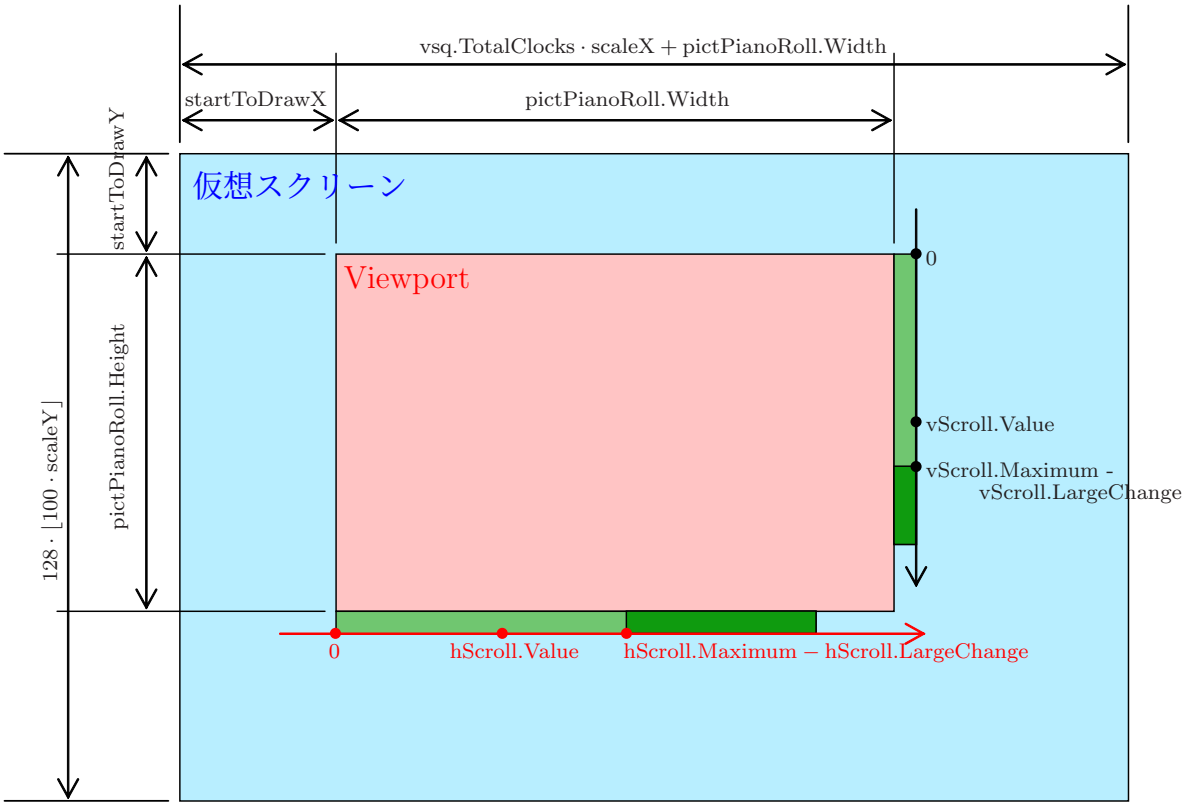
技術情報

kbinani

目 次

1　ピアノロールのスクロール動作の計算

水平および垂直スクロールバー・コントロールの設定と，画面の描画状態との関係を決めるための計算方法を述べる．



上図のように，ビューポートのオフセットは startToDrawX と startToDrawX で定義される．また，水平スクロール・コントロール hScroll の Value プロパティーの単位は clock，垂直スクロール・コントロール vScroll の Value プロパティーの単位は cent である．ピアノロールの画面上の横方向の表示倍率を scaleX[pixel/clock]，縦方向の表示倍率を scaleY[pixel/cent] とする．

startToDrawX，startToDrawY は次で定義される：

$$\begin{cases} \text{startToDrawX} = \text{hScroll.Value} \cdot \text{scaleX}, \\ \text{startToDrawY} = \text{vScroll.Value} \cdot \text{scaleY}. \end{cases} \tag{1}$$

1.1　水平スクロールバー

startToDrawX の最大値および最小値は次のように決める．

$$\max(\text{startToDrawX}) = \text{vsq.TotalClocks} \cdot \text{scaleX}, \tag{2}$$

$$\min(\text{startToDrawX}) = 0. \tag{3}$$

水平スクロールバーが一番右にスクロールされたときに，hScroll.Value が最大値をとらねばならないから，startToDrawX の最大値を scaleX で割ったものが，hScroll.Value の

最大値となるようにする．また，水平スクロールバーの動作の仕様から hScroll.Value の最大値は hScroll.Maximum − hScroll.LargeChange であるので，次の式が成り立つ．

$$\frac{\max(\text{startToDrawX})}{\text{scaleX}} = \text{hScroll.Maximum} - \text{hScroll.LargeChange} \tag{4}$$

さらに，外観上スクロールバーの可動範囲とスクロールボックスの幅の比が，仮想スクリーンとビューポートの幅の比と同一であると美観が良いので，次の式が成り立つようにする．

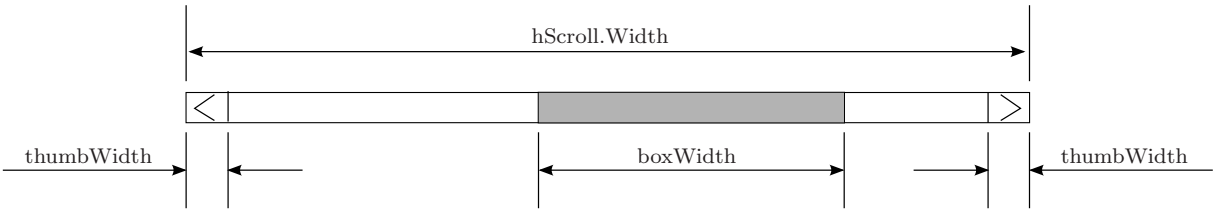
$$\frac{\text{pictPianoRoll.Width}}{\text{vsq.TotalClocks} \cdot \text{scaleX} + \text{pictPianoRoll.Width}} = \frac{\text{hScroll.LargeChange}}{\text{hScroll.Maximum}} \tag{5}$$

式 (4) と式 (5) を連立させて解くと，hScroll.Maximum と hScroll.LargeChange は次で定まる．

$$\text{hScroll.Maximum} = \text{vsq.TotalClocks} + \frac{\text{pictPianoRoll.Width}}{\text{scaleX}}, \tag{6}$$

$$\text{hScroll.LargeChange} = \frac{\text{pictPianoRoll.Width}}{\text{scaleX}}. \tag{7}$$

ここで，スクロールボックスの実際の表示幅がいくらになるかを調べる．水平スクロールバーの構造はだいたい下図のようになっている．



スクロールバーの動作の仕様から，次が成り立つ．

$$\frac{\text{boxWidth}}{\text{hScroll.Width} - 2 \cdot \text{thumbWidth}} = \frac{\text{hScroll.LargeChange}}{\text{hScroll.Maximum}}. \tag{8}$$

従って，

$$\text{boxWidth} = \frac{\text{hScroll.LargeChange}}{\text{hScroll.Maximum}}(\text{hScroll.Width} - 2 \cdot \text{thumbWidth}). \tag{9}$$

この boxWidth は，式 (6) と式 (7) により決まるが，この大きさがある値以下になると非常に扱いづらくなると考えられる．そこで，この boxWidth の最小値を min(boxWidth) とし，式 (6) と式 (7) から求めた値がこの値より小さくなる場合は，以下の要領で計算することにする．

boxWidth を値が min(boxWidth) の定数として扱う．すると，式 (5) は成り立たなくなるが，これは仕方ない．見た目が悪いよりも，スクロールボックスが操作しにくいことの方が駄目だと思うからである．従って最初式 (4) と式 (5) を連立させて解いた代わりに，式 (4) と式 (9) を連立させて解く．解は，

$$\text{hScroll.Maximum} = \frac{\text{vsq.TotalClocks}\,(\text{hScroll.Width} - 2 \cdot \text{thumbWidth})}{(\text{hScroll.Width} - 2 \cdot \text{thumbWidth}) - \text{boxWidth}}, \quad (10)$$

$$\text{hScroll.LargeChange} = \frac{\text{vsq.TotalClocks} \cdot \text{boxWidth}}{(\text{hScroll.Width} - 2 \cdot \text{thumbWidth}) - \text{boxWidth}}, \quad (11)$$

となる．

1.2 垂直スクロールバー

startToDrawY の最大値および最小値は次のように決める．

$$\max(\text{startToDrawY}) = 128 \cdot \lfloor 100 \cdot \text{scaleY} \rfloor - \text{pictPianoRoll.Height}, \quad (12)$$

$$\min(\text{startToDrawY}) = 0. \quad (13)$$

垂直スクロールバーが一番下にスクロールされたときに，vScroll.Value が最大値をとらねばならないから，startToDrawY の最大値を scaleY で割ったものが，vScroll.Value の最大値となるようにする．また，垂直スクロールバーの動作の仕様から vScroll.Value の最大値は vScroll.Maximum − vScroll.LargeChange であるので，次の式が成り立つ．

$$\frac{\max(\text{startToDrawY})}{\text{scaleY}} = \text{vScroll.Maximum} - \text{vScroll.LargeChange}. \quad (14)$$

さらに，外観上スクロールバーの稼動範囲とスクロールボックスの高さの比が，仮想スクリーンとビューポートの高さの比と同一であると美観が良いので，次の式が成り立つようにする．

$$\frac{\text{pictPianoRoll.Height}}{128 \cdot \lfloor 100 \cdot \text{scaleY} \rfloor} = \frac{\text{vScroll.LargeChange}}{\text{vScroll.Maximum}}. \quad (15)$$

式 (14) と式 (15) を連立させて解くと，vScroll.Maximum と vScroll.LargeChange は次で定まる．

$$\text{vScroll.Maximum} = \frac{128 \cdot \lfloor 100 \cdot \text{scaleY} \rfloor}{\text{scaleY}}, \quad (16)$$

$$\text{vScroll.LargeChange} = \frac{\text{pictPianoRoll.Height}}{\text{scaleY}}. \quad (17)$$

ここで，スクロールボックスの実際の表示高さがいくらになるかを調べる．水平スクロールバーの場合と大体同じであるので，スクロールボックスの高さ boxHeight について，次が成り立つ．

$$\text{boxHeight} = \frac{\text{vScroll.LargeChange}}{\text{vScroll.Maximum}}(\text{vScroll.Height} - 2 \cdot \text{thumbHeight}). \quad (18)$$

水平スクロールバーのときと同様に，boxHeight を，値が min(boxHeight) の定数として扱う．式 (14) と式 (18) を連立させて解く．解は，

$$\begin{aligned}
\text{vScroll.Maximum} &= \frac{128 \cdot \lfloor 100 \cdot \text{scaleY} \rfloor - \text{pictPianoRoll.Height}}{\text{scaleY}} \times \\
&\quad \frac{\text{vScroll.Height} - 2 \cdot \text{thumbHeight}}{\text{vScroll.Height} - 2 \cdot \text{thumbHeight} - \text{boxHeight}}, \qquad (19) \\
\text{vScroll.LargeChange} &= \frac{128 \cdot \lfloor 100 \cdot \text{scaleY} \rfloor - \text{pictPianoRoll.Height}}{\text{scaleY}} \times \\
&\quad \frac{\text{boxHeight}}{\text{vScroll.Height} - 2 \cdot \text{thumbHeight} - \text{boxHeight}}, \qquad (20)
\end{aligned}$$

となる.

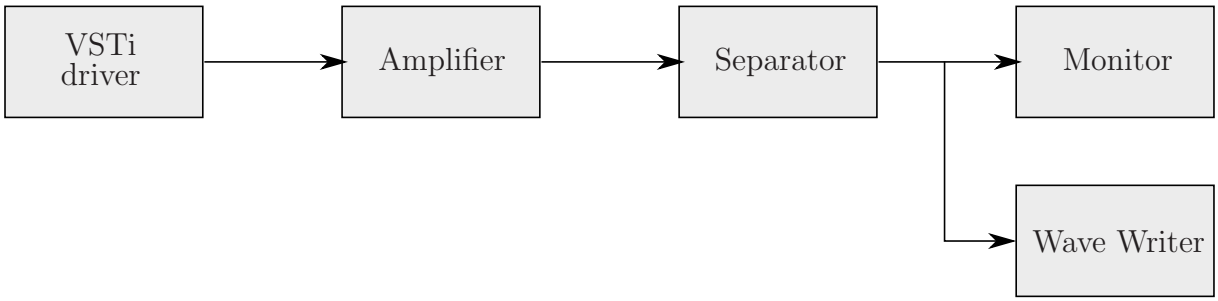
2　波形合成システム

2.1　概要

波形合成システムは，波形を処理する幾つかのクラスのインスタンスを，ブロック線図のようにつなぎ合わせて実現される．ここでいう「波形を処理」とは，

1. 生成
2. 増幅
3. 重ね合わせ
4. 分割

等のことである．例えば，VOCALOID VSTi を駆動してその音声波形をスピーカーからモニターすると同時に，Wave ファイルに記録する場合は，次のようなクラス間の接続が為されている．



上図では，波形を生成する生成器から，増幅器，分割器，音声モニター，および Wave 出力器が接続されている．この節では，これらを実装するための基底クラスや，それぞれの機能ごとの実際のクラス簡易なリファレンスと，使用例を述べる．

2.2　基底クラス: WaveUnit

このクラスは，波形を処理するクラスの継承元となる基底クラスである．クラスの外観は以下のようになっている．

```
class WaveUnit{
    public abstract int getVersion();
    public abstract void setConfig( string parameter );
    public virtual void setGlobalConfig(
        org.kbinani.cadencii.EditorConfig config ){};
    public virtual void paintTo(
        System.Drawing.Graphics graphics,
        int x,
        int y,
        int width,
        int height ){};
}
```

2.3 インターフェース

波形処理の種類には，波形を (active に) 生成する処理，波形を (passive に) 生成する処理，および波形を受け取る処理の 3 種類に大別できる．ここでいう active，passive な処理とは，それぞれ，自律して波形を生成し他の処理装置に波形を送り出す処理，および他からの波形の要求に応じて必要な量の波形を生成する処理のことをいう．

2.3.1 WaveReceiver

2.3.2 WaveSender

2.3.3 WaveGenerator

2.4 WaveGenerator

2.4.1 VOCALOID

2.4.2 vConnect-STAND

2.4.3 UTAU

2.4.4 AquesTone

2.4.5 無音

2.5 WaveSender

2.5.1 WAVE ファイル読込み

2.6 WaveReceiver

2.6.1 モニター

2.6.2 WAVE ファイル書込み

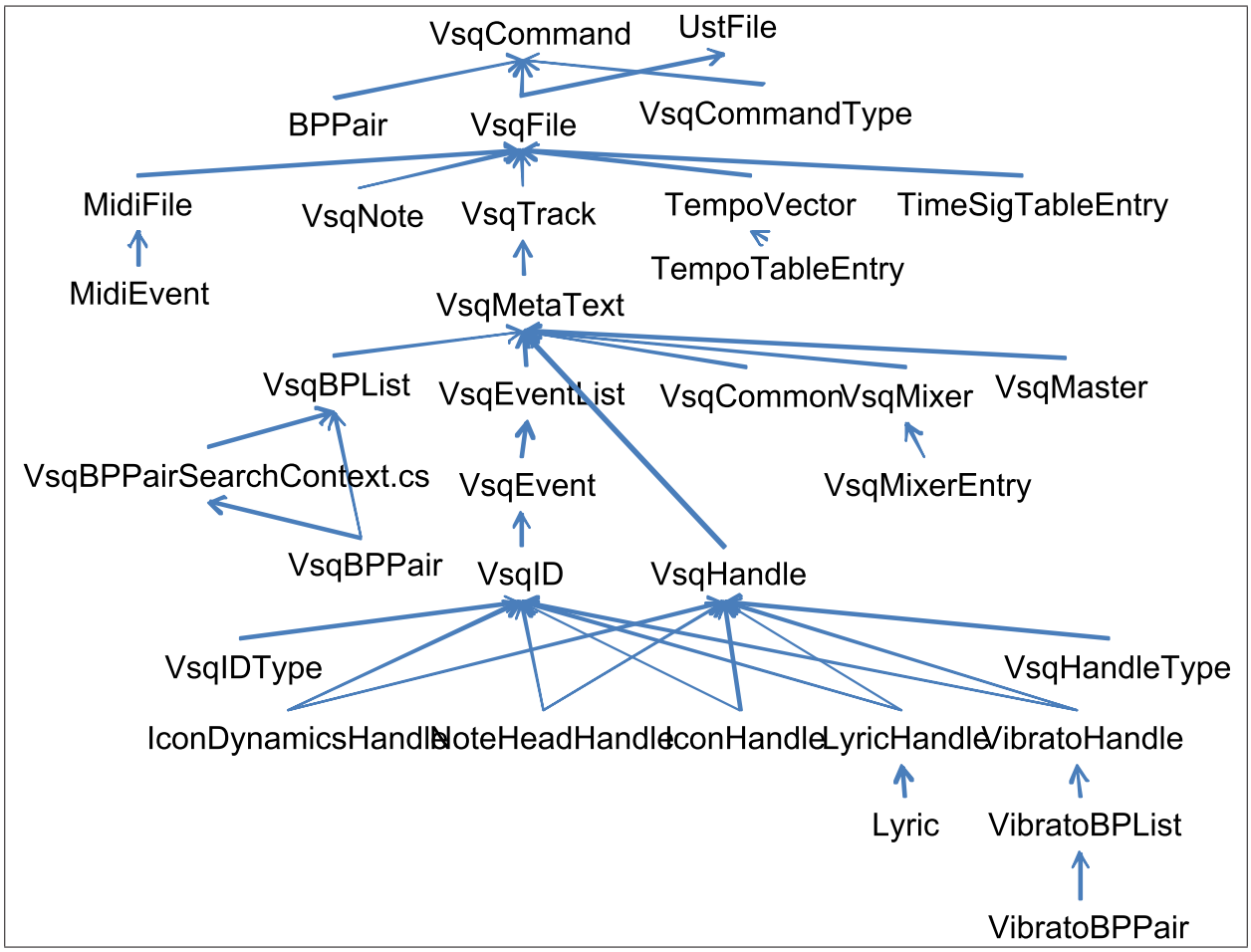
2.7 WaveSender かつ WaveReceiver

2.7.1 増幅器

2.7.2 重ね合わせ

2.7.3 分割器

3　org.kbinani.vsq名前空間にあるクラスの依存関係



4　動作確認用のチェックリスト

4.1　File メニュー

4.1.1　New

クリックしたとき，以下の動作が起こること	
再生中の場合，再生が停止する	
シーケンスが編集中の場合，編集を取り消す旨のダイアログが出る	
Cancel ボタンを押すとシーケンスの編集に復帰する	
OK ボタンを押すと以降に処理が続行される	
シーケンスが新規に作成される	
AppManager.getSelectedEventCount() が 0 になる	
プロパティー・ビューがリセットされる	
ミキサー・ダイアログがリセットされる	
Undo/Redo 履歴がリセットされる	
自動保存を行うためのタイマーが停止される	
AppManager の TempWaveDir プロパティーがリセットされる	
マウス・オーバー時に以下の動作が起こること	
ステータスバーにこのメニューの概要が表示される	

4.1.2　Open

クリックしたとき，以下の動作が起こること	
再生中の場合，再生が停止する	
シーケンスが編集中の場合，編集を取り消す旨のダイアログが出る	
Cancel ボタンを押すとシーケンスの編集に復帰する	
OK ボタンを押すと以降に処理が続行される	
XVSQ を開くためのダイアログが出る	
ダイアログのデフォルトの拡張子が*.xvsq になっていること	
ダイアログの最初のディレクトリが，前回 xvsq ファイルを開いた，または保存した場所になっていること	
Cancel ボタンを押すとシーケンスの編集に復帰する	
Open ボタンを押すと以降に処理が続行される	
指定した XVSQ ファイルの内容が読み込まれ，編集可能な状態になる	
AppManager.getSelectedEventCount() が 0 になる	
プロパティー・ビューが XVSQ の内容に応じて更新される	
ミキサー・ダイアログが XVSQ の内容に応じて更新される	
Undo/Redo 履歴がリセットされる	
自動保存を行うためのタイマーが停止される	

AppManager の TempWaveDir プロパティーが XVSQ の内容に応じて更新される	
マウス・オーバー時に以下の動作が起こること	
ステータスバーにこのメニューの概要が表示される	

4.1.3 Save

クリックしたとき，以下の動作が起こること	
再生中の場合，再生が停止する	
自動保存を行うためのタイマーが動作している場合，停止される	
シーケンスがまだ一度も XVSQ に保存されていない状態の場合，以下の動作が起こる	
XVSQ を保存するためのダイアログが出る	
ダイアログのデフォルトの拡張子が*.xvsq になっていること	
ダイアログの最初のディレクトリが，前回 xvsq ファイルを開いた，または保存した場所になっていること	
Cancel ボタンを押すとシーケンスの編集に復帰する	
Save ボタンを押すと以降に処理が続行される	
シーケンスが指定したパスに XVSQ ファイルとして保存される	
拡張子が xvsq でない場合，末尾に xvsq が追加される	
合成結果のキャッシュが指示されている場合，xvsq と同じディレクトリに末尾が「.cadencii」で終わるディレクトリが作成され，合成結果の WAVE ファイルと，合成ステータスを格納した xml ファイルがトラックごとに出力される	
自動保存を行うためのタイマーが開始される	
マウス・オーバー時に以下の動作が起こること	
ステータスバーにこのメニューの概要が表示される	

4.1.4 Save as

クリックしたとき，以下の動作が起こること	
再生中の場合，再生が停止する	
自動保存を行うためのタイマーが動作している場合，停止される	
XVSQ を保存するためのダイアログが出る	
ダイアログのデフォルトの拡張子が*.xvsq になっていること	
ダイアログの最初のディレクトリが，前回 xvsq ファイルを開いた，または保存した場所になっていること	
Cancel ボタンを押すとシーケンスの編集に復帰する	
Save ボタンを押すと以降に処理が続行される	
シーケンスが指定したパスに XVSQ ファイルとして保存される	
拡張子が xvsq でない場合，末尾に xvsq が追加される	

合成結果のキャッシュが指示されている場合，xvsq と同じディレクトリに末尾が「.cadencii」で終わるディレクトリが作成され，合成結果の WAVE ファイルと，合成ステータスを格納した xml ファイルがトラックごとに出力される	
自動保存を行うためのタイマーが開始される	
マウス・オーバー時に以下のの動作が起こること	
ステータスバーにこのメニューの概要が表示される	

4.1.5 Open VSQ/Vocaloid MIDI

4.1.6 Open UTAU project file

4.1.7 Import

4.1.7.1 VSQ/Vocaloid MIDI

4.1.7.2 Standard MIDI

4.1.8 Export

4.1.8.1 Wave

4.1.8.2 VSQ file

4.1.8.3 MIDI

4.1.8.4 MusicXML

4.1.8.5 UTAU project file

4.1.8.6 Metatext for vConnect

4.1.9 Recent files

4.1.10 Quit