

VSTiつくるよ

2012/6/10

Twitter: @y3eadgbe

VSTiつくるよゼミ(<http://www38.atwiki.jp/vstisemi/>)

ソフトウェアシンセつくりたい

- そもそもシンセってなにによ
 - シンセサイザー入門(<http://www.groovecube.com/ja/tutorial/>)
 - いわゆる減算式シンセサイザーを作りたい
- どうやって音を出すか
 - スタンドアロン（OpenALなりAPI直接いじるなり）
 - DAWとの連携が難しい（Rewire対応?）
 - DAWのプラグイン形式（VSTなど）
 - 信号処理部分に注力できる
 - DAW等のHostとの連携が前提
 - 作ったシンセを使える!!

VSTってなによ

- Steinberg's Virtual Studio Technology
 - DAW等のホストで使えるプラグインの規格
 - Steinberg社の独自規格だがかなり普及している
 - 主にWindows用
 - SDKが公開されている（要developer登録）
(<http://www.steinberg.net/en/company/developer.html>)
 - 言語はC++（Delphi用のSDKもあるらしい）

VSTつくるよ

- 用意するもの
 - Visual C++
 - 動作確認用のVST Host
- SDKをダウンロードする
 - VST2.4と3.0があるが2.4の方が対応Hostも多く無難
- SDKのマニュアルを見る
 - 英語わからん
 - それを差し引いても不親切

VSTつくるよ

- サンプルを見る

- 付属のサンプル

- コメントにもあるようにシンセとしてはまずい (wiki参照)

- かの有名なSynth1の作者Daichiさんによる神サンプル

- Synth1(<http://www.geocities.jp/daichi1969/softsynth/>)
 - サンプル(<http://www.geocities.jp/daichi1969/vstisample/>)
 - 親切かつ大きさも手頃でわかりやすい
 - Webページにシンセ制作のノウハウがいろいろ載っている
 - これ見てもらえば他に書くことないのでは……

VSTつくるよ

- 結局何を作ればよいのか
 - AudioEffectXクラスを継承したクラスを作り
各種メソッドを実装
 - Hostからの問い合わせに答える系メソッド
 - getEffectName() など
 - MIDIイベントの処理
 - processEvents()
 - 信号処理（シンセの本体）
 - processReplacing()
 - ここを簡単に解説

信号を処理するよ

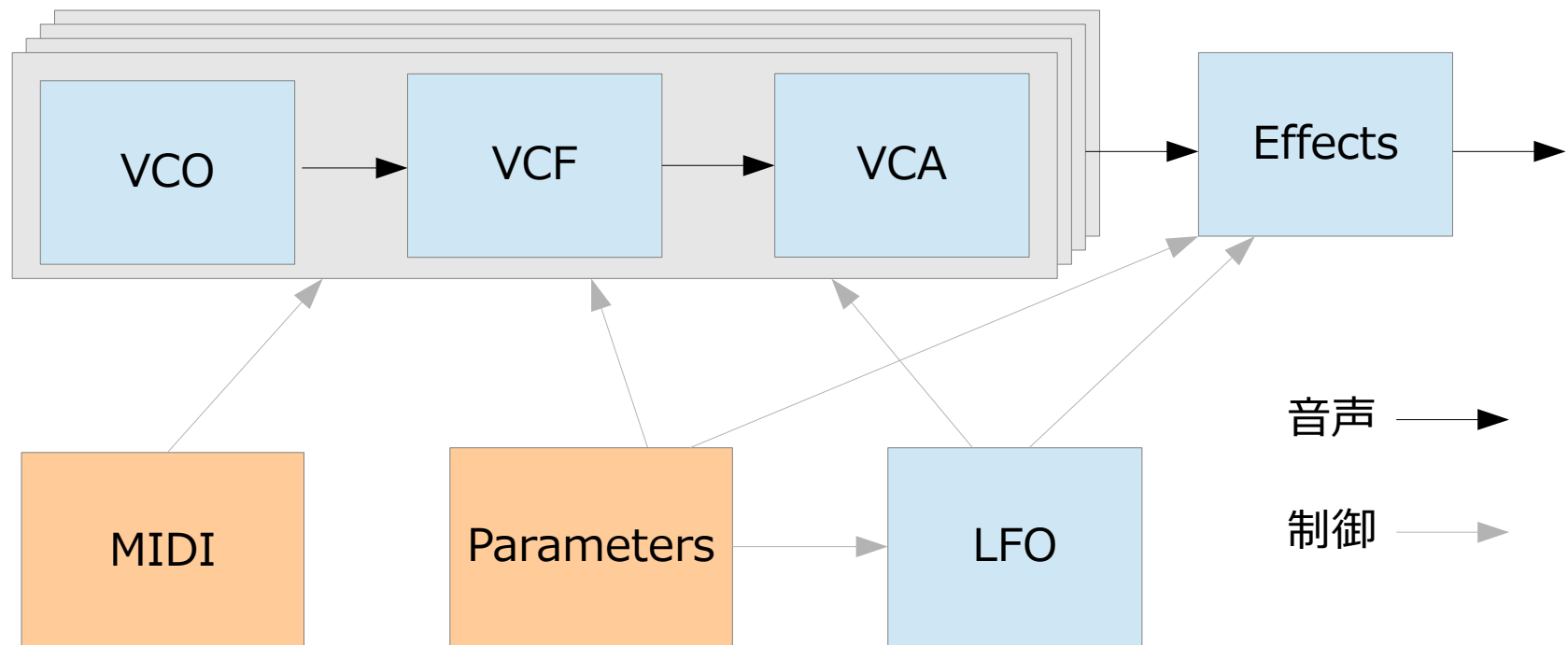
- 減算式シンセサイザーの構成要素
 - VCO(Oscillator)
 - 基本波形の生成
 - VCF(Filter)
 - 音色の調節
 - VCA(Amplifier)
 - 音量の調節
 - その他
 - LFO, エフェクト

信号を処理するよ

- 減算式シンセサイザーの構成要素

- 詳しくはシンセサイザー入門などで

<http://www.groovecube.com/ja/tutorial/architecture.html>



波形を作るよ

- 減算式シンセサイザーの構成要素
 - VCO(Oscillator)
 - 基本波形の生成
 - VCF(Filter)
 - 音色の調節
 - VCA(Amplifier)
 - 音量の調節
 - その他
 - LFO, エフェクト

波形を作るよ

- Oscillator

- 正弦波、三角波、のこぎり波、矩形波などの基本波形を
なるべくノイズなく生成する
- 生成方法(<http://www.geocities.jp/daichi1969/synthprog/>)
 - WaveTable方式
 - 計算が軽くて音質もそこそこ
 - この方式について説明
 - BLIT(Bandlimited Impulse Train)
 - 原理的にエイリアスノイズが発生しないらしい
 - よくわからなので@taranai_bot氏に解説は任せた

波形を作るよ

- まずは正弦波
 - $\sin(x)$ を毎フレーム計算すればよいのでは?
 - $\sin$ とか毎回計算してたら重い
 - 前もって1周期分のテーブルを作っておく
 - 間の値は一次補間で得る
 - そこそこ綺麗な正弦波が得られる

波形を作るよ

- 他の波も同様?
 - 三角波、矩形波などは倍音を無限に含む
 - 単純に波形を離散化するとサンプリング周波数の半分の周波数までしか表すことができない（ナイキスト周波数）
 - 消えた倍音はどこへ? → エイリアスノイズ

波形を作るよ

- ではどうするか
 - ナイキスト周波数までの正弦波の足しあわせで波を表現
 - フーリエ級数展開の結果を用いる
 - 周波数によって含むべき倍音の数が変わるので、それぞれについて波形テーブルを作成

波形を作るよ

- その他
 - ノイズ波形
 - ホワイトノイズの場合は振幅を正規乱数で生成すれば良い
 - Box-Mullar法(logと三角関数使うので少し重そう)
 - 正規乱数で作ったテーブルをループぐらいで問題ないかも?
 - 矩形波のduty比を変える
 - 矩形波のパルス幅を変えると倍音の割合が変わる
 - WaveTable形式の場合事前に全ての波形を作っておくしかない?
 - 誰か教えてください

フィルタかけるよ

- 減算式シンセサイザーの構成要素
 - VCO(Oscillator)
 - 基本波形の生成
 - VCF(Filter)
 - 音色の調節
 - VCA(Amplifier)
 - 音量の調節
 - その他
 - LFO, エフェクト

フィルタかけるよ

- デジタルフィルタとは
 - 離散化された信号にいろいろ処理をするもの
 - http://ufcpp.net/study/digital_filter/
- Low-pass Filter, High-pass Filter, Band-pass Filter
 - それぞれ高周波、低周波をカットするフィルタ
 - “Cookbook formulae for audio EQ biquad filter coefficients” (<http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt>)
 - Biquadフィルタという形式のIIRフィルタについてまとめたもの
 - 日本語解説(<http://www.g200kg.com/jp/docs/makingvst/04.html>)
 - LPF, HPF, BPFが簡単に実装できる

音量かえるよ

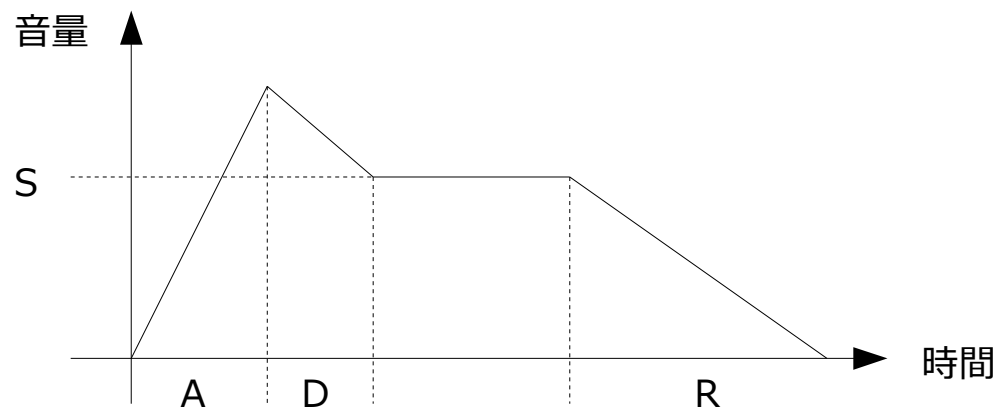
- 減算式シンセサイザーの構成要素
 - VCO(Oscillator)
 - 基本波形の生成
 - VCF(Filter)
 - 音色の調節
 - VCA(Amplifier)
 - 音量の調節
 - その他
 - LFO, エフェクト

音量かえるよ

- ADSR方式

- 鍵盤を押してからの音量の変化の方式

- Attack: 最大音量になるまでの時間
 - Decay: 最大音量になってからSustainまで減衰する時間
 - Sustain: 鍵盤を押し続けた時の音量
 - Release: 鍵盤を離した後音が消えるまでの時間



音量かえるよ

- 音量変化について
 - 人間の耳は振幅の変化を対数スケールで認識する
 - ASDRスロープの変化を指数的にしたほうが自然かも
 - スロープの変化を線形や指数関数から選べるシンセもあるみたい

その他

- パラメータの変化

- パラメータを素早く変えるとクリックノイズが生じる
- ほぼあらゆるパラメータはなめらかに変えるべき
 - 音量、フィルタ

- 非正規化数

- 浮動小数点で計算していると非正規化数というごく小さい値が現れる場合がある
- 非常に処理が重くなるので、ある程度小さくなった値は0にするとよいかも