

ステータス:	終了	開始日:	2010/01/10
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			
コンパイルエンジンにて同時刻内ノートON/OFFの順序関係最適化			
関連するチケット:			
関連している Bug # 35: (V5.50)途中から再生(シーク)させると正しく演奏されない場合がある		終了	2010/02/07 2010/02/07

履歴

#1 - 2014/01/07 18:53 - Redmine Admin

; 《Ver5.5 開発後記》2010.01.09

;

; 今回の強化は外部仕様として目立った変化は無いため、メジャーなバージョンア

; ヅップの扱いをすべきか否か悩みましたが、Museの中核を成すコンパイル・エンジ

; ンに手を入れたのでメジャーアップに踏み切りました。

; 7年程前から多重ノートONの話題が断続的に出ていましたが、要因はひとえに

; 音源性能の問題であると認識し、Muse自体での対応は範疇外であると決め付けて

; いたのです。しかし今回、るうさんから同時刻内に限ってノートOFFを前方に

; シフトさせれば多重ノートONの状態を抑制できるという提案を受けました。

; 言われてみれば、何故7年間もそれに気付かなかったのか不思議なほどシンプル

; で美しく、そして効果的なアイデアです。多分音源の処理負荷に対しても望ま

; しい結果を与えるだろう事が予想できます。るうさんに心から感謝いたします。

; 今回の件で、一つの大切なことを学んだ気がします。人は何らかの不都合な事象

; に出合った時、その原因を他者のせいにした途端に新たな閃きの機会を失ってし

; まうという事です。少なくともそれを解消しようとする精神は希薄になります。

; 今回の改良は単にノートOFFを前方にシフトするだけでなく、中盤にコントロ

; ールチェンジ等の制御系コマンドを配し、最後にノートONを配置するという仕

; 様にしました。アルゴリズムは複雑になりますが、これにより発音前に制御コマ

; ンドが音源に送信され、演奏シーケンスとして単に多重ノートONを抑制する以

; 上の効果があると推測しています。

; また、シフト処理は演奏データ全域で探索する必要があるため、一音ずつ実行さ

; せるとコンパイル速度の低下が懸念されます。そこで、連続するノートOFFや

; ノートONをグループ化して一気にシフトするという方式を取りました。これは

; これで、大変神経を使うデータ処理アルゴリズムとなりました。

; これらの処理をメモリ効率に配慮しながら1パスで完了させるという目標を立て

; て改良に臨みました。もちろん、Muserの記述した演奏順番を極力尊重するという

; 条件も付加しています。久々の心臓部への改訂だったため、開発は苦労の連続で

; したが、すべての設定目標を満足させる出来になったと思います。

; 今回のバージョンアップでは、楠本さんが発見してくれたバグにも対応しました。

; 中身の無いマクロをある条件で記述すると確実にMuseがハングするという不具合

; です。楠本さん、ご報告本当にありがとうございました。

; マクロ処理は再帰呼出しや処理ポインタのリワインドなど、かなり複雑なシーケ

; ンス部分です。アルゴリズムを細部に渡るまでチェックし、マクロ解析用の作業

; メモリを解放する際、範囲を記憶するポインタの制御に不備があり、管理外の領

； 域をアクセスする可能性を見つけ出しました。

； この致命的な障害は、マクロ機構提供当初より抱えていた事になります。プログ

； ラマとしてとても恥ずかしいミスだと反省しています。こんな私ですが、これが

； らもMuseに愛を込めて育てて参りますので、皆さんよろしくお願いします。