

ステータス:	終了	開始日:	2012/08/18
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			
PDF楽譜の出力機構を実装。(要LilyPondインストール)			
関連するチケット:			
関連している Bug # 55: (V6.20)初期化ファイル(muse.ini)に指定したテキスト背景色が正しく...	終了	2011/08/19	2011/08/19
関連している Bug # 56: (V6.20)未定義の起動パラメータ文字を指定するとシステムエラーのダイアログ終了	終了	2012/08/19	2012/08/19

履歴

#1 - 2014/01/08 08:07 - Redmine Admin

; 《Ver6.2 開発後記》2012.08.16

;

; Museメニューの「エクスポート」が、何故「MIDIファイル出力」ではなかったのかという、それは開発当初からいつか実装したいと思っていた機構があって、それをこのメニューの階層下に入れたかったからです。でも、その実現の可能性は全く見えず、もどかしい思いを抱きつつ今に至っております。

; しかし、前回のサウンドフォント対応で他ソフトとの連携という世界に足を踏み入れたことから「ならば一丁試してみるか」というノリで実験プログラムを書き始めました。相手は楽譜作成フリーソフトのLilyPondです。Museと同様のテキストインターフェース。オープンソースで世界中にユーザがいます。伴侶としてこれ程頼りになるソフトはありません。サイトもとても端麗に作り込まれています。

; まず、mid2musの様に都度Museデータから変換する方式にするか、それともMuseコンパイル処理と一体化するかの方針決定が必要でした。初めは前者を考えていました。しかし80%以上がMuseコンパイルと同様の処理であることに気づき、後者に方針変更しました。後者の方式は、楽譜出力元の情報とMuseファイルの一致を保証出来る事、メンバーON/OFF状態などにも連動した楽譜が作れる事等メリットが多いのです。しかし譜刻もしないのに、楽譜を記譜するための情報をMusingの際に抱え続けるのはメモリ効率が劣悪です。そこで極限まで情報を圧縮して保持しておき、PDFエクスポート時にそれを展開処理するという方式を採りました。

; 方針としては間違っていなかったと思うのですが、これが地獄のように大変なプログラミングになりました。コンパイル時間も結構掛かるようになってしまいました。しかも不思議な事に構築時よりもロード時(それもメモリ解放)に酷く時間が掛かるのです。基本関数のfree()が原因ですから、プログラムの見直しでは速度アップは図れそうにありませんでした。更に不思議なのが、WinXPだと速度があまり低下しないのです。これはOSのアーキテクチャとの絡みかもしません。

; そこでVC++6.0を卒業し、VS2008にステップアップしてみました。(といっても相も変わらずコマンドラインで生のC言語のみ利用なんですが・・・)

; すると、なんと呆気なく速くなったのです！ 実際、楽譜機構を備えていない今までのMuseより倍以上速いです。まるで、大リーグボール養成ギブスを外したような気分です。Muse(V3.6)で実装した譜面モニタが、大リーグボール1号だとすれば、今回の楽譜PDFへの出力は大リーグボール2号かもしれません。VS2008にしたことでmuse.exeの容量が一気に増えましたが、それでもなおReadme.txtより小さいです(笑)。VS2010だと更にほんの少し高速になったのですが、WAVEコマ

ンドが不安定になるという副作用があったので今回は見送りました。

極力今まで蓄積したMuseデータに手を加えることなく、それなりの楽譜が出るようにするのが当面の目標でしたが、デジタルなMuseデータをアナログな楽譜に変換するのは至難の業でした。演奏情報であるMuseデータの方が楽譜データよりも格段に情報量が多いため、そこから適切な情報を選別し、かつ適度な情報量に抑さえ込まなければ、人が見て自然な楽譜になりません。譜刻の体裁はLilyPondがよろしくやってくれるのですが、その内容をセットするのはこちらの責任です。構成する音符の音長がバラバラな和音、和音の中の連符や連符の中の和音、そしてその中のコード記述、微妙な修飾音の判断と少し短い音長の組み合わせ・・・。

頭の中でLilyPondの文法とMuse文法が交錯します。楽譜上のタイミングがズレ無い様に音長を貯金したり借金したりしながら、様々な状況に対応しなければなりませんでした。もう、しばらく楽譜なんて見るのもイヤです・・・(ウソです)。

こうして苦勞の末にかなりの自動化を果たしたつもりですが、楽譜構成や全体の体裁等はやはり追加情報として与える必要があります。出し過ぎてしまう自動記譜を抑止したり再開したりする制御も必要です。それに、LilyPondの多様な機能を満喫してもらうためにパススルー命令も欲しい所です。これらを従来のMuse文法と一貫性を保ってまとめ上げなければなりません。いくつもの習作を重ねた結果、LILYコマンドとスルー指定の2つの表記を追加するだけのシンプルで美しい文法が完成しました。まあ、機能区分で逃げている感もありますが(苦笑)。

とは言え、どうしても限界があります。例えば和音要素が複雑な場合、長い音長は全音符にカットされてしまいます。和音内の色々な属性や連符内コードネームも記譜されません。細かい音長のトレモロはそのまま出てしまうので大抵の場合用紙からはみ出て見えなくなります。1つの五線に複数フィンガーを集約し記譜する場合等も、はみ出して見えなくなる傾向にあります。この様に、結構・・・

“消え”ます。なんせ大リーグボール2号ですから(笑)。ちなみに、LilyPondの処理はかなり時間が掛かります。処理中は、Muse自体が(応答なし)になりますが、ウィンドウタイトルの「楽譜生成中...」が消えるまで気長にお待ち下さい。

って、これもしかして大リーグボール3号なのお～？。

それと交響曲などの大作はLilyPondが悲鳴を上げて落ちます。別プロセスで起動させているのでMuseは継続できますが、長編作品は投入しない方が無難でしょう。マクロは繰返し記号ではなく全部展開してしまいます。なのでヴェクサシオンは、確実に落ちます。3名の方がヴェクサシオンを作ってますが、全滅でした。(笑)

あと譜面モニタ絵画も危ないので、v0はオミットすることにしました。v0を休符で記譜する仕様も試しましたが、絵画が素晴らしい程ハングするので止めました。私の様に楽譜参照でMusingしているミュージアにとって、打ち終わったデータを楽譜出力するなんて一体何の意味があるのでしょうか。絶壁の様に立ちはだかるプログラミング課題を目の前に「こんなに苦勞して果たしてこの機構を提供する意義はあるのだろうか？」と自問の日々でした。しかしこの機構は新しくも素晴らしいMusingの世界を切り開いてくれるという予感を感じたのも事実です。確かにいくつか実用的な局面は思い付きます。(1)Museで作曲した楽曲を他者に楽譜で渡せる。(2)耳コピした結果を楽譜として残せる。(3)例えばピアノ譜を弦楽四重奏曲に編曲するといった場合も楽譜化は有効。(4)単純な楽譜打込みであっても楽譜同士を比較する事でデバッグに役立つ。とまあ、こんな所でしょうか。

今回のバージョンアップに伴い、コマンドライン引数の文字アサインを見直しました。引数付きのショートカットで起動している皆様、お手数ですが手直しをお願いします。そして、関連ソフト開発者の皆様、仕様を変更してごめんなさい。それと、LILYコマンドで必要に迫られたため、を"に読替える仕様を立ち上げ、その仕様を従来のTEXT系コマンドにも適用しました。これでテキストエリアにもダブルコーテーションを表示できるようになりました。でも、この対応でいくつかの作品に互換性の破れが(単にテキストエリアの表示上の問題ではありませんが)見られました。申し訳ありません。

今回の開発の総合テストでは、皆さんから頂いた手元の保有データを大いに活用させてもらいました。譜刻パターンは実に多彩で、とても一人の頭の中で網羅し

； きれません。その意味で、現有8,000曲を越える様々なデータ群は最強のテスト
； シナリオです。感謝の気持ちを胸に抱きながら、バグと格闘し続けました(苦笑)。
； さあてと、綺麗な譜刻ができる様、久々のmid2mus改版をしようかなあ～。でも、
； その前に一息つきたい。皆さんの今までの演奏を堪能し英気を養わせて下さい。
； ここ2ヶ月の間、ただただ楽譜を眺めるばかりで、肝心の“音楽”をほとんど耳
； にしていない私なのです。