

# 技術情報

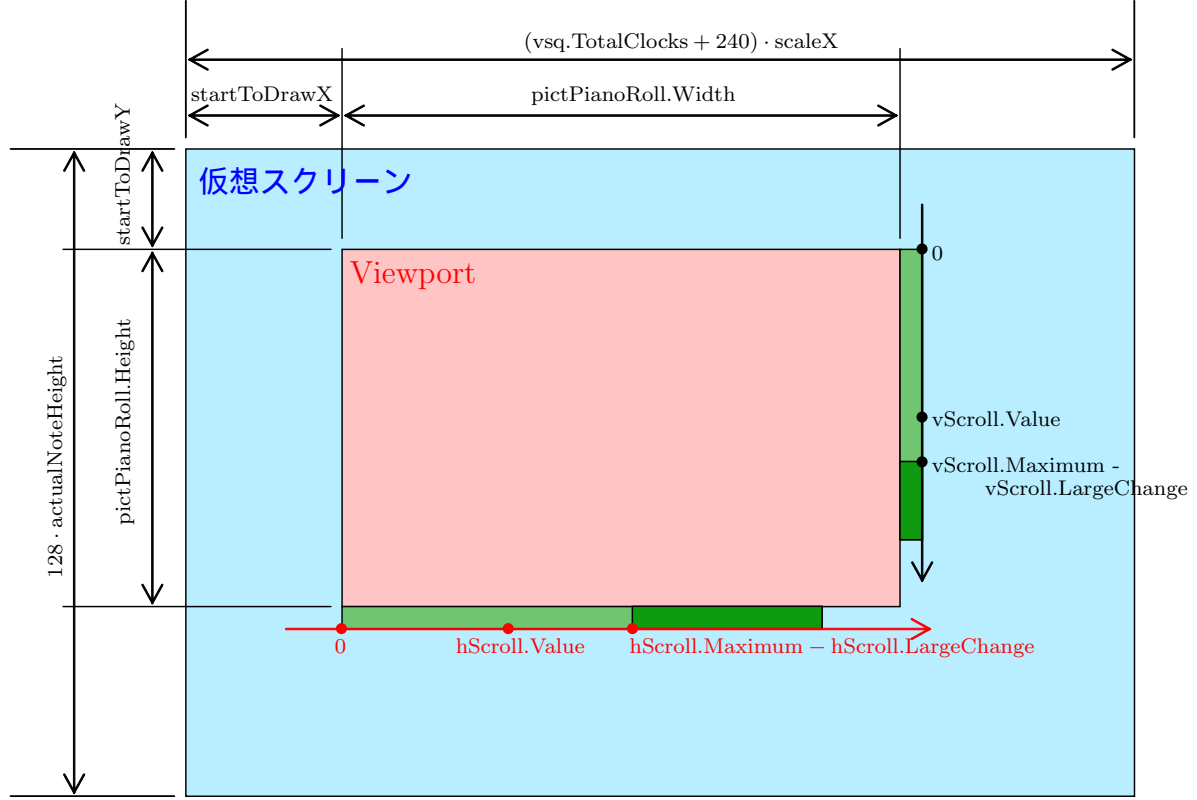
kbinani

## 目 次

1	ピアノロールのスクロール動作の計算	3
1.1	水平スクロールバー	3
1.2	垂直スクロールバー	5

# 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 `startToDrawY` 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` プロパティの単位は `pixel` である。ピアノロールの画面上の表示倍率を `scaleX[pixel/scale]` とする。

`startToDrawX`、`startToDrawY` は次で定義される：

$$\begin{cases} \text{startToDrawX} = \text{hScroll.Value} \cdot \text{scaleX}, \\ \text{startToDrawY} = \text{vScroll.Value}. \end{cases} \quad (1)$$

## 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。

$$\begin{cases} \max(\text{startToDrawX}) = (\text{vsq.TotalClocks} + 240) \cdot \text{scaleX} - \text{pictPianoRoll.Width}, \\ \min(\text{startToDrawX}) = 0. \end{cases} \quad (2)$$

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．

$$\frac{\max(\text{startToDrawX})}{\text{scaleX}} = \text{hScroll.Maximum} - \text{hScroll.LargeChange} \quad (3)$$

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．

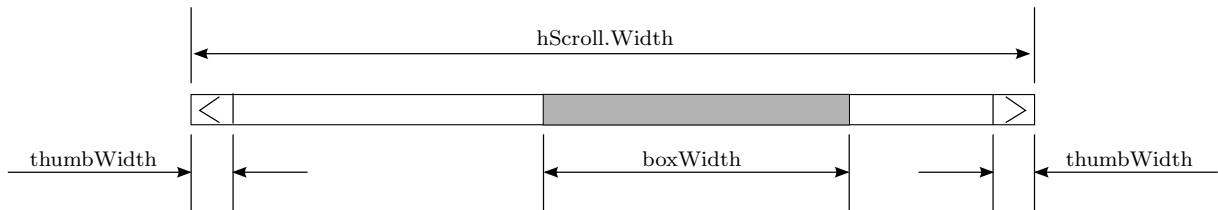
$$\frac{\text{pictPianoRoll.Width}}{(\text{vsq.TotalClocks} + 240) \cdot \text{scaleX}} = \frac{\text{hScroll.LargeChange}}{\text{hScroll.Maximum}} \quad (4)$$

式 (3) と式 (4) を連立させて解くと，`hScroll.Maximum` と `hScroll.LargeChange` は次で定まる．

$$\text{hScroll.Maximum} = \text{vsq.TotalClocks} + 240, \quad (5)$$

$$\text{hScroll.LargeChange} = \frac{\text{pictPianoRoll.Width}}{\text{scaleX}}. \quad (6)$$

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．

$$\frac{\text{boxWidth}}{\text{hScroll.Width} - 2 \cdot \text{thumbWidth}} = \frac{\text{hScroll.LargeChange}}{\text{hScroll.Maximum}}. \quad (7)$$

従って，

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

この `boxWidth` は，式 (5) と式 (6) 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 (5) と式 (6) 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 (4) 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 (3) と式 (4) を連立させて解いた代わりに，式 (3) と式 (8) を連立させて解く．解は，

$$\text{hScroll.Maximum} = \frac{(\text{hScroll.Width} - 2 \cdot \text{thumbWidth})}{(\text{hScroll.Width} - 2 \cdot \text{thumbWidth}) - \min(\text{boxWidth})} \times \left\{ (\text{vsq.TotalClocks} + 240) - \frac{\text{pictPianoRoll.Width}}{\text{scaleX}} \right\}, \quad (9)$$

$$\text{hScroll.LargeChange} = \frac{\min(\text{boxWidth})}{(\text{hScroll.Width} - 2 \cdot \text{thumbWidth}) - \min(\text{boxWidth})} \times \left\{ (\text{vsq.TotalClocks} + 240) - \frac{\text{pictPianoRoll.Width}}{\text{scaleX}} \right\}, \quad (10)$$

となる .

## 1.2 垂直スクロールバー