カットする

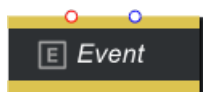


スイッチ
ひとつのラインから
複数のトリガーへ

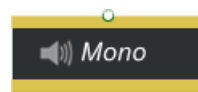


トランジション
マトリックスでトリガーを
分配する

アウトプット



イベント
イベントを送信



モノラル
モノラルアウトプット



ステレオ
ステレオアウトプット

4. ショートカットキー

選択したモジュールをカット	Ctrl + X
選択したモジュールをコピー	Ctrl + C
選択したモジュールをペースト	Ctrl + V
選択したモジュールを削除	Delete
全モジュールを選択	Ctrl + A
全モジュールを選択解除	Ctrl + D
使用していない全モジュールを選択	Ctrl + U
全フロートウィンドウを閉じる	Ctrl + Shift + W
コンテキストメニューを開く	Menu
モジュール群を画面中央に表示	Home
選択モジュールを移動	Arrows
(ケーブルドラッグ時) 最寄りの穴へ自動的にケーブルを繋げる	Return
マウスカーソル下のケーブルを切断	Alt + Click
ランダムにバッチを生成	Ctrl + P
パラメーターウィンドウの表示方法を変更	Ctrl + Tab
選択したモジュール群を左揃え	Ctrl + ←
選択したモジュール群を右揃え	Ctrl + →
選択したモジュール群を上揃え	Ctrl + ↑
選択したモジュール群を下揃え	Ctrl + ↓
選択したモジュール群を中央揃え	Ctrl + *
ジェネレーター用コンテキストメニュー	G
プロセッサ用コンテキストメニュー	P
コントロール用コンテキストメニュー	C
ロジック用コンテキストメニュー	L
アウトプット用コンテキストメニュー	O
<i>Bubble</i> モジュールを出す	Ctrl + M, Ctrl + B
<i>Ladder Filter</i> モジュールを出す	Ctrl + M, Ctrl + F
<i>Gain</i> モジュールを出す	Ctrl + M, Ctrl + G
<i>LFO</i> モジュールを出す	Ctrl + M, Ctrl + L
<i>Mixer</i> モジュールを出す	Ctrl + M, Ctrl + X

<i>Modes</i> モジュールを出す	Ctrl + M, Ctrl + M
<i>Noise</i> モジュールを出す	Ctrl + M, Ctrl + N
<i>Oscillator</i> モジュールを出す	Ctrl + M, Ctrl + O
<i>Output</i> モジュールを出す	Ctrl + M, Ctrl + U
<i>Reverb</i> モジュールを出す	Ctrl + M, Ctrl + R
<i>Sample Player</i> モジュールを出す	Ctrl + M, Ctrl + S
<i>Automation Curve</i> モジュールを出す	Ctrl + M, Ctrl + A
<i>Curve</i> モジュールを出す	Ctrl + M, Ctrl + C
<i>Envelope</i> モジュールを出す	Ctrl + M, Ctrl + E
<i>Meta Parameter</i> モジュールを出す	Ctrl + M, Ctrl + P
<i>Distribution</i> モジュールを出す	Ctrl + M, Ctrl + D
<i>Sequencer</i> モジュールを出す	Ctrl + M, Ctrl + Q