

KYMA PROTOTYPES

Kyma 5 Prototypes in a Nutshell - French Translating. © 2004 Sylvain KEPLER - <http://sylvain.kepler.free.fr>

Kyma 5 - Description Sommaire des Prototypes - Traduction Française.

Algorithms	CellularAutomaton 	Ce module est basé sur le principe de l'automata cellulaire a une dimension decrit par Stephen Wolfram au travers de son livre Theorie et application de la cellule Automata. L'Etat, un nombre binaire de niveau n, ou n est le nombre d'entrées, détermine laquelle des entrées est activée et laquelle est désactivée pour une génération donnée. Un entier, qui fait figure de règle, est utilisé pour déterminer l'état suivant.
Algorithms	ContextFreeGrammar 	Ce Module qui se compose de Concatenations des entrées et de CenteringMixers est générée à partir d'une expression de départ qui est redéfinie selon des règles de production fournies par une grammaire de contexte libre. Une "graine" est utilisée pour reproduire les résultats.
Algorithms	LimeInterpreter 	Lit des fichiers numériques produits par le programme de notation musicale Lime et envoie les valeurs vers les paramètres des sons Kyma. Cela vous permet de jouer des morceaux produits dans Lime en utilisant les sons Kyma en tant qu' instruments.
Algorithms	ParameterTransformer 	Les parametres de l'entrée peuvent être établis ou altérés par les formules écrites dans les champs de Transformation (pour plus de détails, voir les tutoriaux et chapitres correspondants dans ce livre). Toutes les transformations prennent place symboliquement (en d'autres termes il ne s'agit pas de réelles transformations par un traitement de signal mais de transformations appliquées aux champs de paramètres avant que l'entrée ne soit compilée et chargée dans le processeur de signal_ avant qu'il ait commence à générer le son).
Algorithms	RandomSelection 	Sélectionne des sons depuis l'espace d'échantillons de toutes les entrées et leur affecte des départs de lecture qui sont générés aléatoirement selon une distribution exponentielle de durées de retard et dont la moyenne est spécifiée par le paramètre AverageDelay. La valeur entrée dans le champ Itérations représente le nombre total des sons. Des graines (Seeds) sont fournis de telle sorte que l'effet se reproduise.
Algorithms	RhythmicCellularAutomaton 	Ce module est base sur le principe de l'automata cellulaire à une dimension décrit par Stephen Wolfram au travers de son livre Théorie et application de la cellule Automata. L'Etat, un nombre binaire de niveau n, ou n est le nombre d'entrées, détermine laquelle des entrées est activée et laquelle est désactivée pour une génération donnée. Un entier, qui fait figure de règle, est utilisé pour déterminer l'état suivant. Ce module differt du CellularAutomaton par le fait que l'état est interprété horizontalement (rythmiquement) plutot que verticalement (harmoniquement). Chaque génération de l'automaton est interprète en tant que motif rythmique ou 1 signifie que l'entrée est activée et 0 signifie que l'entrée est désactivée.
Algorithms	Script 	Un script est une manière abordable de construire des sons algorithmiquement (plutot que de les agglomerer graphiquement dans le sound Editor). La structure construite sera un melangeur (Mixer) de quelques entrées, chacune ayant son propre temps de départ (ou TimeOffset). Un script est comme n'importe quel autre module en ce sens qu'il peut être lui-même utilise en tant qu'entrée pour élaborer un son plus complexe. Par exemple, un script peut contenir des variable et peut être ensuite utilise en tant qu'entrée d'un autre script.
Analysis	AnalysisFilter 	Ce filtre construit pour isoler individuellement des harmoniques. Il a une sortie en quadrature (le canal gauche est le cosinus du canal droit a la frequence donnee, le canal droit est le sinus de l'entrée a la fréquence donner). Ainsi vous pouvez utiliser le QuadratureOscillator (oscillateur en Quadrature) pour transposer en fréquence cette harmonique en utilisant une modulation en anneau a bande latérale. Le filtre est construit de telle sorte que si vous additionniez la sortie des filtres réglés sur chacune des harmonique depuis 0 jusqu'a l'harmonique la plus proche de la moitié de la fréquence d'échantillonnage (fréquence de nyquist) , l'amplitude de cette somme serait 1.

KYMA PROTOTYPES

Analysis	FFT 	La FFT prend un signal fonction du temps (signal audio) et génère un signal de sortie fonction des fréquences ou l'inverse. Lenght (la durée) est la durée de la FFT. Deux signaux fonction du temps indépendants sont présents : celui du canal gauche et celui du canal droit. Le signal fonction des fréquences se répète chaque Lenght/2 échantillons, en alternant entre le spectre du signal fonction du temps du canal gauche et le signal fonction du temps du canal droit. Le signal fonction des fréquences est fournit dans l'ordre des fréquences / 0Hz, Fs / Length, 2-Fs / Length, etc...Chaque échantillon fonction des fréquences trouve sa partie réelle dans le canal gauche est sa partie imaginaire dans le canal droit. Lorsque l'option Inverse n'est pas active, l'entrée est un signal fonction du temps et la sortie est un signal fonction de la fréquence. Lorsque l'option Inverse est active, l'entrée est une signal fonction des fréquences et la sortie est un signal fonction du temps.
Analysis	FrequencyTracker 	Exécute perpétuellement une estimation de la fréquence du signal d'entrée. Etablissee la gamme des fréquences (de MinFrequency a MaxFrequency) pour être aussi proche que possible de ce que vous savez à propos de tessiture de l'instrument ou de la voix que vous traquez. Il est recommandé que vous laissiez Confidence, Scale, Emphasis, et Detectors sur leurs valeurs par défaut jusqu'à ce que vous ayez obtenu un tracker de fréquence raisonnablement correcte pour un signal d'entrée donné. Ensuite , si vous voulez, faire un accordage fin du tracker, expérimentez des petites variations sur ces valeurs, une à la fois (en commençant par accroître le nombre des détecteurs) de sorte que vous soyez certain de l'effet que chacun de ces paramètres exercera sur le tracking. S'il en résulte que le tracking est altéré e ou amélioré, revenez sur les valeurs de départ. La sortie du FrequencyTracker s'échelonne entre 0 et 1 : pour utiliser cette valeur dans le champs des fréquence, multipliez le par la fréquence maximale possible : SignalProcessor sampleRate hz * 0.5.
Analysis	LiveSpectralAnalysis 	Ce module devra être utilisé en tant que paramètre agissant sur le spectre de l' OscillatorBank (banque d'oscillateurs). Ils analysent l'entrée et produit des enveloppes de fréquence et d'amplitude pour contrôler la banque d'oscillateurs de l' OscillatorBank .
Analysis	OscilloscopeDisplay 	Affiche la sortie à la manière d'un oscilloscope sur la surface de contrôle virtuelle. Utiliser les boutons en bas de l'écran pour zoomer en temps ou en amplitude. La valeur affichée au niveau du curseur (ou se croisent les barres rouges du viseur) est affichée dans le coin supérieur gauche. Cliquer sur l'écran permet de figer l'affichage en cours de telle sorte que vous puissiez bouger le curseur librement sur la courbe pour lire des valeurs exactes. Un oscilloscope peut être placé à n'importe quel niveau de la chaîne d'un traitement du signal. Si une structure dispose de plus d'un oscilloscope, toutes les traces seront affichées cote à cote dans la fenêtre de surface de contrôle virtuel. Vous pouvez aussi voir la trace de l'oscilloscope de n'importe quel son depuis le menu Info. (La méthode depuis le menu Info ne permet d'observer qu'un signal a la fois et ne vous permet pas d'ajuster le trigger pour obtenir un affichage stable).
Analysis	PeakDetector 	Extrait une enveloppe d'amplitude à partir du signal d'entrée en traquant la valeur absolue de ce dernier. Il répond aux variations de volume dans la limite du paramètre Attacktime et répond également à la décroissance de l'amplitude du signal d'entrée dans la limite du paramètre ReleaseTime. Scale est un atténuateur qui permet d'agir sur l'amplitude du signal d'entrée avant son traitement.
Analysis	RMSSquared 	Ce module peut être utilise pour obtenir une estimation de l'amplitude du signal d'entrée. La sortie vaut : $\text{Input}^2 * \text{timeConst} + \text{prev} * (1 - \text{timeConst})$ C'est la racine carre de l'entrée moins la racine carre finale du signal de sortie.

KYMA PROTOTYPES

Analysis	SpectrumAnalyzerDisplay 	C'est un analyseur de spectre en temps réel. Il affiche le spectre du signal d'entrée sur la surface de contrôle virtuel. Utiliser les boutons situés en dessous l'afficheur pour zoomer en fréquence et en amplitude. La valeur affichée au niveau du curseur (ou se croisent les barres rouges du viseur) est affichée dans le coin supérieur gauche. Cliquer sur l'écran permet de figer l'affichage en cours de telle sorte que vous pouvez bouger le curseur librement sur la courbe pour lire des valeurs exactes. Un analyseur de spectre peut être placé à n'importe quel niveau de la chaîne d'un traitement de signal. Si une structure dispose de plus d'un analyseur , tous les spectres seront affichées côte à côte dans la fenêtre de surface de contrôle virtuel. Vous pouvez aussi la trace du spectre de n'importe quel son depuis le menu Info. (La méthode depuis le menu Info ne permet d'observer qu'un spectre à la fois et ne vous permet pas d'ajuster la longueur de la FFT sinon seulement en le faisant depuis les Preferences).
Arithmetic	AbsoluteValue 	La sortie du module AbsoluteValue est la valeur absolue de son entrée. (choisir Full waveform depuis le menu Info pour voir si la forme d'onde de sortie contient effectivement seulement des valeurs positives plutôt qu'alternatives).
Arithmetic	ArcTan 	La sortie d'ArcTan est l'arc tangente à quatre quadrants du rapport du signal droit par rapport au signal gauche.
Arithmetic	Difference 	Extrait la différence entre les entrées Input et minusInput.
Arithmetic	Equality 	A chaque fois que InputA est égal à InputB (+ ou - une certaine tolérance donnée), la sortie de ce module vaut 1. Dans tout les autres cas, la sortie vaut 0.
Arithmetic	PhaseShiftBy90 	C'est une combinaison de deux filtres réglés pour obtenir un déphasage exact de 90 degrés entre le canal gauche et le canal droit pour une fréquence donnée (en transposant de 45 degrés en avance le canal gauche et en transposant de 45 degrés retard le canal droit par rapport au signal d'origine). Utilisez la fonction expand sur le prototype SingleSideBandRM pour obtenir un exemple sur l'utilisation de ce module en tant qu'enveloppe d'un QuadratureOscillator pour réaliser une modulation en anneau à bande latérale. Note : Ces filtres sont très sensibles à l'amplitude du signal d'entrée. Essayez d'atténuer l'amplitude d'entrée par un facteur 0.05 puis en l'augmentant progressivement jusqu'à ce que vous entendiez une distorsion puis en le reculant un peu. (Il est utile également de jeter un coup d'œil sur la sortie du module PhaseShiftBy90 en utilisant l'oscilloscope du menu Info tout en ajustant l'amplitude du signal en entrée).
Arithmetic	Product 	Exécute le produit de ses entrées. S'il est question de deux entrées audio, cela donne une modulation en anneau. Si une des entrées est un infra-son et si l'autre entre est un signal audio ordinaire, l'effet produit sera le même qu'une modulation d'amplitude par la forme d'onde de l'infra-son
Arithmetic	RunningMax 	La sortie de ce module scrute les maxima d'amplitude des signaux présents sur l'entrée et conserve le plus haut niveau observé depuis le début de la lecture. Pour l'initialiser à 0 et recommencer le traitement du running max, donner une valeur différente de 0 à Reset. Du début à la fin d'un son, si aucun Reset n'est appliqué, la valeur de sortie donnera le plus haut niveau observe parmi tous les échantillons ayant été appliqués en entrée.
Arithmetic	RunningMin 	La sortie de ce module donne l'amplitude minimale rencontrée sur l'entrée depuis le début du traitement jusqu'à l'instant ayant court. Pour initialiser et recommencer le traitement du running min, établissez le Reset à zéro. A chaque reset, le minimum observé est alors ignoré et le traitement reprend. Du début à la fin d'un son, si aucun reset n'est exécuté, la valeur de sortie donne le plus bas niveau observé parmi tous les échantillons ayant été appliqués en entrée.

KYMA PROTOTYPES

Arithmetic	ScaleAndOffset 	La sortie de ce module est égale à : $(\text{Input} * \text{Scale}) + \text{Offset}$. Cela peut être utile pour modifier la valeur minimale et la dynamique d'un signal de contrôle avant de l'utiliser pour moduler un son, par exemple lors de l'étalonnage et du réglage d'offset des canaux gauche et droit en sortie d'un module SpectrumFromRAM avant qu'ils ne soient envoyés vers l'OscillatorBank. (Cependant, pour ces cas où le signal de contrôle est appliqué directement dans le champs de paramètre, il sera plus simple d'y inscrire une expression arithmétique pour étalonner une grandeur).
Arithmetic	SetRange 	Ceci permet de recadrer la dynamique d'un signal appliqué en entrée selon deux valeurs newMin et newMax.
Arithmetic	SqrtMagnitude 	C'est la racine carrée de la somme du signal gauche et droit élevés au carré. Si la racine carrée de la somme des carrés est plus grande que 1, le son sature à 1. Cela peut être utile pour réaliser une analyse spectrale ou le canal gauche est défini pour être la partie réelle et le canal droit la partie imaginaire d'un nombre complexe. Vous pourriez aussi utiliser ceci comme une étrange sorte de mesure des distances instantanées entre deux signaux, l'un placé à gauche et l'autre à droite.
Arithmetic	Threshold 	La sortie vaut 1 lorsque l'amplitude du signal d'entrée dépasse le seuil spécifié, sinon la sortie vaut 0. Plus la valeur d'hystérésis est petite, plus le seuil est sensible aux variations momentanées du signal d'entrée. Si vous essayez donc de détecter un dépassement d'amplitude, il est habituellement plus judicieux d'envoyer votre signal d'entrée d'abord vers un module AmplitudeFollower ou un module PeakDetector de telle sorte que vous détectiez les variations de l'amplitude de l'enveloppe plutôt que de scruter les échantillons un par un.
Combiners	OverlappingMixer 	Mixeur qui chevauche les moments de départ de ses entrées selon le paramètre OverlapTime spécifié.
Delays	DelayWithFeedback 	Délai ordinaire avec feedback ajustable.
Delays	Eugenio Reverb 	Eugenio Giordani's réverbération algorithm.
Delays	EUverb LITE 	Une version de EUverb avec quelques retards nécessitant moins de puissance de calcul.
Delays	EuverbLeft 	Utiliser ce module conjointement avec EuverbRight pour créer une réverb stéréo. Très utile avec la Timeline ou avec un module à sorties multiples où le canal gauche et le canal droit doivent être traités en tant que sons indépendants. Rejoignez ensuite le canal gauche et le canal droit en utilisant le module ChannelJoin, la Timeline, le mixer ou un des modules à sorties multiples.
Delays	EuverbMono 	Ce module est comme la Euverb excepté que l'effet est mono et n'a pas le signal direct ré-injecté avec le signal réverbéré. Cela peut être utile avec la Timeline lorsque vous avez le son direct déjà présent sur une autre piste et que vous le routez vers la réverbération en utilisant un submix.

KYMA PROTOTYPES

Delays	EuverbRight 	Utilisez le conjointement avec EuverbLeft pour créer une réverbération stereo. Très utile dans la Timeline ou dans un module à sorties multiples où le canal gauche et le canal droit doivent être traités indépendamment. Rejoignez les canaux gauche et droit en utilisant le module ChannelJoin, la Timeline, le Mixer ou un des modules à sorties multiples.
Delays	ReverbSection 	Identique au module DelayWithFeedback excepté que les propriétés sont spécifiées en terme de temps de décroissance (DecayTime), c'est à dire le temps nécessaire pour que l'entrée retardée et l'entrée ré-injectée se taisent 60dB en dessous leurs niveaux initiaux. Vous pouvez utiliser des combinaisons de ces modules avec d'autres pour construire vos propres algorithmes de réverbération.
Delays	TimeControl 	Ce module "Ralentit" ou "Accélère" la progression du temps sur son entrée en contrôlant la façon avec laquelle le compteur de temps est incrémenté par le calculateur. Cela affecte uniquement le moment de départ des événements de l'entrée (ex : si l'entrée est un script ou une concaténation ou contient des TimeOffsets), pas la fréquence de lecture des échantillons.
Delays	TimeOffset 	Retarde le temps de départ de l'entrée par la valeur SilentTime spécifiée. Si la fonction Rétrograde ou Reverse est activée, une Constante zéro est concaténée à la fin de l'entrée; cela peut être utile pour additionner du silence à la fin d'une entrée acheminée vers une réverbération ou un écho de telle sorte que cela puisse donner le temps à la réverbération de se taire.
Delays	TimeStopper 	Permet à l'entrée être chargée dans le processeur de signal, être jouée puis être arrêtée dans le temps. Le temps se "fige" seulement lorsque la valeur située dans le champ Résumé devient différente de zéro. Par exemple, même si l'entrée a une durée d'un seul échantillon, ce dernier dure jusqu'à ce que Résumé soit différent de 0. Si l'entrée contient de multiples événements qui se manifestent séquentiellement, seulement le premier aura lieu immédiatement mais les événements suivants se produiront seulement après que Résumé soit différent de zéro.
Delays	WaitUntil 	Ce module ne démarre pas le son présent sur son entrée tant que la condition Résumé demeure invérifiée. Cela ressemble au TimeStopper sauf qu'un WaitUntil ne laissera pas non plus sa propre entrée débiter tant que la valeur Résumé est égale à 0 (Par contre TimeStopper laisse son entrée démarrer mais ne la laisse pas se terminer tant que Résumé est égal 0.)
Disk	DiskCache 	Enregistre l'entrée sur le disque dur à l'appui sur le bouton record. Lorsque le bouton record est relâché, l'entrée est rejouée à partir du disque plutôt que de l'être en temps réel. Cela peut être utile lorsque vous tentez de réduire les quantités de calculs dsp requises par une structure contenant une grande quantité de modules. Quelque soit le moment où vous effectuez un changement sur l'entrée, n'oubliez pas de cliquer sur record et de recapturer la nouvelle version de l'entrée sur le disque.
Disk	DiskPlayer 	Ce module rejoue des enregistrements à partir du disque depuis le nom de fichier spécifié dans le champ FileName , débute la reproduction au moment spécifié dans le champ FilePosition et pour une durée déterminée par Duration et à la vitesse RateScale. A chaque fois que Trigger prend une valeur positive, le jeu reprend depuis FilePosition et joue l'enregistrement à nouveau. Pour traiter ceci comme une lecture d'échantillon contrôlé depuis un Clavier MIDI, donnez à FileName le nom de l'échantillon voulu, donnez à Duration la durée pour laquelle vous tenez à entendre l'échantillon. Affectez au paramètre Trigger la valeur !Keydown (ou interrupteur MIDI) et réglez RateScale à la vitesse à laquelle vous voulez lire l'échantillon. Par exemple : !Pitch hz / 2 a hz si vous utilisez un Clavier MIDI pour contrôler un échantillon dont la vitesse d'enregistrement valait 2 a.
Disk	DiskRecorder 	Enregistre son entrée sur le disk pour une durée CaptureDuration lorsque son Trigger devient positif.

KYMA PROTOTYPES

Disk

MultifileDiskPlayer



Ce module est similaire au DiskPlayer sauf que vous aurez à spécifier un groupe de noms de fichiers plutôt qu'un nom de fichier unique. La valeur Index détermine quel fichier sera joué à chaque redeclenchement (l'index 0 choisit le premier fichier dans le groupe, l'index 1 choisit le second etc..). Un seul fichier sera joué à la fois, mais le choix du fichier peut être effectué en temps réel. (pour obtenir plus de polyphonie, cablez ce module vers le module MIDIVoice et établissez la polyphonie désirée).

Vous pouvez utiliser une vitesse de lecture commune pour tous les fichiers du disque, mais vous pouvez aussi éventuellement faire que la vitesse soit fonction de l'Index de fichier si vous voulez jouer différents fichiers à différentes vitesses.

Le MultiFileDiskPlayer peut être utilisé à chaque fois vous avez besoin d'un accès aléatoire en temps réel vers différents enregistrements sur disque via un clavier Midi ou un contrôleur Midi. Par exemple il peut être utilisé pour choisir un jeu d'effets sonores en live et les synchroniser à la main pour un illustrer un film, pour créer un « sampler-disque » avec un échantillon différent pour chaque touche du clavier, pour exécuter une composition constituée de longs enregistrements sur disque, ou (si le trigger est lié au module FrequencyTracker ou au module EnvelopeFollower) en tant que "morceau de bande" synchronisable qui réponds à un instrumentiste en temps réel.

Disk

SamplesFromDiskSingleStep



Tant que Trigger est supérieur à zéro, SamplesFromDiskSingleStep lit les échantillons depuis un fichier du disque. Si Trigger est inférieur ou égal à zéro, le dernier sample est interprété. Gate repositionne "la tête de lecture" en début du fichier.

Dynamics

DynamicRangeController



Ecrase ou Etire la dynamique du signal d'entrée en fonction du niveau de l'enveloppe du signal Sidechain.

Les niveaux d'entrée et de sortie seront les mêmes excepté lorsque le niveau de l'enveloppe Sidechain croise le seuil spécifié.

Si Compressor est sélectionné, l'entrée sera atténuée chaque fois que le niveau de SideChain dépasse le seuil.

Si Expander est sélectionné, l'amplitude du signal d'entrée sera boostée chaque fois que le niveau de SideChain tombe en dessous du seuil. Le rapport d'amplitude entre l'entrée et la sortie est à spécifier.

Les utilisations typiques sont :

Le Limiting (expansion avec un très grand taux et un seuil élevé)

Le Gating (expansion avec un taux très faible et un seuil élevé)

Le Ducking (régler attack et decay entre 1 et 2 secondes, placer le signal a "ducker" en entrée et placer le signal de contrôle sur SideChain, et mixer enfin le signal SideChain avec la sortie du module DynamicRangeController), rends les percussions brèves plus massives (compression puis augmentation du gain moyen), et adoucis les variations d'amplitude extrêmes (particulièrement utile lors d'enregistrements sur des médias avec une dynamique réduite tels que cassettes Audio et Vidéos).

Dynamics

MultiplyingWaveshaper



Multiplie l'entrée par une valeur lue dans la table d'onde (Wavetable) à l'index fourni par l'entrée NonlinearInput et atténue (ou amplifie) le résultat par le facteur Scale.

Peut être utilisé comme un contrôleur de dynamique ne nécessitant que peu de puissance de calcul si NonlinearInput est un signal envoyé vers un détecteur de crête ou un détecteur RMS et si l'entrée est ce même signal retardé d'une certaine durée. Dans ce cas de figure, la table d'onde caractérise la fonction d'atténuation de l'amplitude du signal de sortie par rapport au signal d'entrée.

Pour élaborer une nouvelle fonction entrée-sortie, ouvrir l'éditeur Sample/Wavetable et utiliser le modèle InputOutputCharacteristic pour générer une nouvelle fonction de transfert avec les paramètres de compression/expansion désirés.

Envelopes

ADSR



Génère une enveloppe ADSR traditionnelle à 4 segments. Aussi longtemps que le gate est à 0, aucune enveloppe n'est générée. Lorsque Gate devient positif, l'enveloppe est déclenchée et se déroule selon le cheminement bien connu. Dans un prototype avec un champ de paramètres d'enveloppe (oscillateur par exemple), vous pouvez bien entendu utiliser l'enveloppe pour contrôler un paramètre autre que l'amplitude.

Pour appliquer une enveloppe à n'importe quel son, utilisez un son et un module de génération d'enveloppe que vous placez aux entrées d'un module VCA (le VCA n'est autre qu'un multiplieur).

KYMA PROTOTYPES

Envelopes

AR



Identique au générateur ADSR mais ne comporte que deux phases, attack et release.

Envelopes

Chopper



Multiplie l'entrée par une salve de "grains" ou d'enveloppes (un cycle de la table d'onde sélectionnée), chacun durant une durée GrainDuration avec un InterGrainDelay de silence entre eux. Durant l'InterGrainDelay, l'entrée vaut 0.

Envelopes

FunctionGenerator



Lit une table d'onde spécifiée pour une durée OnDuration à chaque fois que Trigger est actif. Très utile pour la génération d'enveloppes ou pour la lecture d'enregistrements stockés dans une table d'onde en mémoire (voir aussi le prototype Sample). Dans un prototype comportant un champs de paramètres d'enveloppe (oscillateur par exemple), vous pouvez utiliser FunctionGenerator directement en tant que paramètre d'enveloppe. FunctionGenerator peut aussi être utilisé pour contrôler d'autres paramètres (tels que la fréquence ou la durée). Pour appliquer une enveloppe à n'importe quel son, utilisez un son et un module de génération d'enveloppe que vous placez aux entrées d'un module VCA (le VCA n'est autre qu'un multiplieur).

Envelopes

GraphicalEnvelope



Similaire à une enveloppe ADSR, sauf que vous pouvez spécifier graphiquement et arbitrairement un nombre de segments et les points de bouclages. Les utilisations typiques incluent les enveloppes portant sur l'amplitude, la fréquence et les fonctions d'indexages des durées.

Envelopes

MultisegmentEnvelope



Similaire à une enveloppe ADSR, sauf que vous pouvez spécifier graphiquement et arbitrairement un nombre de segments et les points de bouclages. Voir également le prototype MultiSlopeFunctionGenerator et GraphicalEnvelope. Utilisez le GraphicalEnvelope sauf dans les cas où vous avez besoin de niveaux et de points de discontinuité paramétrables. Les utilisations typiques incluent les enveloppes portant sur l'amplitude, la fréquence et les fonctions d'indexages des durées.

Envelopes

MultislopeFunctionGenerator



Ce prototype est similaire au MultiSegmentEnvelope, excepté que vous spécifiez les points temporels et les inclinaisons entre ces points (plutôt que des intervalles de temps et des niveaux), et vous ne pouvez pas boucler l'enveloppe. GraphicalEnvelope est plus facile à utiliser que le prototype MultiSlopeFunctionGenerator excepté dans des situations où vous avez besoin de paramétrer les durées et/ou les inclinaisons. La valeur restante de cette enveloppe est de 1. Chaque durée est redeclenchée et il génère une enveloppe une seule fois exactement.

Envelopes

VCA



Multiplie ses entrées l'une par l'autre. Cela est utile pour appliquer une enveloppe d'amplitude à un son.

Envelopes

WarpedTimeIndex



Peut être utilisé en tant qu'entrée vers n'importe quelle structure qui nécessite un index de temps (ex : GAOscillators, REResonator, SampleCloud, SpectrumInRAM). Génère un index de temps avec une pente variable de telle sorte qu'il puisse couvrir plus ou moins rapidement certaines zones du son ou plus ou moins lentement d'autres zones de ce son. Vous fournirez un ensemble de points temporels qui seront ajustés avec une avance ou un retard dans le temps pour atteindre les valeurs requises. La fonction d'index temporel ainsi générée fluctue de -1 jusqu'à 1 avec une pente variable.

KYMA PROTOTYPES

Event sources

AnalogSequencer



Génère des séquences de notes et transmet des valeurs de contrôles au paramètre EventValues de l'entrée Input. Il s'agit d'une séquence de numéros de notes MIDI, des durées, des rapports cycliques et des vitesses fournis à l'entrée Input, et le paramètre ExtraValues vous permet de fournir une séquence de valeurs pour chacun des contrôleurs continus EventValues présents dans les paramètres de l'entrée Input. La longueur totale d'un ensemble de séquences est la longueur de la plus longue des séquences de cet ensemble; toute séquence plus courte qu'une autre répètera ses valeurs finales de sorte à devenir aussi longue que l'autre. Par exemple, si vous voulez que toutes les valeurs d'une séquence soient égales à 0.5, vous avez simplement besoin d'entrer un chiffre pour un seul temps car elle sera ensuite reproduite aussi longtemps que nécessaire pour durer aussi longtemps que les autres séquences.

Si le Step (pas) est réglé sur une valeur constante 1, alors les durées et les rapports cycliques sont utilisés pour déterminer quand les événements !Keydown devraient être générés et combien de temps une touche du clavier devrait rester enfoncée. Si aucune autre unité de temps n'est utilisée et si la valeur Rate est à 1, les nombres affectés aux paramètres Duration de la séquence seront interprétés en secondes.

Rate est un diviseur de longueur pour chaque durée, donc si le Rate est plus grand que 1, les durées seront raccourcies et si Rate est plus petit que 1 alors elles seront plus grandes.

Utiliser Step pour avancer pas à pas dans la séquence en tenant compte bien entendu du Trigger. Par exemple vous pourriez utiliser !Keydown pour contrôler la cadence des pas depuis un séquenceur MIDI, ou vous pourriez utiliser l'expression 1 bpm: (!Speed * 1024) pour le contrôler avec un métronome interne, ou vous pourriez lui introduire un signal audio avec la commande paste qui aurait été injecté dans un module amplitude follower et un détecteur de seuil (threshold) pour déclencher chaque élément de la séquence en synchronisation avec le signal audio, ou vous pourriez utiliser !TimingClock pour avancer pas à pas en utilisant l'horloge MIDI depuis un séquenceur logiciel ou un synthétiseur externe.

Pour déclencher les événements en utilisant le champs Step, vous devrez régler toutes les durées Duration sur la même valeur; Cette valeur (ainsi que la valeur DutyCycle et Rate) déterminera la durée pour chaque note, et le déclencheur de pas (Step trigger) spécifiera le moment du déclenchement de la note. La valeur que vous adopterez pour les durées agira comme une sorte de "masque" sur la vitesse des step triggers. Les notes ne peuvent pas être déclenchées un plus grand nombre de fois que la valeur minimum spécifiée dans le champs Durations. Cela peut être utile si vous êtes en train de déclencher à partir de !TimingClock (24 déclenchements par beat) ou un signal audio contenant beaucoup de crêtes, parce qu'il forcera le séquenceur à ignorer les triggers qui auront lieu plus rapidement qu'à la vitesse souhaitée.

Event sources

SoundToGlobalController



Reçoit un nombre, un son , (colle avec la commande paste), ou une expression d'évènement en tant qu'entrée et génère à partir de la un EventValue (valeur d'évènement) , (même un simple évènement ou un flux de contrôleur continu) qui, pour tous les autres sons Kyma, apparaît de même nature que les EventValues provenant de la surface de contrôle virtuel ou provenant d'une source MIDI externe.

Filters

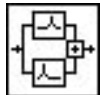
AveragingLowPassFilter



Filtre passe-bas qui fonctionne en prenant comme moyenne le flux des valeurs entrantes. La longueur de la moyenne courante est la période de la fréquence de coupure. La fréquence de coupure et ses harmoniques sont annulées par le filtre. Les fréquences aux alentours des fréquences annulées quant à elles sont atténuées.

Filters

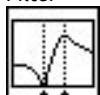
DualParallelTwoPoleFilter



Il s'agit de deux filtres du second ordre ayant des zéros fixes (au point complexe 1%0). La sortie de ce module est la somme des sorties des deux filtres. Ce module est utile si vous connaissez déjà la position complexe des pôles du filtre que vous voulez réaliser. Sinon, utilisez le prototype TwoFormantElement; cela vous donnera les mêmes résultats mais avec des paramètres plus intuitifs tels que la fréquence est la bande passante.

Filters

Filter



Filtre IIR du type spécifie, fréquence de coupure et ordre avec un gain ou une atténuation sur l'entrée et un attenuator pour doser la quantité de signal réinjecte (résonance).

KYMA PROTOTYPES

Filters	GraphicEQ 	Cela vous donne un contrôle indépendant sur les niveaux de 7 bandes larges d'un octave chacune et dont le centre va de 250Hz à 16000Hz. Vous pouvez l'utiliser pour atténuer ou accentuer des zones annexes du spectre. Lorsque tous les niveaux sont à 1, vous devriez obtenir le signal original entrant.
Filters	HarmonicResonator 	Il s'agit d'un filtre ayant des résonances à une fréquence spécifiée ainsi qu'aux harmoniques de cette fréquence.
Filters	HighShelvingFilter 	Accentue ou coupe les fréquences au-dessus de la fréquence de coupure spécifiée.
Filters	LowShelvingFilter 	Amplifie ou atténue le spectre en dessous de la fréquence de coupure spécifiée.
Filters	PresenceFilter 	Agit comme un filtre passe-bande ou un filtre réjecteur de bande. Spécifier une fréquence centrale, une bande de fréquence et indiquer la valeur de l'amplification ou d'atténuation en dB (les positives sont des amplifications et les négatives sont des atténuations)
Filters	REResonator 	C'est un filtre variant dans le temps dont les coefficients proviennent de l'analyse d'un enregistrement numérique (un échantillon) en utilisant l'outil d'analyse RE. L'analyse RE (resonator/exciter) établit que le son est le produit de l'excitation d'un résonateur par un signal secondaire. Ce module est en fait le résonateur et son entrée c'est l'exciteur. Les résultats les plus intrigants se produisent lorsque le signal analyse provient d'une source qui comprends elle-même également des variations de résonance (par exemple la voix humaine, les instruments comme le didgeridoo, le tabla). En ce qui concerne les analyses d'instruments ou d'autres sources dont l'onde varie peu au cours du temps, le REResonator semblera stable un peu comme un filtre statique.
Filters	ScaleVocoder 	Vocodeur dont les fréquences centrales sont accordées par rapport au ton de base et à l'échelle.
Filters	TwoFormantElement 	TwoFormantElement fonctionne un peu comme le DualParallelTwoPoleFilter; cependant, plutôt que de spécifier un filtre en termes de position des pôles, vous spécifiez la fréquence centrale et la bande passante des deux formants.
Filters	Vocoder 	Le Vocoder imprime le caractère spectral du son SideChain sur le son d'entrée. La banque d'analyse est utilisée pour mesurer la quantité d'énergie dans chacune des bandes de fréquence du son SideChain. La banque de resynthèse est utilisée pour filtrer le son entrant. Vient ensuite un suiveur d'amplitude à la sortie de chacun des filtres dans la banque d'analyse. Les enveloppes d'amplitude résultantes sont ensuite appliquées aux filtres respectifs de la banque de resynthèse. De cette façon, le signal SideChain contrôle les enveloppes d'amplitude des filtres de resynthèse.

KYMA PROTOTYPES

Filters	VocoderChannelBank	Dans a plupart des cas, vous utiliserez le Vocoder plutôt que le VocoderChannelBank car le Vocoder est un prototype de niveau plus élevé possédant un degré plus important de paramètres et de contrôles. Utilisez le VocoderChannelBank dans des cas qui nécessitent un contrôle indépendant sur la fréquence centrale, l'amplitude, et la bande passante de chacun des filtres de la banque d'analyse ainsi que de la banque de resynthèse.
---		
Frequency & Time Scaling	Monotonizer	Le VocoderChannelBank travaille en routant l'entrée sidechain au travers d'une batterie de filtres passe-bande (filtres d'analyse), puis en extrayant des enveloppes d'amplitude à la sortie de chacun de ces filtres, en appliquant ces enveloppes à une seconde batterie de filtres (les filtres de synthèse) et en routant l'entrée vers cette banque de filtres de synthèse.
		Pour voir un exemple sur la façon avec laquelle le VocoderChannelBank peut être utilisé, disposez un Vocoder dans la fenêtre du fichier son (Sound file window) et activez la fonction expand ; Il se décomposera en une cascade de modules VocoderChannelBanks.
Frequency & Time Scaling	SimplePitchShifter	Annule les variations de tonalité du signal d'entrée et utilise la fréquence spécifiée en remplacement.
		
Frequency & Time Scaling	SingleSideBandRM	Transpose le ton du signal d'entrée (vers le haut ou vers le bas) par demi-tons.
		Effectue un re-étalonnage non harmonique du signal d'entrée : Prends l'entrée et effectue un déphasage de 90 degrés entre le signal gauche et le signal droit à la fréquence spécifiée par le paramètre Frequency. QuadratureOscillator Multiplie ensuite par un sinus sur le signal gauche et un cosinus sur le signal droit. La modulation en anneau résultante donne une somme et une différence des fréquences d'entrée mais, parce qu'ils sont décalés de 90 degrés, la fréquence a laquelle a lieu le traitement est pratiquement absente, ne laissant passer qu'une modulation a bande simple mais sans "la porteuse". Utilisez la fonction Expand pour comprendre le fonctionnement.
Generators	Constant	Constant fait passer en sortie la valeur donne par le paramètre Value. Si Value est un nombre, la sortie de Constant prend cette valeur. Si vous copiez-collez (avec la fonction Paste) un évènement dans le champs du paramètre Value, la sortie de Constant est égale à cet évènement. Cela est utile pour le traitement d'évènements (Event Values) comme s'il s'agissait de sons. Par exemple, si vous étiez sur le point de copier-coller le message de contrôle !cc07 dans le champs du paramètre Value, vous pourriez ensuite câbler le module Constant vers un Delay, un Waveshaper, multiplier le résultat par un oscillateur sinusoïdal ou bien réaliser toute autre opération.
		
Generators	GenericSource	Ce module peut représenter une entrée live, un échantillon lu à partir du disque dur ou un échantillon lu depuis la RAM. Si l'option Ask est cochée, vous pouvez choisir entre trois types de sources chaque fois que vous compilez le module.
		
Generators	PulseTrain	Si l'option VariableDutyCycle n'est pas cochée, alors PulseTrain prends la valeur 1 au premier échantillon de chaque période et 0 pendant le reste de la dite période. Si l'option VariableDutyCycle est cochée, alors DutyCycle contrôle de quelle manière quelle fraction de chaque période génère un 1 et quelle autre fraction restante de la dite période génère un 0.
		

KYMA PROTOTYPES

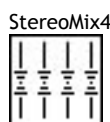
Memory	ForcedProcessorAssignment 	Force le traitement d'un module par l'une des cartes d'extensions spécifiées. Dans la plupart des cas, il est plus sur de laisser kyma assurer automatiquement le déroulement du traitement parmi les différents processeurs. Cependant, ce module vous permet de contrefaire les allocations par défaut et de forcer le calcul d'un module en particulier par l'un des processeurs. Les raisons d'un tel choix pourraient être : le besoin d'enregistrer un échantillon dans une table d'onde d'une carte d'extension particulière (ou un ensemble de cartes) et être en mesure de relire l'échantillon plus tard à partir de cette même carte, ou bien essayer de réorienter le traitement d'un module a la main si l'allocation dirigée par kyma ne peut pas s'effectuer parfaitement en temps réel (faire votre propre gestion des processeurs est très complexe et demande du temps, ce qui n'est pas forcément requis ! Voir le tutorial "What is Real Time Really" pour connaître d'autres façons de réduire la complexité du traitement d'un module qui ne peut pas être exécuté en temps réel).
Memory	MemoryWriter 	Lorsque trigger devient positif, ce module enregistre l'entrée (Input) dans le mémoire de table d'onde du processeur pour une durée spécifiée par le paramètre CaptureDuration. Tout module qui relit des tables d'ondes peut être alors utilisé dans le but de rejouer cet enregistrement (par exemple les modules FunctionGenerator, Sample et bien d'autres).
Memory	TriggeredTableRead 	Tant que trigger est plus grand que zéro, le module TriggeredTableRead lit les échantillons de la Wavetable; si le trigger est moins grand que ou égal à zéro, le dernier échantillon sera lu. Gate réinitialise le pointeur au début de la Wavetable.
Midi In	MIDIMapper 	MIDIMapper considère son entrée comme étant une voie MIDI ayant une polyphonie à spécifier. Le canal MIDI de cette voie est également à spécifier. La voie MIDI est alors jouable depuis un appareil MIDI externe en temps réel ou à partir d'un fichier MIDI file. Left et Right sont des atténuateurs des canaux gauche et droit de la sortie audio de ce module. Une zone de jeu fournie dans le champs de zone aura priorité sur la zone de jeu MIDI globale pour toute valeur d'événements (Event Values) présente sur son entrée. Si vous n'avez pas besoin de redéfinir une zone de jeu, utilisez le module MIDIVoice a la place.
Midi In	MIDIVoice 	Considère son entrée comme étant une voie MIDI ayant une polyphonie à spécifier. Le canal MIDI de cette voie est également à spécifier. La voie MIDI est alors jouable depuis un appareil MIDI externe en temps réel ou à partir d'un fichier MIDI file. Left et Right sont des atténuateurs des canaux gauche et droit de la sortie audio de ce module
Midi In	Preset 	Cliquer sur le bouton "Set to Current Event Values " pour sauvegarder les réglages, faders et les boutons de la surface de contrôle virtuel (et/ou les réglages courant de tous les contrôleurs MIDI). Les valeurs courantes sont écrites dans le champs EventValues comme un "pense bête" (mais elles ne sont pas éditables). Chaque fois que vous jouer la sonorité, toute valeur d'événement (EventValues) présente sur ses entrées sera initialisée sur les valeurs qui ont été précédemment sauvegardées; vous pouvez ensuite les changer en utilisant la surface de contrôle virtuelle ou via MIDI.
Midi Out	MIDIFileEcho 	Ce module interprète tous les événements MIDI présents dans le fichier désigné sur un groupe de canaux et les réinterprète ensuite sur la sortie Midi. Il ne fournit pas d'informations Midi dans Kyma même mais dirige la lecture du fichier Midi vers la sortie Midi physique.
Midi Out	MIDIOutputController 	Ce module envoie son paramètre Value vers la sortie MIDI sur le canal spécifié et fait office de contrôleur continu.
Midi Out	MIDIOutputEvent 	Lorsque Gate devient positive, a message note-on est envoyé avec les valeurs courantes de Fréquence et d'Amplitude sous forme de numéro de note et valeur de vélocité sur le canal Midi donne. Lorsque Gate retourne à 0, un message de not-off est envoyé.

KYMA PROTOTYPES

Midi Out	MIDIOutputEventInBytes 	Ce module envoie une séquence non interprétée d'octets vers la sortie Midi. Vous pouvez utiliser ce module pour envoyer des évènements Midi arbitraires.
Mixers & Attenuators	Attenuator 	Le module Attenuator multiplie ses canaux gauche et droit par les valeurs Left et Right. Il peut être utilisé pour positionner le signal d'entrée dans le champs stéréo des Haut-parleurs ; Par exemple, si l'entrée est multipliée par 1.0 à droite et si l'entrée est multipliée par 0 à gauche, il semblera que la source de l'entrée est localisée à droite de l'auditeur.
Mixers & Attenuators	CenteringMixer 	Aligne les "milieux" des durées des entrées avec les "milieux" des durées des entrées les plus longues et transmet en sortie le mixage de toutes ces entrées ainsi reconfigurées. Le point milieu "Midpoint" est défini en tant que la moitié de la durée de l'entrée la plus longue. On peut faire l'analogie avec le centrage des lignes d'un texte au milieu de la page d'un document Word.
Mixers & Attenuators	ChannelCrossover 	ChannelCrossover permet de sectionner n'importe quelle portion du signal gauche et de le reporter dans le signal droit et vice versa.
Mixers & Attenuators	ChannelJoin 	Ce module place le canal gauche de Left sur la sortie gauche et place le canal droit de Right sur la sortie droite.
Mixers & Attenuators	Channeller 	Si l'option LeftChannel est cochée, le canal gauche de l'entrée est envoyé sur les deux canaux de sortie. Si l'option RightChannel est cochée, le canal droit de l'entrée est envoyée sur les deux canaux de sortie. Si les deux options sont cochées, le module est "transparent". Si rien n'est coché, rien ne sera joué en sortie.
Mixers & Attenuators	Crossfade 	Effectue un crossfade entre les deux entrées tandis qu'il est possible optionnellement de faire un pan et d'en atténuer le résultat final.
Mixers & Attenuators	Matrix4 	Mixeur Matriciel a 4 entrées, 4 sorties. Les 4 entrées sont routées et mélangées vers les 4 canaux de sortie du processeur de signal. Ce module fonctionne correctement s'il est le module situé le plus à droite de la structure.
Mixers & Attenuators	Matrix8 	Mixeur Matriciel a 8 entrées, 8 sorties. Les 8 entrées sont routées et mélangées vers les 8 canaux de sortie du processeur de signal. Ce module fonctionne correctement s'il est le module situé le plus à droite de la structure.
Mixers & Attenuators	Pan 	Positionne l'entrée entre les haut- parleurs gauche et droit et atténue éventuellement la sortie générale.
Mixers & Attenuators	StereoMix2 	Additionne les signaux de sortie des deux modules connectés sur In1 et In2, chacun ayant une valeur de panoramique et d'atténuation réglable. La sortie globale peut elle aussi être atténuée et avoir son propre réglage de panoramique.

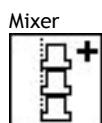
KYMA PROTOTYPES

Mixers & Attenuators



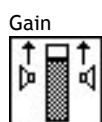
Additionne In1, In2, In3, et In4, chacun ayant son propre contrôle de panoramique et d'atténuation. Scale et Pan contrôlent l'atténuation et la position stéréo du mixage final.

Mixers & Attenuators



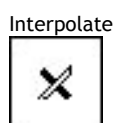
Additionne toutes ses entrées ensemble. Mixe les sorties de tous les modules indiqués dans les champs de paramètre prévus à cet effet.

Mixers and Attenuators



Le module Gain peut être utilisé pour booster l'amplitude au-dessus du maximum 1.0. Cela peut être utile lorsque vous voulez multiplier par des nombres supérieurs à 1 ou lorsque vous avez une amplitude de signal faible. Si vous avez besoin d'un contrôle de gain variable, vous pouvez utiliser le module Attenuator avec une atténuation variable (en utilisant les échelles Left et Right) appliquée au son avant son entrée dans le module Gain.

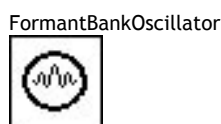
Mixers and Attenuators



Réalise une combinaison linéaire des deux entrées. Le canal gauche, entre dite Input0, est multiplié par 1-leftInterp, le canal gauche de l'entrée Input1 est multiplié par leftInterp, et les deux sont additionnées ensemble (idem pour les canaux droits et rightInterp).

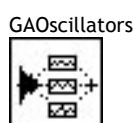
Ce module est utile pour l'interpolation de fonctions de contrôle et d'enveloppes (par exemple l'interpolation entre deux sets d'enveloppes d'analyse et en envoyant la sortie vers l' OscillatorBank ce qui donne un morphing spectral).

Oscillators



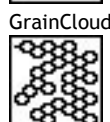
Synthétise un train de pulsations filtrées dont le filtre est basé sur la forme du FormantImpulse et sur les fréquences, les amplitudes, et les bande-passantes du formant que vous fournissez via le paramètre Spectrum (qui est généralement un module SyntheticSpectrumFromArray).

Oscillators



Synthèse additive utilisant des oscillateurs avec des formes d'onde complexes (plutôt que des sinusoïdes). Chaque oscillateur a sa propre enveloppe d'amplitude et tous les oscillateurs partagent la même enveloppe de modulation de fréquence.

Oscillators



Génère un nuage de grains sonores de courte durée, chacun ayant une Waveform spécifique et chacun ayant une enveloppe d'amplitude dont la forme est donnée par le paramètre GrainEnv. La densité de grains simultanés dans le dit nuage est contrôlée par Density, et le nombre maximal de grains est donné par le paramètre MaxGrains.

Amplitude contrôle l'enveloppe d'amplitude globale du nuage entier (chaque amplitude de chaque grain est contrôlée par GrainEnv). Pareillement, Duration est la durée du nuage entier et non pas la durée individuelle de chaque grain.

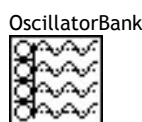
Vous pouvez contrôler Frequency, ainsi que le panoramique et la durée de chaque grain. Vous pouvez aussi spécifier la quantité de tremblement (jitter) aléatoire qui sera additionnée à chacun de ces paramètres (ce qui apporte au nuage un son plus disparate)

Oscillators



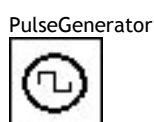
Ce module génère un bruit blanc ou rose, ce qui correspond à des nombres pseudo-aléatoires. Choisissez différents états initiaux pour générer différentes suites de nombres aléatoires. Pour obtenir des nombres aléatoires à basse vitesse, utilisez quelque chose comme "1ms random" dans le champ de paramètre.

Oscillators



Génère la somme de plusieurs oscillateurs ayant la forme d'onde spécifiée, chacun ayant ses propres enveloppes d'amplitude et de fréquence.

Oscillators



Génère un signal rectangulaire avec un rapport cyclique DutyCycle spécifié. L'onde carrée a toujours un décalage nul quel que soit le réglage de la largeur du créneau; Cela signifie que le minimum et le maximum de l'onde changeront en fonction du réglage de rapport cyclique.

KYMA PROTOTYPES

Oscillators

QuadratureOscillator



Multiplie le canal gauche d'Envelope par un oscillateur sinusoïdal et le canal droit d'Envelope par un oscillateur cosinusoidal. La sortie est la somme des canaux droits et gauches modules en anneau. Si Envelope presente les mêmes signaux gauche et droit mais dephase de 90 degrés, la bande latérale basse sera annulée, ne laissant donc passe que la bande latérale supérieure (la somme des fréquences de Envelope et de QuadratureOscillator).

Activez la fonction Expand sur le prototype SingleSideBandRM pour avoir une idée sur la manière d'utiliser ce module en tant que transposition de fréquence non harmonique.

Oscillators

SumOfSines



Resynthetise des sons à partir d'analyses spectrales stockées dans Analysis0 et Analysis1. le Paramètre dbMorph exécute une interpolation entre les amplitudes d'Analysis0 et d'Analysis1, et le paramètre pchMorph exécute une interpolation entre les hauteurs d'Analysis0 et d'Analysis1.

Sampling

AudioInput



AudioInput représente les entres analogiques ou numériques situées à l'arrière du Capybara. Si l'option Digital Input et sélectionnée dans la fenêtre d'état DSP, alors ce dernier représente une entrée numérique. Si des micros pré-amplifiés sont connectes aux entrées, alors ce module représente les entrées de ces micros. Vous pouvez aussi connecter une source audio ou toute autre entrée ligne (CD, DAT) a l'entrée.

Les cases à cocher de canaux individuels contrôlent quelles sont les entrées audio qui seront utilisées. Si un seul canal est coché, AudioInput le dupliquera à deux fois sur ses deux canaux de sortie.

Sampling

KeyMappedMultisample



Une manière rapide de disposer d'une large banque de samples et de les assigner aux touches du Clavier MIDI. Cela peut être utile pour constituer des multi-échantillons d'instrument ou pour déclencher une grande quantité d'effets en temps réel (depuis le séquenceur ou depuis le clavier MIDI).

Pour spécifier les échantillons qui appartiennent à la même banque, placez-les tous dans le même répertoire. Kyma regardera au sommet de votre répertoire, donc tout autre répertoire inclus dans ce même répertoire sera ignore. L'ordre des échantillons au sein de ce répertoire sera alphabétique.

Les fichiers contenus par ce répertoire doivent être tous mono ou stéréo, des mélanges de ces deux catégories ne sont pas garantis d'être interpretees correctement.

Sampling

Multisample



Ce module fournit un moyen rapide de sélectionner un certain nombre samples. Les fichiers de samples sont listes dans le champs des Samples, et le champs d'Index est utilise pour déminer quel fichier de samples doit être joue des lors que Gate prend une valeur positive.

Sampling

Sample



Interprète le Sample à partir de la mémoire de table d'onde du processeur de signal. Si une boucle est stockée dans le fichier de sample ou si le paramètre SetLoop est coché, le sample sera joue une fois jusqu'à la fin de la boucle LoopEnd; ensuite il sera rebouclé à partir du LoopStart et continuera la boucle aussi longtemps que Gate prendra une valeur positive; lorsque Gate retourne à zéro, le sample sera jouera jusqu'à sa fin.

Sampling

SampleAccelerator



Prends un sample et le raccourci la boucle a chaque passage (en raccourcissant de plus en plus le début ou la fin de la boucle).

Sampling

SampleAndHold



Le module SampleAndHold retient la valeur courante de l'entrée pour une durée HoldTime à spécifier. Durant la retenue de cette valeur, le module ignore toute variation du signal d'entrée. Lorsque la durée HoldTime est expirée, le module observe à nouveau la valeur du signal d'entrée et retient cette nouvelle valeur encore pour un temps HoldTime et ainsi de suite.

Cela décroît effectivement la vitesse de lecture de l'échantillon présente en entrée.

Essayez de copier-coller ce module dans le champs de fréquence d'un autre module et multipliez le par la gamme de fréquences désirée additionnez un offset sur la fréquence, par exemple :

$4c + ([SampleAndHold] * 12nn)$ ou $[SampleAndHold]$ est le module copie-colle dans le champs de fréquence de l'autre module.

KYMA PROTOTYPES

Sampling	SampleCloud 	Génère un nuage de grains de courte durée, chaque grain utilise GrainEnv en tant qu'enveloppe d'amplitude sur un court segment de son pris à partir d'un Sample spécifique à un point donné par TimeIndex. La densité de grains joués simultanément au sein du nuage est contrôlée par Density, avec le nombre maximum de grains donné par MaxGrains. Amplitude contrôlée par une enveloppe d'amplitude du nuage entier (l'amplitude de chaque grain est contrôlée par GrainEnv). Pareillement, Duration est la durée du nuage entier, pas la durée de chaque grain sonore. Vous pouvez contrôler la fréquence Frequency, la position stéréo, le point de lecture dans l'échantillon, et la durée de chaque grain tout en spécifiant également quel taux de tremblement aléatoire pourrait être ajouté à chacun de ces paramètres (ce qui donnera au nuage un son plus centré ou plus disparate selon les aléas conférés à chacun des paramètres).
Sampling	TriggeredSampleAndHold 	Lorsque ce module est déclenché, il lit une valeur du signal d'entrée et la retient jusqu'à ce qu'il soit déclenché de nouveau. Comportement similaire au module SampleAndHold sauf que l'échantillonnage a lieu seulement sur des déclenchements et non pas périodiquement.
Scripts	TextFileInterpreter 	Reads and interprets a line at a time from a text file. This Sound can be used to interpret scores prepared for Music N languages (such as csound) and map the parameters to the variable parameters of Kyma Sounds, essentially treating Kyma Sounds as "instruments". It can also be used more generally to read data from text files and map them to the parameters of Kyma Sounds. This is like the Script in that it constructs a new Sound algorithmically and then plays it; it does NOT set up fixed "patches" and then update the parameters by reading them out of the text file as the Sound is playing. Lit et interprète une ligne à la fois à partir d'un fichier texte. Ce module peut être utilisé pour interpréter des pièces musicales préparées avec des langages musicaux (tels que csound) et répartir les paramètres vers les variables des sons Kyma, en considérant les sons Kyma en tant qu'instruments. Il est plus généralement utilisé pour lire des données à partir de fichiers texte et les répartir vers les paramètres des prototypes Kyma. C'est un peu comme le module Script qui permet d'élaborer un nouveau son algorithmiquement et qui le joue ensuite; Cependant il n'établit pas des patches définitifs mais il met à jour les paramètres en les lisant à partir du fichier texte au fur et à mesure de l'interprétation du son.
Sources	TwoFormantVoiceElement 	Un signal d'excitation similaire à une pulsation glottale (avec un taux de vibrato aléatoire) est utilisé en tant qu'entrée vers une paire de filtres parallèles du second ordre qui simulent les deux formants de la cavité vocale.
Spatializing	Output4 	Ce module dirige les 4 signaux d'entrée vers les 4 canaux de sortie du processeur de signal. Ce module fonctionne correctement dans une structure en tant que le module le plus à droite de la chaîne de traitement.
Spatializing	Output8 	Ce module dirige les 8 signaux d'entrée vers les 8 canaux de sortie du processeur de signal. Ce module fonctionne correctement dans une structure en tant que le module le plus à droite de la chaîne de traitement.
Spectral Modifiers	SpectrumFrequencyScale 	Prends une source spectrale (qui doit être harmonique) en tant qu'entrée et ré-étalonne les enveloppes de fréquence sans changer les enveloppes d'amplitude. Cela permet de transposer le ton du son re-synthétise tout en conservant les formants tels quels. Le module SpectrumFrequencyScale devrait alimenter un OscillatorBank de façon à entendre le nouveau spectre obtenu.
Spectral Modifiers	SpectrumLogToLinear 	Un spectre peut se présenter sous deux aspects : fréquences linéaires ou fréquences log. Ce module convertit un spectre de fréquences log en un spectre de fréquences linéaires. En général, un spectre provenant d'un fichier de spectre a des fréquences log et un spectre généré en temps réel a des fréquences linéaires.

KYMA PROTOTYPES

Spectral Modifiers

SpectrumModifier



Il faut d'abord veiller à ce que les modules précédant Le SpectrumModifier fassent partie de la catégorie des sources spectrales (Spectral Sources).Le SpectrumModifier en modifiera le spectre. Pour re-synthétiser le spectre modifié, reliez la sortie de ce module vers l' OscillatorBank.

De façon à modifier la source spectrale, le SpectrumModifier sélectionne ou exclue des pistes du spectre selon certains critères établis, ensuite il ré-étalonne éventuellement et décale la valeur de fréquence et/ou d'amplitude des pistes sélectionnées.

Décidez d'une sélection ou d'une réjection de pistes qui correspondent aux critères choisis.

Décidez ensuite si les pistes rejetées auront une amplitude nulle ou si elles passeront sans altération les modifications d'échelles et de décalage. Les valeurs d'hystérésis de fréquence et d'amplitude permettent d'éviter que des pistes proches de la zone sélectionnée basculent par intermittence de façon gênante. Probability est un paramètre (allant de 0 à 1) portant sur la possibilité qu'une piste soit sélectionnée (ou rejetée) à chaque cadrant.

Enfin, vous pouvez choisir de ré-étalonner ou de décaler la fréquence ou l'amplitude (ou les deux) de chaque cadrant de chacune des pistes sélectionnées.

Spectral Sources

SpectralShape



Le module SpectralShape règle les fréquences et les amplitudes des oscillateurs du module OscillatorBank selon les paramètres Spacing et spectralenvelope. Cette sorte de module n'a de sens que lorsqu'il est suivi par l' OscillatorBank. Les fréquences sont sur le canal gauche et les amplitudes sont sur le canal droit.

Spectral Sources

SpectrumInRAM



This Sound is used only as the Spectrum input to an OscillatorBank.

It reads an analysis file that contains a series of spectra indexed by TimeIndex. It outputs a spectrum as a sequence of (amplitude,frequency) pairs on every sample tick for nbrPartials samples. After nbrPartials samples, it starts over again from the fundamental and outputs the entire spectrum again.

Il convient de faire suivre ce module par l' OscillatorBank.

Il lit et analyse des fichiers qui contiennent une série de spectres indexés par le paramètre TimeIndex. Il transmet un spectre sous forme d'une liste de binômes amplitude;fréquence à chaque coup d'échantillonnage en nbrPartials étapes cadences à la vitesse d'échantillonnage

Spectral Sources

SpectrumOnDisk



Ce module peut être utilisé à la place du module SpectrumInRAM , toujours penser à placer en aval l' OscillatorBank. La différence est qu'il lit le fichier d'analyse directement depuis le disque dur plutôt que de lire le fichier analyse depuis la RAM. Cela est utile pour les analyses SOS qui sont trop longues pour être logées dans la RAM.

Spectral Sources

SyntheticSpectrumFromArray



Génère un spectre synthétique à partir de deux tableaux : un tableau de valeurs d'amplitudes pour chaque piste du spectre et un tableau de valeurs de fréquences pour chaque piste du spectre. (Et aussi, si l'option SendBandwidths est cochée, un tableau de bande passantes correspondant pour chacune des pistes). Le module SyntheticSpectrumFromArray devra être relié à l'entrée de l'OscillatorBank, du FormantBankOscillator ou bien du VocoderChannelBank de manière à synthétiser les partiels, les formants ou les banque des filtres de vocoder. Le module SyntheticSpectrumFromArray génère un ensemble d'enveloppe pour contrôler les paramètres de l'OscillatorBank, du FormantBankOscillator, ou du VocoderChannelBank.

KYMA PROTOTYPES

Spectral Sources	SyntheticSpectrum-FromSounds	Génère un spectre synthétique dont les amplitudes, fréquences et optionnellement bande-passantes sont contrôlés par deux entrées. Une entrée fournit les amplitudes et l'autre fournit les fréquences (alternant optionnellement avec les bandes passantes). Vous pouvez identifier chaque cycle une définition complète du spectre. Si l'entrée varie de cycle en cycle, alors le contenu du spectre de sortie évoluera. Le module SyntheticSpectrumFromSounds (comme les autres modules de la catégorie des sources spectrales) fournit des enveloppes spectrales dans le format suivant : Canal gauche : Amp1 Amp2 ... AmpN Canal droit : Freq1 Freq2 ... FreqN Pour chaque cadran, Amp1 est l'amplitude du premier partiel (et Freq1 est la fréquence ou hauteur du premier partiel), Amp2 est l'amplitude du second partiel (correspondant avec Freq1) et AmpN est l'amplitude du partiel le plus haut (spécifie dans NbrPartials). La séquence entière est ré-itérée pour le cadran du spectre suivant. A cause de cette répétition, la sortie du module SyntheticSpectrumFromSound a une période égale au nombre de partiels contenu dans chaque cadran.
Time & Duration	SetDuration	Etablit la durée et le point de départ de son entrée. (Cela équivaut à placer un son sur la timeline en changeant sa durée et son départ graphiquement). Sans le réglage SetDuration , le programme d'entrée continuerait indéfiniment; Avec SetDuration vous pouvez marquer un arrêt au bout d'un certain temps.
Time/Frequency Scaling	TimeFrequencyScale	Transpose et étire temporellement un enregistrement sur disque ou un échantillon stocké dans la mémoire de table d'onde.
Time/Frequency Scaling	FrequencyScale	Ré-étalonne la fréquence de l'entrée avec la valeur entrée dans le champs FrequencyScale. Ceci est réalisé par une granulation de l'entrée et par chevauchement des grains pour monter en fréquence, ou par écartement des grains pour descendre en fréquence.
Tracking Live Input	AmplitudeFollower	Suit l'amplitude de l'entrée en traquant une moyenne des valeurs absolues des échantillons. Cela est similaire au RMS mais nécessite moins de calcul. TimeConstant contrôle sur quelle portion de temps est calculée la moyenne. Ainsi, plus TimeConstant est grand, plus la sortie est adoucie (mais plus TimeConstant est petit et plus le module peut répondre à des transitoires rapides).
Tracking Live Input	SoundToGlobalController	Accepte un nombre, un son copié-collé, une expression d'évènement Event de façon le traduire en une valeur évènement EventValue (même un simple évènement ou une grandeur de contrôle continue) qui, pour tous les autres sons Kyma, paraît être de même nature que les EventValues provenant de la surface de contrôle virtuelle ou provenant d'une source Midi externe.
Variable & Annotation	Variable	Variable est un conteneur qui représente un son/module/structure. En rédigeant un Script, vous pouvez assigner à Variable une valeur ou un son en tapant le nom de Variable suivie par une colonne, un espace puis le nom du son que vous voulez assigner à la variable.
Variables & annotations	Annotation	Une Annotation contient un texte de commentaire sur une entrée. Cela n'affecte pas la chaîne de traitement. Le Texte d'une Annotation apparaît dans la surface de contrôle virtuelle a chaque fois que la structure est exécutée.
Variables & Annotations	SoundCollectionVariable	Ceci représente un ensemble de modules. Il peut apparaître dans n'importe quel champ de paramètre qui concerne plus d'un module. Celui ci est typiquement utilisé pour créer des Sound Classes qui ont un nombre arbitraire d'entrées.

KYMA PROTOTYPES

Waveshaping

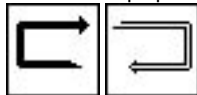
IteratedWaveshaper



Similaire au waveshaping sauf que la sortie est réinjectée dans le waveshaper pour un nombre spécifié d'itérations avant que le résultat ne soit finalement donné.

Xtra

FeedbackLoopOutput et FeedbackLoopInput



FeedbackLoopInput et FeedbackLoopOutput doivent toujours être utilisés par paire en partageant le même nom de Connection, le même départ et la même durée. FeedbackLoopInput écrit dans la ligne de Delay spécifié dans Connection, et FeedbackLoopOutput relit à partir de cette même ligne de delay.

Cela diffère des autres méthodes pour réaliser un feedback tel que cela permet à l'organisateur de Kyma de repartir l'entrée et la sortie de la ligne de délai parmi différentes cartes d'extension, libérant ainsi plus de puissance de traitement pour les autres modules qui sont inclus dans la boucle. Pour des cas simple de feedback, utilisez le module DelayWithFeedback ou écrivez simplement dans le MemoryWriter et relisez le avec le TimeOffset Sample, l'Oscillator ou TriggeredTableRead.

Xtra

Oscillator



La table d'onde Wavetable est traitée comme étant le cycle unitaire d'une fonction périodique. Il existe des options pour la modulation et l'interpolation. En général, plus d'options peuvent être sélectionnées et plus de paramètres varieront au cours du temps, mais plus compliqué sera le traitement de l'Oscillator et plus réduit sera le nombre d'Oscillators pouvant être traités en temps réel.

Xtra

SOS Oscillators



Génère la somme de quelques oscillateurs avec la forme d'onde spécifiée, chacun ayant sa propre enveloppe de fréquence et sa propre enveloppe d'amplitude.

Sylvain KEPLER.