

ステータス:	終了	開始日:	2010/02/18
優先度:	通常	作業時間の記録:	0.00時間
担当者:			
カテゴリ:			
対象バージョン:			
説明			
譜面モニタにテンポを表示する機構を実装			

**履歴**

#1 - 2014/01/07 18:58 - Redmine Admin

; 《Ver5.6 開発後記》 2010.02.17

;

; Museに譜面モニタを実装して10年近くが経とうとしています。MMLコーディング作業は音符タイミングずれのミスが発生しやすく、その確認もままなりません。譜面モニタが実装される前は、そのデバッグ効率を考えると各フィンガーに記述されている拍数をカウントし、タイミングずれのフィンガーを検出しやすくするためのダイアログを作りました。それが現在のフィンガー情報ウィンドウです。しかし譜面モニタが出来てから、全くと言っていい程Musingの際にそれを参照していない自分に気づきました。気づくのには10年も掛かってしまった(笑)。そういう目で改めて眺めてみると、この情報エリアは闇雲に作業スクリーンを圧迫しているにすぎません。当時、かなりの時間を掛けて音符のピットマップを作ったり、音長の演算構成の工夫もいろいろと仕込んだりして、愛着がある機能なのですが、無駄なものには邪魔なもの。思い切って取り払うことにしました。まるで機能ダウンのようですが、こういうバージョンアップがあっても良いと個人的には思っています。これでウィンドウ配置が楽になります。一種のエコですね(苦笑)。

; ついでに、フィンガー表示のフッターやヘッダー部をダブルクリックすると、リスト上に存在しているフィンガーをピッタリとストレッチ状態にする機構も実装してしまいました。もう一回ダブルクリックするとコンパクトになります。加えてメンバーIDのキー入力で該当行がフォーカスするようにし、参照したいメンバーをスピーディに見つけることができるようにしました。

; メンバー情報ダイアログも、ちょっとした変更を行いました。視認性を高めるため、行頭のマークをマークに変更しました。全体のデザインがシャープになった気がします。

; さてもう一つ、最後にご紹介する強化は、譜面モニタに関する新しい機構です。実は、この強化が他を圧倒して数倍も苦労してます。その操作性、デザイン、フォント視認性から、内部のデータ構造、高速処理を維持するためのアルゴリズムまで検討は多岐に及び、そしてMuseプログラムのかなりの部分に手を入れることになりました。何度も習作を作り上げ、そして陶芸家のように釜から出した作品を叩き割ることを繰り返しました。

; こうして出来上がったのがテンポ属性の譜面重ね表示です。以前より、譜面モニタ上にメンバー属性やフィンガー属性を表示して欲しいという要望が多数あり、私もそれを実現できないか考え続けていました。当然、初めに思い付くのはグラフ表示です。他のDTMソフトでもそういったデザインが多いですから。しかし、グラフは見栄えは良いのですが表示エリアをかなり取り、ましてや属性に変化が無い場合単なる水平線が延々と続くだけで、そのためにMusing作業のスクリーンエリアが圧迫されることに納得できず、開発する気になれずにいました。

； 今回閃いたのは、属性表記をそのまま描いてしまうデザインです。まさにMML  
； ソフトウェアならではの発想だと自負しております。出来上がってみると、グラ  
； フよりもMuseデータとの対応付けが容易で、格段に優れている事がわかりました。  
； 短い間隔で長めの属性表記があると文字が重なってしまうので、上下二段にして  
； 回避する工夫も施しました。むろん回避しきれない場合もありますが、頻度的に  
； 問題無いでしょう。図らずも人間味を感じる配置となり気に入っています。  
； しかし肌理細かいテンポ指定が無い場合には、音符上部の表示だけでは現在のテ  
； ンポがどうなっているのかわからない状態になってしまいます。そこで音部記号  
； エリアに、直前で指定されたテンポの最終値を表示する配慮も行いました。実は  
； この処理の高速化が大変でした。このテンポ表示自体のON/OFF機構も加え  
； ました。OFFにしてしまえば、従来通りのスッキリとした譜面モニタのフィー  
； ルをほぼ継承する事ができます。起動直後は敢えてスッキリ状態にしてあります  
； ので、テンポ変化のある曲でぜひ音部エリアの<>部分をクリックしてみてください。  
； 今回対応したのはテンポ属性のみですが、これだけでもテンポ変化の激しいクラ  
； シックなどのMusingには絶大な威力を発揮すると確信しています。そして、今回  
； の開発を通じて、属性値を譜面モニタに描画する方向性が見えてきました。今後  
； は音量などに代表されるメンバー属性、フィンガー属性をどう扱うかを継続的に  
； 検討していきたいと思っています。全体属性のテンポと異なり、メンバーやフィ  
； ンガーの表示仕分けや指定方法の考察が必要になってくるので難度は増しますが、  
； 実に挑戦し甲斐のあるテーマです。布石としてフィンガー情報選択行を目立つよ  
； うにしておきました。もうすぐ10周年を迎える譜面モニタ。新しい開発展開へ  
； の道が開けた事にワクワクしています。クラフトマンシップ、万歳！