

Muse 取扱説明書

加藤一郎

目次

第 1 章	アプリケーション概説	1
1.1	Muse とは何か	1
1.2	インストールとアンインストール	1
1.3	Muse データの基本的な作成手順	1
1.4	機能概要	2
1.4.1	Muse ファイルのロード	2
1.4.2	演奏の仕方	2
1.4.3	シークバーの扱い方	3
1.4.4	メンバー情報の確認	3
1.4.5	MIDI 音源の選択	3
1.4.6	楽器やドラムの試聴	4
1.4.7	データ編集とマニュアル表示	4
1.4.8	フィンガー情報の確認	5
1.4.9	譜面モニタの使い方	5
1.4.10	最大化ボタンについて	7
1.4.11	演奏会場の設定	7
1.4.12	フォントの確認	7
1.4.13	MIDI ファイルのセーブ	8
1.4.14	キー操作による演奏制御	8
1.4.15	履歴について	9
1.4.16	コマンドラインによる起動	10
1.4.17	初期化ファイルでのユーザ指定	11
第 2 章	Muse コーディングの手引き	13
2.1	ピアノ独奏	13
2.1.1	まずはドレミを鳴らしてみよう	13
2.1.2	オクターブの指定はどうやるのかな	13
2.1.3	半音の指定はどうやるのかな	14
2.1.4	調性の指定方法	14
2.1.5	音の長さを指定しよう	15
2.1.6	付点の表現	15
2.1.7	タイの表現	15
2.1.8	コメントを使って可読性を高める	16
2.1.9	スタッカートの指定方法	16
2.1.10	省略音長について	17
2.1.11	連符の表現	17

2.1.12	テンポを変えてみる	18
2.1.13	装飾音の付け方	19
2.1.14	音の強さを指定しよう	19
2.1.15	音部記号について	20
2.1.16	和音とタイミングコントロール	20
2.1.17	コードによる伴奏	21
2.1.18	音符の連結	22
2.1.19	再現表記の活用	23
2.1.20	アクセントを付ける	24
2.1.21	グループ感を与える	24
2.1.22	アルペジオの記述方法	25
2.1.23	最小分解能と微分音長	26
2.1.24	エコーをかける	27
2.2	アンサンブル演奏	28
2.2.1	メンバーとフィンガーの概念	28
2.2.2	演奏タイミングを強制的に合わせる	29
2.2.3	繰返し記述と複写機構	31
2.2.4	パーカッションの演奏方法	31
2.2.5	フィンガー単位に指定する値	32
2.2.6	メンバー単位に指定する値	32
2.2.7	音色の指定	34
2.2.8	クレッシェンドとデクレッシェンド	35
2.2.9	ステレオ効果の利用	36
2.2.10	移調の仕方	36
2.2.11	楽器の音色をアレンジする	37
2.2.12	なめらかなグリッサンド	38
2.2.13	ダンパーペダルの操作	39
2.2.14	全体に影響を及ぼす値	39
2.2.15	リタルダンドとアツチェランド	40
2.2.16	ノンレガートな奏法の指定	41
2.2.17	強弱と音量の関係	41
2.2.18	コードの転回形	42
2.3	より高度な記述	44
2.3.1	コマンドによる制御	44
2.3.2	カラオケとして利用する	44
2.3.3	楽曲中の位置決めを容易にする	45
2.3.4	途中で強制的に停止させる	45
2.3.5	表示フォントを切替える	46
2.3.6	音声ファイルを再生させる	46
2.3.7	マクロで各種の繰返しに対応する	47
2.3.8	演奏パターンの部品化	49
2.3.9	スコア譜とパート譜の混在	52
2.3.10	メンバー色をカスタマイズする	52
2.3.11	ドラマーの転向について	53
2.3.12	ドラムロールの響きを向上させる	53

2.3.13	演奏会場を設定する	54
2.3.14	移調楽器をフィンガー毎に割り当てる	55
2.3.15	フィンガー属性を一気に指定する	56
2.3.16	音色バリエーションを指定する	57
2.3.17	波形を加工する	58
2.3.18	コントロールの指定	58
2.3.19	エクスクルーシブの指定	59
2.4	補足	61
2.4.1	コマンドの実行タイミングについて	61
2.4.2	メンバー色の表示優先度について	61
2.4.3	連符・和音・コードの入れ子関係	61
2.4.4	メンバー属性の競合について	62
2.4.5	エクスポート MIDI ファイルの定格性	62
第 3 章	Muse 文法リファレンス	64
3.1	全域属性	64
3.1.1	テンポ	64
3.1.2	タイミング	65
3.1.3	調性	65
3.2	メンバー属性	66
3.2.1	メンバー宣言	66
3.2.2	音色	66
3.2.3	音量	66
3.2.4	ステレオ	67
3.2.5	残響	67
3.2.6	揺らぎ	67
3.2.7	コーラス	67
3.2.8	ピッチ	68
3.2.9	ペダル	68
3.2.10	移調	68
3.2.11	コントロール	69
3.3	波形加工	69
3.3.1	出力変位特性	69
3.3.2	変調特性	69
3.3.3	周波数特性	70
3.4	フィンガー属性	70
3.4.1	フィンガー宣言	70
3.4.2	音名タイプ	70
3.4.3	強弱	71
3.4.4	アクセント	71
3.4.5	絶対オクターブ	71
3.4.6	相対オクターブ	71
3.4.7	音部記号	72
3.4.8	出音調整	72
3.4.9	止音調整	72

3.4.10	移調楽器	72
3.5	音符記述	73
3.5.1	音符	73
3.6	群記述	74
3.6.1	連符	74
3.6.2	和音	74
3.6.3	コード	74
3.7	補助記述	75
3.7.1	連結	75
3.7.2	再現	76
3.8	コマンド記述	76
3.8.1	コマンド	76
3.9	マクロ記述	77
3.9.1	繰返し	77
3.9.2	定義マクロ	77
3.9.3	展開マクロ	77
著作権などについて		78
収録楽曲一覧		78
開発後記		80

第 1 章

アプリケーション概説

1.1 Muse とは何か

Muse (ミュージズ) はテキストファイルで譜面を記述し、そのファイルを読込ませて音楽を演奏させるソフトウェアです。演奏中はリアルタイムかつカラフルに行われるピアノ自動演奏を楽しめます。作成したデータは、MIDI ファイルに出力可能です。100 を越える楽器やドラムの音色を試聴できるウィンドウを装備。更にバリエーションやドラムセットの組合せで音源の持つ音色をフルに引き出せます。残響や揺らぎ等の指定、波形特性の加工も可能です。指定したタイミングで文字列を表示するエリアもあり、歌詞を入力しておけばカラオケソフトとしても活用可能。曲の途中で WAVE や MP3 など音声を含むマルチメディア系のデータファイルを再生し、効果音を添えることもできます。入力したデータを疑似楽譜で表現し、音程やタイミングを確認する譜面モニターを実装。演奏に合わせ譜めくりも自動で行います。Windows95/98/Me/NT/2000/XP/Vista/Windows7 に対応しています。

1.2 インストールとアンインストール

必要なファイルは、Muse.exe と Readme.txt の 2 つだけです。適当なフォルダにコピーすればインストールは完了です。「マニュアル表示」のために上記 2 ファイルは同一フォルダに置いて下さい。

Vista 以降の OS で利用する場合、Program Files フォルダや My Document などアクセス権のあるフォルダにインストールすると Muse.ini が生成されません。アクセス権の縛りの無い階層下に、予め Muse 用のフォルダを作成し、そこにインストールする事をお勧めします。

添付ファイルの Clap.mp3 は必須ではありませんが、Readme.txt の曲例で使用しているため同一フォルダに配置することを推奨します。Sample*.mus のファイル群は文字通りサンプルです^[1]。どんなフォルダに置いて構いません。削除して頂いても結構です。

アンインストールは、インストールしたフォルダにあるファイルをごっそり消去すれば完璧です。レジストリは一切いじっていません。

Muse は MIDI 音源で駆動します。もし、お使いのパソコンに MIDI 音源が搭載されていない場合、残念ながら Muse は起動しません。起動したにもかかわらずうまく音が出ない場合は、メニューの「音源 (V)」で使用する音源デバイスを切替えてみて下さい (1.4.5 MIDI 音源の選択 (p.3) を参照のこと)。

1.3 Muse データの基本的な作成手順

Muse データは単純なテキストファイルです。このデータファイルは、メモ帳などを使って作成します。以下、Muse データを新規作成する場合の手順を示します (初心者の方のために、一般のテキストエディタを“メモ帳”と表現しています)。

1. メモ帳を起動し、Muse 文法に従ったテキストを入力して、「名前を付けて保存」をする
2. Muse の「ファイル」 「開く」にて、対象データを読込む

3. Muse の鍵盤をクリックして演奏
4. メモ帳でデータを修正し「上書き保存」を行う
5. Muse の「ファイル」「リロード」にて修正したデータを再度読み込む以降、3~5 の繰り返し

Muse データのファイルは ASCII か Shift_JIS コードで保存して下さい。

Muse データの文法に関しては 2 Muse コーディングの手引き (p.13) を参照して下さい。また本節での説明は基本的な流れだけですので、派生的な機能に関しては次節 1.4 機能概要 (p.2) をご覧下さい。

新規作成ではなく既存データを手直しする場合は、上記 2 の作業から始めることになります。その際、初回の 4 で Muse の「ファイル (F)」「データ編集 (D)」を選択すると、対象データが読み込まれたメモ帳が起動するので便利です。

手順 1 で「名前を付けて保存」する際、他のテキストデータと区別するために、拡張子を .mus にすることを推奨します。更に、この拡張子を Muse に関連付けすることで、エクスプローラから Muse データのダブルクリックするだけで、簡単に Muse を起動できるようになります^[2]。

1.4 機能概要

1.4.1 Muse ファイルのロード

「ファイル (F)」「開く (O)」の選択で、Muse ファイルを指定し読み込みます。Muse ファイルは譜面を文字で表現するテキストファイルです。そのファイルはメモ帳などのエディタで、あらかじめ作成しておく必要があります。また、その書式は次章で規定された文法に従っていないとなりません。なお、一般のテキストファイルと区別するためにファイルの拡張子を .mus にすることを推奨します。

ロード時に文法のチェックが行われ、エラーがあるとダイアログでその内容と Muse ファイル上の行番号を知らせます。うまくロードできると、ウィンドウのタイトル部に “ 00:00 ” の形式で演奏総時間が表示されます。エラーがあると “ x ” の表示となり、演奏に移れません。

データを作成している過程でロードとエディットを頻繁に繰り返すことを想定し、「ファイル (F)」「リロード (L)」の機能を設けました。このメニューは選んだだけで現在ロードされているファイルを再び読み込むので、カット&トライがより簡便に行えます。また「開く (O)」と異なり、シークバーを巻戻さないで、曲のある部分をデバッグする時も位置決めが容易です。

Muse のショートカットを作成することで、ロード用ダイアログの初期表示フォルダを制御できます。ショートカットのプロパティで「作業フォルダ」の欄に、表示させたいフォルダを指定して下さい。

1.4.2 演奏の仕方

データロード後、鍵盤部分をマウスクリックすることで演奏を制御します。演奏中は、メニュー右端の “Presented by ...” がグレーアウトします。

左クリック：演奏を開始します。演奏中ならば一時停止と再開を繰り返します。

右クリック：曲頭へ巻戻します。演奏中ならば演奏を停止し曲頭へ巻戻します。

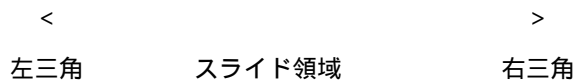
エクスプローラから Muse ファイルをドラッグ&ドロップすると、直ぐに演奏を開始します。既に何かの曲が演奏されている状態で新たにドロップした場合は、今の曲を停止し新たにドロップされた曲の演奏を開始します。

データ内に STOP コマンドがある場合は、文字列の表示色が赤くなり一時停止の状態になります。演奏の再開はマウスの左ボタンで行います。

「機能 (G)」「繰り返し演奏 (R)」の選択で現在のデータを繰り返し演奏させることができます。再度選択すると繰り返し演奏は解除されます。データ内に STOP コマンドがある場合は、STOP コマンド間に限定した範囲での繰り返しになります。

1.4.3 シークバーの扱い方

鍵盤上部のシークバーを動かして演奏開始位置を指定できます。また、スライド領域（下図）をクリックすることで、一気にその位置へつまみをシークさせることができます。



データ内に STOP あるいは MARK コマンドがある場合は、左右の三角をクリックすることで、そのコマンドのポイントに位置決めできます。また、テキストの表示領域をマウス右ボタンでクリックすると STOP や MARK の位置を示すゲージマークが ON/OFF します。更に、このゲージマークを表示した状態でテキスト表示領域をマウスの左ボタンでクリックすると、指定された位置におけるゲージ区間の演奏時間が表示されます。

シークバーのスライド領域（上図）をマウス右ボタンでクリックすると、リロード クリック位置にシーク 演奏開始の一連の処理が実行されます。この機構は曲を入力作成する際に、頻繁に繰返される操作手順を想定しています。

左三角（上図）をマウス右ボタンでクリックすると、リロードした後に曲頭より演奏を開始します。リロードのみを実行したい場合は、右三角（上図）をマウス右ボタンでクリックして下さい。メニューの「リロード(L)」と同じ処理となります。

1.4.4 メンバー情報の確認

「機能(G)」 「メンバー情報(M)」の選択で、鍵盤上に表示される各メンバーの色を確認できます。データがロードされている状態であれば、その演奏に参加しているメンバーに 印が付き、そのメンバーの使用楽器が表示されます。演奏途中で楽器の持替えが起ると、リアルタイムに表示が切替わります。更に、パリエーション指定されている楽器には“/数字”という形式で、楽器名にパリエーション番号が添えられます。

印の付いているメンバーをマウス左クリックすると、印が に変り、そのメンバの演奏だけを止めさせることができます。再度クリックすると に戻り演奏を再開します。また、マウスの右クリックで、すべての 印を一斉に切替えることができます。これらの機能は、Muse データ作成過程において各メンバーを分離してデバッグしたり、楽器の音色を微調するのに便利です。また、主旋律を止めて伴奏だけにすれば、カラオケソフトとしても活用できます。

メンバー毎の ON/OFF 機構は、キーボードからも行えます。それぞれのメンバーに該当する小文字のアルファベット (a~o, z) の押下で OFF になり、大文字のアルファベット (A~O, Z) で ON になります。全ての音を消す場合は小文字 x を、全ての音を出す場合は大文字 X を押下して下さい。

1.4.5 MIDI 音源の選択

「音源(V)」の選択で、パソコンに搭載されている MIDI 音源の一覧がプルダウンメニューで現れます。現在、Muse に割付けられている音源には黒丸 マークが付いています。これを切替えることで、使用デバイスを選択できます。さしずめ、演奏楽団を変更するといった所です。

演奏中に音源を切替え、更に間髪入れず再生を行うと、音源ドライバの解放が追従できず、雑音が入ったりハングしたりする場合があります。特に異なるソフト音源間の切替え時に発生しやすいので、切替え後の再生は 5 秒程度待ち、ドライバの解放を見計らってから行う方が無難です。

プルダウンメニューの一段目に表示される音源は“MIDI マッパー”です^[3]。この音源は、Windows で定義されている既定の MIDI デバイスで、Media Player や Web サイトなどで MIDI が再生されるときに使用される MIDI 音源です。実際の物理的な音源は、2 段目以降のどれかに相当しています。実際にどの音源に相当しているかは、コントロールパネル配下にあるオーディオ関係のデバイス指定欄にて確認して下さい^[4]。

1.4.6 楽器やドラムの試聴

「機能 (G)」 「楽器の試聴 (P)」および「ドラムの試聴 (Z)」の選択で、ボタンの沢山並んだダイアログが出現します。それらのボタンを押して音色を確認できます。

それぞれの試聴ダイアログには、[x0]の音名タイプボタンがあり、これを押下すると、音名の表示形式を切替えることができます。また、スクロールバーで音色エフェクト (残響・揺らぎ・コーラス) や音程・ピッチ変化を確認できます。音を止めるには、マウスの右ボタンを任意の位置でクリックします。「楽器の試聴」には、音を止めるための [] ボタンや、再開するための [] ボタンも付いています。

「楽器の試聴」「ドラムの試聴」共に、最後の試聴音に対応した文字列がクリップボードにセットされます。したがってテキストエディタ側で貼り付けをすることで、入力を簡便に行えます。

「楽器の試聴」ダイアログでは、選択された楽器番号が表示される欄に、上下方向のスピンボタンが付いています。このスピンボタンを押すことで、各楽器のバリエーション変化を確認することができます。ただし、音源によってはすべてのバリエーションをサポートしていません。

「ドラムの試聴」ダイアログの下辺に並んでいる選択ボタンで、ドラムセットを選ぶことができます。標準以外のセットを選ぶと、標準と異なる音色ボタンに黒丸 が付きます。また、Ethnic や SFX のドラムセットを選ぶと、ボタン名が効果音名称に切替ります^[5]。

「ドラムの試聴」の左下には、確認した音の音名が表示される欄があります。この欄の直上に添えられた数値は、MIDI で定義されたその音のノート番号が 10 進および 16 進で表示されます。X コマンド等を使用する場合にご活用下さい。

ダイアログ右下の [波形] ボタンにより、波形加工ダイアログが出現します。このダイアログは波形を加工することで、どのような音色の変化が起きるかを確認するためのものです。各特性パラメータをスクロールバーで調整すると、波形のイメージ図が変化しその傾向を目視で確認できます。また波形加工の効果は、強弱 v によっても変化するため、強弱のスクロールバーも付いています。

波形加工ダイアログのスクロールバーの位置決めは、矢印キーを利用して調節すると微調が容易です。また、波形イメージ図の領域を左クリックすると発音をし直し、任意の場所を右クリックすると音を止めます。音作りの試行錯誤の際に活用して下さい。更に、クリックした領域に対応する書式が、発音時にクリップボードにセットされます。なお“標準”の列に並んでいる [R][W][Q] の各ボタンは、それぞれの特性を初期状態に戻す機能を持っています。

各試聴ウィンドウは、あるタイミングでクリップボードに情報をコピーします。これにより、データ作成時にエディタ側で貼付けを行うことで簡便に入力できます。コピーされる情報は以下のようになっています。

- 楽器の試聴
発音時に、メンバー属性をコピーする。
- ドラムの試聴
発音時に、ドラム音程をコピーする。またドラムセットの選択時に、メンバー属性をコピーする。
- 波形加工
各特性エリアでの発音時に、各加工属性をコピーする。

1.4.7 データ編集とマニュアル表示

「ファイル (F)」 「データ編集 (D)」の選択で、テキストエディタが起動し、現在ロードされているデータが表示されます。起動するエディタは拡張子 .txt と関連付けられているアプリケーションです^[6]。

「ファイル (F)」 「マニュアル表示 (H)」選択で、Readme.txt が表示されます^[7]。

「機能 (G)」 「常に手前 (T)」の選択で、Muse のウィンドウ群を他ウィンドウに対して常に手前に表示させることができます。再度選択すると解除されます。ただし、「データ編集 (D)」や「マニュアル表示 (H)」で起動するテキストエディタは、他ウィンドウに属します。

1.4.8 フィンガー情報の確認

「機能 (G)」 「フィンガー情報 (F)」の選択で、現在ロードされているデータの最終的な属性値が表示されます。各フィンガーのオクターブ、強弱値などに関して、曲尾の段階で保持されている値を確認できます。

64 分音符以下の音長 (微分音長) を活用した際、演奏の位置ずれミスを起こしやすいため、各フィンガー行に音長の端数 (i0 ~ i59) を表示します。すべてのフィンガーが同一値であれば、演奏タイミングは一致しています。

ウィンドウの最右列にはメンバーの属性値も併記され、その内容はコンボリスト選択にて切替えることができます。更にウィンドウ上部左側には、最終時点のテンポ値 % も表示されます。

ウィンドウ下部には、リスト上で選択されているフィンガーの音名タイプ x やアクセント w 、出音・止音のタイミング p, q 、移調楽器 y に加え、そのフィンガーの属するメンバーの波形加工値や調性なども表示されます。

1 つのメンバーに複数のフィンガーが存在する際、その中の 1 つを絞り込んで確認したい場合があります。その様な時はフォーカス機構が便利です。ウィンドウ左上の FOCUS チェックボックスを ON にすると、現在選択されているフィンガーのみが演奏され、他のフィンガーは演奏しないようになります。ただし、フォーカスを当てたフィンガーが属するメンバー以外のメンバーは、通常通り全フィンガーが演奏されますので、それもオミットしたい場合はメンバー情報ダイアログの マークにて、個々に無演奏指定をして下さい。フォーカス機構が働いているメンバーはメンバー情報による ON/OFF 指定に抛らずフォーカスされたフィンガーが演奏されますので、全てのメンバーを OFF にしておくと、単一フィンガーのみを演奏させる事ができます。なお、フォーカス機構が働いているメンバーは記号が \blacksquare に変化しています。

フォーカス機構による演奏の絞込み状態は、譜面モニタや MIDI エクスポート、鍵盤演奏にも反映します。

ウィンドウのヘッダーやフッターをマウスでダブルクリックすると、リスト上にあるフィンガー行をすべて表示させる状態と、最小サイズ状態を交互に切り替える事ができます。また、フッターを右クリックした場合はリロードします。

キーボードからメンバー ID としての A~0, Z の文字を押下すると、該当行にフォーカスされます。連続入力で同一メンバー内をトグルシフトします。SHIFT キーの併用でトグルの方向を逆にできます。

1.4.9 譜面モニタの使い方

「機能 (G)」 「譜面モニタ (S)」の選択で、擬似的な譜面表現によるデータの確認ができます。また、演奏を開始すると“指揮棒カーソル”が表示され、演奏に追従して移動します。ただし、本譜面モニタはあくまでモニタであり、データ入力や編集はできません。現在のデータ状態を監視するのみです。

音符はメンバー色の付いた矩形で表示され、長さで音長を表します。各色の表示は「メンバー情報」ダイアログ上の ON/OFF に従います。シャープ・フラットは音符矩形から上下に伸びる矢印で表され、ダブルシャープやダブルフラットの場合は、矢じりが二重になります。高音域および低音域に引かれた水平の点線は、メインウィンドウの鍵盤範囲を表現しています。この範囲外の音程は鍵盤表示はされませんが、発音はします。

文字列表示タイプのコマンド (HEAD, TEXT, STOP, MARK) の内容は、それぞれの対応色でウィンドウ上部に表示されます。更に WAVE コマンドも深緑色で表示します^[8]。また、表示部をクリックすると HEAD 以外を ON/OFF することができます。

調性のシャープ・フラットは五線曲頭には表示されず、すべて各音符に展開し付加されます。また音符矩形の音程は、鍵盤での演奏音程と同一であり、移調属性 T などを処理した結果の実音で表示されます。この点において移調楽器の場合は、実際の譜面と譜面モニタとは音程が一致しませんのでご注意ください。どうしても譜面との比較をダイレクトに行いたい場合は、データ側を一時的に T0 に変更するなど工夫して対応して下さい。

拍子の分母・分子付近をクリックすると、各々の値を変えられます。音部記号の付近をクリックすると、ト音・ハ音の大譜表を八音記号の五線に切替えられます。これらの機能は、左右のマウスボタンで変化方向が異なります。ウィンドウ左部の上下隅にある *8va* をクリックすると、五線が 1 オクターブ上下にシフトします。再度クリックすると戻

ります。

音部記号の上方に表示されている <> 記号を左クリックするとデータ中の属性を譜面上部に表示する事ができます。表示属性は、<> 部分の右クリックで出現すポップアップで選択することができます。選択できる属性は、テンポ %、音量 V、ピッチ U、ペダル Y、強弱 v, w、グループ p, q です。

<> 内に表示される数値は、その時点の譜面モニタ先頭位置における値です。ただし遅延指定が存在する場合は、遅延終了時点の最終値を示します。

<> 内のパラメータ値の表示ルールには以下の例外があります。

- 強弱の遅延指定の場合は、最終値ではなく各音符の値が表示される。
- グループ属性の場合は、パラメータ値そのものが表示されない。

テンポ以外の属性はフィンガー情報ダイアログで選択されているメンバーあるいはフィンガーを対象とし、キーボードからメンバー ID としての A~0, Z の文字を押下すると切り替える事ができます。連続入力で同一メンバー内のフィンガーをトグルシフトします。SHIFT キーの併用でトグルの方向を逆にできます。

小節の途中から曲がはじまる弱起タイプの楽曲（アウフタクト）で小節線を合わせるには、小節番号付近の小節線をマウスでつまみ、横方向にドラッキングして下さい。その時点の拍単位で調整できます。この機構は変拍子などの楽曲に対して、局所的に小節線を合わせて拍ズレを確認するといった使い方にも活用できます。ただし、拍子自体を切替えると、この弱起シフトはクリアされます。小節線は小節番号の付近をクリックすることで、表示を ON/OFF できます^[9]。

小節番号付近の小節線をマウスで右クリックすると、その位置を小節番号 1 とするオフセットをかけられます。オフセットをかけると、小節番号 1 以前の小節には番号が表示されなくなります。再び小節番号 1 の小節線をマウスで右クリックするとオフセットが解除されます。

マウスによるグリップスクロールも可能です。マウスカーソルがノーマルになっている領域で、マウスを押下したまま左右にスライドします。マウスの移動に従い横スクロールを行うので、微妙な位置決めができます。一方、水平スクロールバーの左右の黒三角ボタンでは、1 小節単位のスクロールとなります。キーボードの左右矢印キーや、マウスホイールでもスクロールが可能です。

マウスカーソルがノーマルになっている領域でマウスを右クリックすると、その場でデータのロードを行い、更にクリック位置の付近から演奏を開始します。この機構は同じ位置で何度でもリロードをトライアルできるため、デバッグ作業が簡便になります。

ウィンドウの左下隅にある [P][M][nlbA] のボタンは各々以下の機能を提供します。

	キー操作
[P] 現在表示されている譜面先頭から演奏を開始します (Play)	Enter
[M] 指揮棒カーソル位置が譜面先頭になるよう譜めくりをします (Manual)	Tab
[A] 譜めくりを自動で行います (Auto)	@

音符を手掛りに位置決めするため、位置決めには多少の誤差が生じます。特に、長い休符が続く無音の領域では誤差が顕著になります。

SHIFT キーを押下しながら任意の音符矩形を左クリックすると、譜面モニタのウィンドウタイトルにテキストデータ上の行番号が表示されます。表示形式は以下の通りです。

(#A3 16 \$123)

γ. 発音実体の行番号 (マクロや再現表記などの場合)

β. 発音位置の行番号

α. フィンガー宣言

マクロや再現表記ではない通常の音符は、γ の行番号は表示されません。また定義部と展開部が同一行番号の場合も、γ は表示されません。

マウスクリック時点で、検索された行番号 β がクリップボードにコピーされるので、テキストエディタ側でペーストする事が可能です。なお、音符が重なっていることで複数音符が検索された場合は、譜面モニタ上で最上位に描画露出している音符矩形がクリップボードへのコピー対象となります。

ウィンドウのストレッチ（寸法変更）にて、常に五線譜の上下位置がウィンドウの中央に配置されるように調整されますが、ヘッダー部のダブルクリックでも、上下中央フィッティングがなされます。

1.4.10 最大化ボタンについて

Muse における最大化ボタンは、通常の Windows インターフェースと動作が異なり、メンバー情報・フィンガー情報・譜面モニタ の 3 つのサブウィンドウを一括して ON/OFF する機能として働きます。

1.4.11 演奏会場の設定

「機能 (G)」 「演奏会場の設定 (A)」の選択で、データ再生時の音場空間を調整することができます。調整は“残響設定”と“コーラス設定”の 2 系統のパラメータを組み合わせることで行います。残響設定のパラメータは 6 つ、コーラス設定のパラメータは 7 つあり、水平レバーを使用してその値を決めます。

各々のパラメータの意味は 2.3.13 演奏会場を設定する (p.54) を参照して下さい。

これらのパラメータの組合せには標準的なものが用意されており、それらはダイアログの左側に縦に並ぶ選択ボタンにより行います。残響・コーラスそれぞれに 8 個ずつの選択肢があります。自分なりの演奏会場を設定する際、まずこれらの標準パターンの組合せを選んだ後、水平レバーで微調していくと良いでしょう。なお残響設定とコーラス設定は、それぞれメンバー属性のエフェクトである残響 R とコーラス Q と強い相関があり、R や Q の値がゼロのデータでは音場空間の調整効果を出すことはできません。

ダイアログの上部には、設定した演奏会場を表現する ROOM コマンドが表示されます。右上の [COPY] ボタンを押下することで、クリップボードにコマンドがコピーされますので、データ作成時にエディタ側で貼付けを行うことで簡便に入力できます。なお、演奏会場の設定に関する効果は、使用される音源によってかなり差異がありますので、その時点の音源名称をコメントとして付帯させるようになっています。あくまでもコメントですから削除しても構いません。

このダイアログで設定した音場空間を、再生してるデータに直接反映させたい場合は、左上にある“Fix”をチェックして下さい。チェックをしていない時はダイアログの設定は無視され、ロードされているデータ内に記述されている ROOM コマンドの設定が有効になります。逆にチェックをしている場合は、ROOM コマンドの方が無視され、常にこのダイアログの設定が有効になります。Fix チェックをせず、データ上にも ROOM コマンドがない場合は、デフォルトの会場 (HALL-2・CHORUS-3) になります。

演奏の最中に、水平レバーを動かしたり、標準パターンを切替えたりしても、その時点では演奏に反映しません。反映は演奏再開の時点で行われます。ただし演奏中に上述の [COPY] ボタンを押下すると、Fix チェックの有無に関わらず、その時点の設定を演奏に反映させることができます。この [COPY] ボタンによる即時反映の機構は「楽器の試聴」や「ドラムの試聴」の際にも有効です。

Fix チェックがなされていない場合、[COPY] ボタンによる会場切換えの効果は、演奏を一度止めると無効になります。ただし「試聴」系で使用した場合は「試聴ダイアログ」を閉じない限り効果し続けます。

1.4.12 フォントの確認

「機能 (G)」 「フォントの確認 (N)」の選択で、お使いのパソコンでどのようなフォントが利用可能かを確認することができます。このダイアログの上部には Muse のテキスト表示領域での表示イメージが描画されます。フォントの形状だけでなく、修飾 (太字・斜体) やコマンド色 (HEAD・TEXT・MARK・STOP) なども切り換えて確認することができます。

フォントリストの行頭には、英文フォントなら [ENG] 和文フォントなら [JPN] という形式でフォントセットの

種類が表示されます。また、プロポーショナルなフォントの場合は、その横に P の文字が添えられます。なお、リストはこれらの種別群でソーティングされます。

表示する文字列はダイアログの 2 段目の欄の内容を変更することで変えることができます。また、右下の [COPY] ボタンを押下すると、現在のフォント状態を指定する Muse コマンドをクリップボードに出力します。

フォント選択の際には、リストボックスにフォーカスを与え、キーボードの上下矢印 [↑] [↓] キーで次々とフォントを切り換えていくと便利です。

1.4.13 MIDI ファイルのセーブ

「ファイル (F)」 「エクスポート (E)」の選択で、現在ロードされているデータを MIDI ファイルとして出力しませんが^[10]。「メンバー情報」ダイアログで、印刷状態にしたメンバーは出力しないので、メンバー毎の MIDI ファイルを作成することも可能です。ただし、*TEXT などコマンドに対応するメタイベントは、すべて出力されます。また「演奏会場の設定」で Fix チェックがなされていてもダイアログ上の設定は出力されません。ROOM コマンドでの設定を出力対象とします。

出力される MIDI ファイルの諸元は以下の通りです。

形式	SMF フォーマット 0
初期化コマンド	GS リセット ^[11]
拍子メタイベント	4/4
TPQN	120
セットアップ小節テンポ	800
ソングボディテンポ	125

Muse には小節の概念がないため、拍子のメタイベントは固定です。また、テンポを定値出力し、デルタタイム数値演算で高精度の演奏速度制御を行います。Muse の音量指定 V はエクスプレッションに反映し、対してマスタボリュームはデフォルト値を保持します。ランニングステータスは使用しません。

Instrument Name メタイベントに “Generated by MUSE” と刻印させて頂きました。ただし、楽曲の著作権はその作曲者が、歌詞の著作権はその作詞者が保有します。そして MIDI データの制作者は、演奏表現の創出者として著作権隣接権が与えられます。道具 (Muse) を開発した私 (加藤) には、制作された MIDI データにおける何の著作権も存在しません。

1.4.14 キー操作による演奏制御

演奏の制御はマウスによる方法以外に、以下のキー操作でも行えます。

スペース	演奏の開始/一時停止 (鍵盤上のマウス左ボタンと等価)
BS キー	演奏の停止 曲頭巻戻 (" 右ボタンと等価)
矢印 [↑] [↓]	演奏位置のシーク
矢印 [←]	リロード
矢印 [→]	リロード後、曲頭から演奏再開

矢印 [↑] [↓] でシークを行う際、SHIFT キーを併用すると、シーク速度が上がります。通常で端から端まで 30 秒かかる所を、2 秒で到達します。離れた位置へ素早くシークしたい場合に利用して下さい。また、データ内に MARK や STOP のコマンドがある場合、コントロールキーとの併用でコマンドのポイントに位置決めできます。

矢印 [←] [→] でリロードを行う際、文法エラーが存在すると、その内容がクリップボードにもコピーされます。

譜面モニタやフィンガー情報にて矢印キーを押下した場合は、それぞれのウィンドウでのスクロールとして機能し、リロードやシークは行いません。

メニュー上の全ての機能は、コントロールキーとメニュー英字キーを、同時に押下することで発行することもできます。

1.4.15 履歴について

履歴メニューは最新のアクセス曲を数曲分（デフォルト 8 曲）記憶し、次回に同じ曲をアクセスすることを簡便にします。

履歴ファイル Muse.log は、実行ファイル Muse.exe と同一フォルダに自動生成されるファイルであり通常はユーザが編集する必要はありませんが、単純なテキストファイルですので気軽に閲覧できます^[12]。書式は以下の通りです。

```
2006/12/06(23:40:20) 169> D:\MuseData\Sample2.mus
```

```
日付      時刻      カウンタ  Muse データへのパス
```

履歴曲は最新アクセス曲で順次更新されますが、もしお気に入りの曲があり、固定的に記憶させておきたい場合は、直接履歴ファイルを編集する事でそれが可能となります^[13]。履歴と言うよりもむしろアルバムと言った方が相応しい機構です。履歴ファイル編集は任意のエディタ（メモ帳等）で行って下さい。履歴ファイル記述ミスがある場合はエラーは発生せず、その行は無視されます。

履歴ファイルをアルバム化する方法は至って簡単です。自動的に生成された履歴行の要所々々に、アスタリスク * で始まる行を挿入し、そこにアルバムのタイトルを記述するだけです。曲の順番も自由に移動できます。履歴メニューでは、タイトルを階層名としたグループが形成され、その配下のログ曲は更新対象にならず固定的に保持されます。アスタリスクのみでタイトル無いグループは、通常の履歴ログ更新対象の集団となります。以下に指定例を示します。

```
*ポップス                                タイトル行
2007/05/03(09:45:16) 61> D:\MuseData\Sample01.mus
2003/09/15(15:35:59) 152> D:\MuseData\Sample02.mus   ログ行
2005/07/10(03:28:42) 9> D:\MuseData\Sample03.mus
*ゲーム音楽
2007/05/04(13:18:09) 29> D:\MuseData\Sample04.mus
2007/04/18(23:42:18) 54> D:\MuseData\Sample05.mus
*クラシック
2003/12/06(08:08:32) 219> D:\MuseData\Sample06.mus
2007/09/03(11:28:11) 7> D:\MuseData\Sample07.mus
2006/11/12(20:58:29) 32> D:\MuseData\Sample08.mus
*
2007/05/13(21:38:22) 88> D:\MuseData\Sample09.mus
2007/05/13(03:48:15) 90> D:\MuseData\Sample10.mus
```

タイトル行をひとたび設定すると、上記のようにエディタで編集することなく、Muse 履歴メニュー上のマウス操作で、手軽にメンテナンスする事ができるようになります。ただし、可能な操作は登録と除去の 2 つだけです。

- アルバムへ曲を登録
タイトルの部分でマウスを右クリックすると、そのタイトルの階層下に現在ロードされている曲が登録されます。
- アルバムから曲を除去
登録曲の部分でマウスを右クリックすると、その曲を除去します。

階層を多段にしたい場合は、履歴ファイルの編集で実現することができます。例えば以下のような階層を履歴ファイルで定義する場合について解説します。黒丸 () がタイトル行、二重丸 () がログ行とします。音楽とは関係ない、お酒の階層例で恐縮です (苦笑)。

```
醸造酒
  果実系
```

```

        ワイン.mus
        シードル.mus
    穀物系
        日本酒.mus
        ビール.mus

    蒸留酒
        焼酎
            麦焼酎.mus
            米焼酎.mus
            いも焼酎.mus
        ブランデー.mus
        ウイスキー.mus
        泡盛.mus

    混成酒
        カクテル.mus
        まむし酒.mus
        梅酒.mus

```

上記のような多段の階層を実現したい場合、タイトル行にアスタリスクを連ねる事で定義します^[14]。例では、お酒を醸造酒・蒸留酒・混成酒の3つに大分類しています(*1つ)。さらに醸造酒を果実系と穀物系に分類しています(*2つ)。

```

*醸造酒
**果実系
2007/05/03(09:45:16)    19> D:\MuseData\ワイン.mus
2003/09/15(15:35:59)    38> D:\MuseData\シードル.mus
**穀物系
2005/07/10(03:28:42)    77> D:\MuseData\日本酒.mus
2007/05/04(13:18:09)    208> D:\MuseData\ビール.mus
*蒸留酒
**焼酎
2007/04/18(23:42:18)    23> D:\MuseData\麦焼酎.mus
2003/12/06(08:08:32)    19> D:\MuseData\米焼酎.mus
2007/09/03(11:28:11)    36> D:\MuseData\いも焼酎.mus
**
2006/11/12(20:58:29)    112> D:\MuseData\ブランデー.mus
2007/05/13(21:38:22)    79> D:\MuseData\ウイスキー.mus
2004/02/06(18:08:22)    48> D:\MuseData\泡盛.mus
*混成酒
2007/05/13(03:48:15)    53> D:\MuseData\カクテル.mus
2008/03/31(21:02:18)    30> D:\MuseData\まむし酒.mus
2008/03/31(21:02:18)    18> D:\MuseData\梅酒.mus

```

1.4.16 コマンドラインによる起動

Windows であっても、以下の局面でコマンドラインを活用することができます。

- ショートカットのプロパティにおける“リンク先”欄の記述
- “アプリケーションの関連付け”による定義
- [スタート] [ファイル名を指定して実行]による起動
- MS-DOS プロンプトからの起動

コマンドラインによる起動では、以下のような引数を与えることが可能です。データパスが指定されると、起動時にそのデータが読み込まれます。

muse [データパス] [オプション]

データパスにスペースが含まれる場合は、[データパス] 全体をダブルコーテーションで括弧して下さい。なお、データパスとオプションの順番は入替えても問題ありません。

オプションは以下の指定を組み合わせることができます。

- *i 最小化されたウィンドウで起動します。
- *x Muse の多重起動を抑制します。
- *p 起動直後に演奏を開始します (データパスが存在する時にのみ有効)。
- *e 演奏が終了すると Muse 自体が終了します。
- *m 起動時に「メンバー情報」を表示します。
- *f 起動時に「フィンガー情報」を表示します。
- *s 起動時に「譜面モニタ」を表示します。
- *t 起動時点で「常に手前 (T)」を ON にします。
- *r 起動時点で「繰返し演奏 (R)」を ON にします。
- *a 初期起動時の「譜めくりモード」を自動 [A] にします。
- *o MIDI ファイルを出力し演奏せずに終了します (MML コンパイラ相当)。

*o オプションはノンウェイトで実行されるので、バッチファイルなどで連続してコンパイルさせたい場合は、以下のように start コマンドにて、1 ファイルずつウェイトを取って実行した方が無難です。

```
start /w muse ファイル名.mus *o
```

1.4.17 初期化ファイルでのユーザ指定

初期化ファイル Muse.ini は、実行ファイル Muse.exe と同一フォルダに自動生成されるテキストファイルであり通常はユーザが編集する必要はありません。しかし、以下の定義行 ([USR] ブロック) に関しては、積極的にユーザが指定する事が可能であり、Muse の動作を制御することができます^[15]。

```
----- [USR]
#ED = C:\Program Files\notepad.exe      データ編集で使用するエディタ
#EP = /a /t /m                          エディタの起動パラメータ
#C0 = C:\WINDOWS\Cursors\harrow.cur     鍵盤と文字エリア上のカーソル形状
#C1 = C:\WINDOWS\Cursors\coin.ani      シークバー上のカーソル形状
LGM = 32      履歴メニューにおける表示曲数
LGP = 1      履歴曲選択時に演奏開始      ( 0 : しない / 1 : する )
DWT = 1      ウィンドウタイトルにフォルダ名表示 ( 0 : しない / 1 : する )
```

注

[1] すべての Sample 曲および本書に収録している楽曲は、日本音楽著作権協会の作品検索で著作権が消滅していることを確認しております。詳細は著作権などについて (p.78) をご覧下さい。

[2] 関連付けの方法 (特に必要が無ければ、実施しなくとも構いません)

- (1) エクスプローラにて、任意の Muse データを右クリックする。
- (2) 出てきたポップアップから「プログラムから開く (H)」を選択する。
- (3) 「プログラムの選択 (C) ...」をクリックする。
- (4) 表示されたウィンドウ上の [参照 (B)] ボタンを押す。
- (5) ファイル選択のダイアログにて、インストールした Muse.exe を選ぶ。
- (6) 「この種類のファイルを開くときは、いつも...」をチェックしておく。
- (7) [OK] ボタンを押下する。

[3] Windows Vista では MIDI マッパーが撤去されていますが、Muse では過去 OS との互換性のために敢えて残してあります。

[4] 具体的にコントロールパネル下のどのメニューがそれに相当するかは、Windows の OS バージョンにより様々なため、残念ながらこの場で

は丁寧なご紹介ができません。お手元の Windows ヘルプなどでお調べ下さい

- [5] MIDI 音源によっては、Ethnic や SFX のドラム音色が使用できない機種があります。この場合、直前に選んでいたドラムセットが維持され、結果としてボタンフェースと異なる音になります。また、o1 列や o6 列の一部のドラム音が鳴らない音源もあります。
- [6] テキストファイル (.txt) の関連づけにおいて、そのテキストエディタのファイルパスにスペースが含まれる場合は、ダブルコーテーションでくくっておく必要があります。
- [7] 「マニュアル表示」の機能を使用するためには Muse.exe と同一のフォルダに Readme.txt を配置する必要があります。
- [8] この文字表示には、FONT コマンドの効果は与えられません。
- [9] Muse には小節の概念がないため、ここでの小節線は音符の位置を把握するための目盛と考えて下さい。いわばグラフ用紙の方眼線のようなものです。
- [10] セーブ時、「保存 (MIDI File)」ダイアログの最下段にある「MIDI ファイル容量を抑えるため、テキスト系コマンドのデータを出力しない」をチェックすることで、通常の MIDI プレーヤでは無効となる Lyric イベント等の出力を抑止することが可能です。
- [11] 初期化は GS リセットですが、ほとんどの XG 音源において TG300B モードが存在するため、現在この仕様が最も汎用性の高い初期化方式です。ドラムマップやバンクセレクト、波形加工の指定なども、多くの音源で再現させることができます。
- [12] 履歴として保持する最大曲数などを、初期化ファイル Muse.ini で制御することができます。詳細は 1.4.17 初期化ファイルでのユーザ指定 (p.11) を参照して下さい。
- [13] 履歴ファイルは、Muse を終了させるタイミングで反映出力されます。Muse を起動しながら履歴ファイルを編集する際、Muse を終了してから最終的な結果をセーブしないと、折角の編集作業が徒労に終わる場合があるため、留意が必要です。なお、ログ行の日付・日時・カウンタの値は単なる参考データですので、編集の際にその論理性を意識する必要はありません。
- [14] 正確には、アスタリスクの個数が階層レベル値に対応している訳ではなく、個数の増減傾向で階層の上昇・下降を決めています。例えば、* を 1 つ付けたタイトル行の直下に、* を 3 つ付けたタイトル行を記述しても、階層レベルは 1 つ下がるだけです。Muse 終了時に自動的に履歴ファイルセーブが起こりますが、その際アスタリスク個数は正規化されます。
- [15] 初期化ファイルは、Muse を終了させるタイミングで出力されます。Muse を起動しながら初期化ファイルを編集する際、Muse を終了してから最終的な結果をセーブしないと、折角の編集作業が水泡に帰す場合があるため注意して下さい。なお、初期化ファイルの反映には Muse の再起動が必要です。

第 2 章

Muse コーディングの手引き

2.1 ピアノ独奏

2.1.1 まずはドレミを鳴らしてみよう

楽典における音名は以下のように 4 系統あります。

日本	ハ	ニ	ホ	ヘ	ト	イ	ロ
英米	C	D	E	F	G	A	B
独	C	D	E	F	G	A	H
伊	Do	Re	Mi	Fa	Sol	La	Si

古代旋法の名残で H である。

Muse はこの内、日本の小学生に馴染の深いイタリア系と、音楽的素養をお持ちの方向けに英米系、更にグレードの高い独系の 3 種類の記譜をサポートしています。デフォルトでは、イタリア語になっています。つまり“ドレミファソラシ”です。これらの各音を d, r, m, f, s, l, c という小文字のアルファベットに対応させています。これを並べれば音が出ます。休符も一種の音名ととらえ、アンダーバー _ で表現します。各音の間のスペースはなくとも構いません。

d r m f s l c _



英米系における対応文字は c, d, e, f, g, a, b です。独系は英米系に対し、最後の b が h となり、b は h の半音下がった音として解釈されます。これら音名の切り替えは、記譜する前に x 指定を与えることで行います。x0 がイタリア系、x1 が英米系、x2 が独系です。

```
x0 d r m f s l c _
x1 c d e f g a b _
x2 c d e f g a h _
```

実は Muse の音名は“音階”の名称というよりは、五線紙上の“位置”の名称と定義する方がふさわしいのです(2.1.15 音部記号について(p.20)参照)。ですから、まさに日本人の多くが愛用している“ドレミ”の概念に近いと言えます。以降、本マニュアルではイタリア系の音名で説明を進めます。

2.1.2 オクターブの指定はどうやるのかな

オクターブの指定は、o-, o0, o1, o2, o3, o4, o5, o6, o7, o8, o9 で行います。添えられた数字が大きくなる程高いオクターブを表します。基本のドは o4 に設定されています。この他にも、現在のオクターブを相対的に上下させる <, > があります。

o4 d m s o5 d m s o6 d
o4 d m s < d m s < d



相対オクターブ指定のみで記述しておく、出だしの絶対オクターブの指定を切替えるだけで、全体のオクターブをシフトできるので便利です。

鍵盤上に表示される音域は、o0 f から o8 m までですが、それを越える音域も発音は可能であり、譜面モニタには音符が表示されます。なお実際の発音範囲は、o- d から o9 s までです。

2.1.3 半音の指定はどうやるのかな

シャープは音名の後ろに + を添え、フラットは音名の後ろに - を添えます。

d d+ r r+ m f f+ s s+ l l+ c < d
d > c c- l l- s s- f m m- r r- d



また、ダブルシャープ、ダブルフラットを指定する場合は、それぞれ ++, -- と 2 つの記号を続けて音名の後ろに添えて下さい。

d++ m--



なお、Muse 記法は小節の概念がないため、臨時半音記号が小節内で効果があるという楽譜の一般則は当てはまりません。必要な箇所には個々に付けて下さい。ただし、次の節で紹介する調性の指定は可能です。

ドイツ音名の場合、Bes (ベス) は b- と記述します。

2.1.4 調性の指定方法

調性が決まっている場合、いちいち各音に +, - を添えるのが面倒なのは譜面と同じです。このような場合は \ の後に +, - を必要な数だけ添えて調を指定できます。以下はイ長調の例です。この例では降下する時にイ短調になるようにナチュラルにしてみました。ナチュラルの指定は音名の後ろに = を添えて指定します。当然のことながら、臨時記号として添えた +, -, ++, --, = は調性より優先します。

\+++
l c < d r m f s l
l s= f= m r d= > c l



調の変更は任意のタイミングで行うことができます。八長調に戻す時は \= という記述をします。

ドイツ音名の場合、音名 b は調性の影響を受けません。また、音名 h はフラット系の調性の影響を受けません。

乗算記号と加算記号は混在して記述できます。この場合、乗算記号が優先して演算されます。なお、乗算記号は連続して記述することはできません。

```
d1^2^4^8^3      d1^2^2^2^2^8^8^8 (乗算記号が優先する例)
d2^4^3          NG          (乗算記号は連続記述不可)
```

2.1.8 コメントを使って可読性を高める

音程と音長が組み合さると、音楽の様相を成してきます。1つ比較的長めの旋律を書いてみましょう。本格的に曲を記述する場合、可読性のあるコーディングを意識する必要があります。Muse は可読性を高めるため、次の様なコメント記述を準備しています。

- ライン・コメント
セミコロン ; 以降改行まではコメントとして無視する。
- ブロック・コメント
感嘆符 ! で囲まれた部分はコメントと見なす。8 個までの感嘆符を列挙することで、その数で対を成した領域指定ができる。後部の感嘆符がない場合は、前部からファイル末尾までをすべからく無視する。
- キャンセル文字
スペース、タブ、改行、全角文字、半角カタカナ、半角縦棒 | は無視する。

線路は続くよどこまでも (アメリカ民謡)

```
\+
| s4.r8s8.r16s8.116 | c2s2 | <d4.d8>s4l4 | c2._4 |
| s4.r8s8.r16s8.116 | c2s2 | l4.l8l4c4 | l2._4 |
| l4.l8s+8.l16c8.116 | s2r2 | <d4d4>s4l4 | c2._4 |
| m4.f8s8.f16s8.m16 | r4.r8s2 | c4.<d8>c4l4 | s2._4 |
```

特に、半角の縦棒 | や全角の縦棒 　 は小節の区切りとしてお勧めです。他にも、= < > - :

† ‡ ¶ などなど、全角文字の豊富な表現力を、すべてコメントとして使用できます。

セミコロンは、ある行をちょっとコメントアウトしたい時や、行の末尾にコメントを付けたい時などに便利です。

感嘆符は、開始側だけ記述すれば、データの途中から最後までを全てコメント化することができるので、ある程度書き進んだデータの途中部分をデバッグする時に便利です。また、このブロックコメント ! は、その感嘆符の数で対応する領域をコメント化します。例えば以下の様な場合、領域 β , δ は ! で囲まれているので、当然コメントですが、 α , γ も、全体が !!! で囲まれているためコメントとなります。

```
!!!
 $\alpha$  !  $\beta$  !  $\gamma$  !  $\delta$  !
!!!
```

このように Muse のブロックコメントは感嘆符の数により対を成し、入れ子を構成することが可能です。ただし、感嘆符の列挙記述の上限数は 8 個 (!!!!!!!) までです。

2.1.9 スタッカート指定方法

スタッカートは音長数字や付点の後にスラッシュ / を添えることで表現します。スラッシュは最大 4 つまで付けることができ、その数でスタッカートの強さを表します。

```
; ベートーベン 交響曲第 6 番「田園」
\~
%80 ; 2.1.12 テンポを変えてみる (p.18) を参照のこと
|_8l8c8<r8 |d8>c16l16s8 d8 |f8s8l8c16l16|s2
```

```
|_8l8c8<r8///|d8>c16l16s8////d8/////|f8s8l8c16l16|s2
```

スタッカートの目安は以下の通りです。

```
/      メゾスタッカート
//     スタッカート
///    スタッカティシモ
////   スタッカティシシモ
```

スラッシュの数と強さの定量的な関係は、2.2.16 ノンレガートな奏法の指定 (p.41) あるいは 3.5.1 音符 (p.73) を参照して下さい。

なお、加算記号で音長を連ねた場合は、最後の音長のみスタッカートを記述することができます。ただしスタッカートの効果は、加算記号などを含めたその音の全体の音長に対して与えられます。最後の音長値のみに掛かるわけではありません。

2.1.10 省略音長について

“省略音長”とは、音長部分の指定が省略された場合、直前の音符や休符で指定した音長が採用されるという機構です。音長のvari目だけを意識すれば良いので、記述量がぐっと減ります。

アルプス一万尺 (アメリカ民謡)

\+

s8 s8 l8 c8 s8 c8 l8 r8 s8 s8 l8 c8 s4 f8 r8 ; 省略音長を使用しない例

s8 s l c <d> c l s f r m f s4. ; 省略音長を使用した例

省略音長は音長数字がない部分に、直前の音符の音長がセットされます。加算記号を使いつつ、局所々々で音長を省略すると、その各部分に省略音長がセットされます。そして結果として生成された全体の音長が新たな省略音長として次の音に継承されていきます。

前音の4分音符が省略位置に入る

```
d4 m ^8^ s      d4 m4^8^4 s4^8^4
```

前音の音長全体が省略位置に入る

省略音長の対象には、スタッカートの指定も含まれます。次にスタッカートを明示しない限り、直前のスタッカートが引継がれます。省略音長からスタッカートをクリアしたい場合は、スタッカートのない音長数字を明示して下さい。

スタッカートのみ切替え ここでスタッカートをクリア

```
d8/   r      m      f///   s8   c   ; 省略音長を使用した場合
```

```
d8/   r8/   m8/   f8///   s8   c8 ; 省略せずに記述した場合
```

2.1.11 連符の表現

連符は、丸括弧 () で音名をくくり、閉じ括弧側に音長を添えることで表現します。添えた音長内で括弧内の音がすべて納まるように各音の音長が配分されます。8分音符内に納める3連符は次のようになります。

```
( d r m )8
```

なお連符における音長も省略可能であり、省略音長の概念が適用できます。

ヴェルディ「アイーダ行進曲」

```
\-
d4 |f2. (sds)4 | 1 1 1 (lcf) | 14. s8 f4 _ |
|s8 1 18. s16 f8 _ s8. 116 | 14 s8. 116 14 f8. s16 | s2.
```

この連符記法は括弧内に入る音符の数で何連符か判断します。また括弧末尾に付ける数値で音長を自由に指定できますから、トリルやグリッサンドなどの奏法記述に利用することができます。

```
( drdrdrdrdrdr )2 ; トリル
( mff+ss+ll+c<dd+rr+m> )2 ; グリッサンド
```

トリルのような繰返しパターンは 2.2.3 繰返し記述と複写機構 (p.31) が有効です。繰返し数でトリルの速度を制御できるので調整が簡便になります。

```
( fsfsfsfsfsfs )2 ( {fs}6 )2
```

連符内の音長指定はその配分比率を決定するために存在するので、直接実際の音長を表現しませんが、連符内の不均一な音長記述や、装飾音記述が可能となります。

```
o5 d4 (ddd) d (d8d8.d16)
```



連符においての省略音長は、まず閉じ括弧に添えられた音長が効果しますので、スタッカートで連符を演奏したい場合は、最後の音長にスタッカートの指定をすれば、全体に渡ってスタッカートが効きます。

```
( d r m f s )2//
```

中身のない連符は、直前に記述した連符の再現を意味します。この記法を利用すると、同一のアルペジオパターンが繰り返される演奏を簡素に表記できます。

```
( ds<dmmfmd> )2 ( ds<dmmfmd> ) ( ds<dmmfmd> ) ( ds<dmmfmd> )
```

```
( ds<dmmfmd> )2 ( ) ( ) ( )
```

連符の中に連符が入っているような譜面の記述方法に関しては、2.4.3 連符・和音・コードの入れ子関係 (p.61) を参照して下さい。

2.1.12 テンポを変えてみる

テンポの指定は%に続く 1~999 の範囲の数値で行います。この数値は 1 分間あたりの 4 分音符の拍数を表します。テンポ指定は任意のタイミングで何度でも変更できます。

```
おお牧場はみどり (チェコスロバキア民謡)
\+
%120
o4 r4 >c8.< d16 r4 c s m r2
%80
o2 c8 c4 <d8> c4 1 18 14 c8 1 s c4
%300
o5 r4 >c8.< d16 r4 c s m r2 c8 c4 <d8 r4 d> m f s _
```

2.1.13 装飾音の付け方

装飾音がある場合、それが付いている音符を装飾音の音長分だけ短くし、そこに装飾音を埋込んで、全体として拍が合うようにする必要があります。このような局面の記述をする場合には音長を少しだけ短くする指定があると便利です。この指定方法はチルダ ` を用いて行います。この指定記号を音長の“減算記号”と呼びます。例えば、4分音符の長さを16分音符だけ短くするには次のように指定します。

```
d4~16
```

装飾音符をつけた例を示します。

サラサーテ「チゴインエルワイゼン」

```
%60 \---
```

```
o5 > s8 < d8 r8 m2 r4 (drmr)16 d4~16 > c=16 < d2 _16
```

加算記号 ` を音長のプラス記号とすれば、減算記号 ` は音長のマイナス記号と例えることができます。よって、同じ結果になる表現方法が存在します。

```
d4^8^16^32 = d2~32
```

減算記号 ` 指定は、いくつでもつなげられる性質や省略音長に対するルール等、すべて加算記号 ` の指定と同等です。また、両者を混在して記述することもでき、結果として3つの演算記号（加算・減算・乗算）は数式風に記述できます。

```
d2^4^8^3~16      2分音符 + 4分音符 + (8分音符 × 3) - 16分音符
```

なお、減算記号における音長も省略可能であり、省略音長の概念が適用できます。例えば以下の様な前打音装飾であれば、うまい具合に装飾分の音長が継承されます。

```
(slcl)16 s8.~ ; 省略音長を利用した記述
```

```
(slcl)16 s8.~16 ; 実際に演奏される結果
```

2.1.14 音の強さを指定しよう

音の強弱は `v` に続く 0~127 の範囲の数値で行います。この数値は数が大きいほど強くなり、0 で無音状態となります。

スメタナ 交響詩「我が祖国」より“モルダウ”

```
%100 \+
```

```
v90 >c8< m4 f8 s4 l8c4c8c4. <d4. d> c4.^4
```

```
v50 >c8< v80 m4 v60 f8 v90 s4 v60 l8c4c8c4. v120 <d4. d> v70 c4.^4
```

また `v` の後に符号 `+`、`-` 付の数値を指定すると、現在の強弱値を相対的に変化させることができません^[3]。相対指定で記述しておく、出だしの絶対指定を切替えるだけで、強弱の流れを保ったまま、全体の強さをシフトできるので便利です。

```
v30 d4 v+30 m4 v+30 s4 v+30 <d4> v-30 s4 v-30 m4 v-30 d4
```


2.1.15 音部記号について

音部記号には、一般に馴染み深いト音記号やヘ音記号の他に八音記号というものもあり、しかも八音記号はソプラノ記号からバリトン記号まで5種類も存在します。結局、音部記号は合計7つの種類があることとなります。馴染みの薄い音部記号でかかれた譜面を読みとるのは大変労力が要ります。

そこで音部記号?を導入しました。?に続く0~6の数字で7種の音部記号を表します。この7つの分類に関しては3.4.7 音部記号(p.72)を参照して下さい。この記号は、譜面上の音部記号を指定することで、あたかもその譜面がト音記号で書かれたものとしてd r mの入力ができるというもので、まさに譜面を写し取る記法としては極め付け、しかし音楽的には邪道な記法です^[4]。ここまでくると、Museにおけるd r mの記号は、“音程”ではなく、五線上の音符の“位置”を表す記号であると解釈してもらった方が良くも知れません。以下の例は、ヘ音記号で書かれた譜面をト音記号で読み取り記述したものです。

シューベルト「未完成交響曲」

```
\++ ?6 s2. l2c4 s2. f4rm >c2l4< r2.^2.^4 _2
```



2.1.16 和音とタイミングコントロール

いよいよ複数の音を同時に鳴らす節にたどり着きました。あるタイミングで同時に音を鳴らすには、その音を鍵括弧 [] でくるだけです。この鍵括弧内にはいくつでも音を指定でき、すべての音が一斉に鳴り出します。

```
[ d4 m4 s4 ]
```



また、それぞれの音に音長を指定できるので、直ぐに終る音から最後まで残る音まで混在させることができます。

```
[ d2 m2 s2 ] _2 [ d2 f4. l4. ] _2 [ >c2< r2. s1 ] _2
```

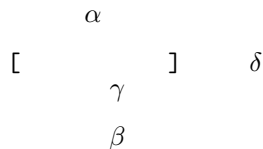
さて、今までの節ではある音を出すタイミングは、その1つ前の音が鳴り終わってからという暗黙の了解がありました。しかしこの和音においては、それぞれの音の鳴り終わるタイミングを変えることができるため、次の音が鳴り始めるタイミングをどこかで指定する必要があります。その指定は、閉じ鍵括弧に音長を添えることで行います。つまり鍵括弧に添えた音長が、和音全体の占有する時間であるというわけです。よって、この“占有音長”よりも長い音長の音が和音内に記述されていれば、それが鳴りやむ前に次の音が鳴り始めることとなります。

```
[ d2 m2 s2 ]4 < d8 r8 m8 _8
```

和音における省略音長は、まず閉じ括弧に添えられた占有音長に伝わり、和音の先頭に戻って、順次伝播していきます。一般に和音は、和音を構成するすべての音の長さ、占有音長が一致しているため、省略音長を有効に利用できます。

```
[ d2. m2. s2. ]2. [ d m s ]2.
```

和音における省略音長の継承プロセスは多少複雑なため、以下に図解しておきます。



- α. 占有音長が省略されていると、直前の音長値が採用される。
- β. 和音の初音が省略されていると、占有音長の値が採用される。
- γ. 和音内の省略音長値の継承は、和音内に閉じている。和音外へ継承されない。
- δ. 次の音への省略音長値は、占有音長の音長が採用される。

(例) d8/ [r m4 f] s d8/ [r8/ m4 f4]8/ s8/

連符における省略音長の継承プロセスも上記図解と同様のパターンを取ります。

(例) d8/ (r m4 f) s d8/ (r8/ m4 f4)8/ s8/

鍵括弧 [] の記述は、主に和音表現のために使用しますが、括弧内の音長に関わりなく次の音を出すタイミングをコントロールできる記法とも解釈できます。このために、鍵括弧に添える音長にゼロを添えることに意味があります。例えば、ドの音を全音符で鳴らし続けながら、1 オクターブ上でドミソの3連符を演奏する場合は、次のように記述できます。

[d1]0 < (d m s)1

なお、中身のない和音は、休符と同じ意味になります。

[]4 _4

また、ほんの少しだけ次の音にかぶる長さを指定することで、粘りのあるスラーやレガート、テヌートの表現に利用することもできます。この際、省略音長の伝播プロセスを利用すると、和音括弧内には、粘る音長だけの簡素な記述が可能です。

d4 m4 s4 <d4> _1 ; 通常演奏
d4 [m4^16]4 [s4^16]4 [<d4^16>]4 _1 ; 16分音符だけ次音と重なる指定
d4 [m^16] [s^16] [<d^16>] _1 ; 省略音長を活用

本来のスラーは、各音の発音アクセントを柔らかくする奏法指定であり、前音を次音にかぶせるといった音長に関するものではありません。言換えると音の終了側（音長を延す）ではなく、音の開始側（アタックを弱くする）に関わる効果です。したがって上記の手法は、あくまでも擬似的なものです。

2.1.17 コードによる伴奏

伴奏を表現する手段として、音楽には“コード”という優れた記法があります。Muse は 43 個のコードをサポートしており、容易に伴奏を付けることができます。対応している各コードの詳細については 3.6.3 コード (p.74) を参照して下さい。

コードはシングルコーテーション ' ' でくくって表記します。通常譜面には、英米語表記の CDEFGAB の音階に ♯ や ♭ が添えられてルート音が指定され、更にコードネームが続けて書かれています。Muse 記法ではシングルコーテーションの中に、それをそのまま書写せば良いようになっています。また、シャープやフラットは +, - の表記ではなく # と b (小文字の B) を使って下さい。音長の指定は、閉じ側のシングルコーテーションの後に音長表現を添えることで行います。なお、この音長も省略可能であり、省略音長の概念が適用できます。また、オクターブも指定に従います。

'C#m7'4 ; C♯のマイナーセブンスを4分音符で演奏する

コードネームの後には、丸括弧で括ったテンションを与えることができます。テンションは、-9, 9, +9, 11, +11, -13, 13 の 7 種類の中から任意に指定できます。複数のテンションを与える場合は、カンマで区切って下さい。また、コードネームやテンションの後に、分数コード記述によるベース音の追加も可能です。分数コードはスラッシュ / で区切り、その後にルート音と同じ表記方法でベース音を記述して下さい。

このようにコード記述はいろいろな付加指定が可能なので、これらを理解して頂くために、少しずつ指定を膨らませていく例を以下に示します。

```
'D'4      'D'      ; コード D を 4 分音符で 2 回鳴らす
'Dsus4'4  'Dsus4'  ; コード D をサス 4 にする
'Dsus4/G'2      ; これにベース音 G を添えて 2 分音符で鳴らす
'Dsus4(9)/G'2   ; ナインスのテンションを追加する
'Dsus4(9,13)/G'2 ; 更にサーティーンスのテンションも追加する
```

本来、伴奏コードは次章 2.2 アンサンブル演奏 (p.28) で紹介するメンバーを用いた方法で記述した方が格段に楽に入力できますが、まだ紹介していないので、ここでは前述の和音記法を使用して書いてみましょう (メンバー記述での例は 2.2.18 コードの転回形 (p.42) を参照のこと)。

以下の例は、始めは伴奏なしで演奏し、次に同旋律に伴奏を付けて演奏してみます。少々ごちゃごちゃしていますが、 α と β で旋律の位置は合わせてありますので、よく見比べてみて下さい。

山の音楽家 (ドイツ民謡)

```
%92 \+
;  $\alpha$ . まずは無伴奏
r8 s8 s l l c8. <d16r8d> c c l l s4 _8
;  $\beta$ . 次は伴奏付き
r8 [s'G'4]8s[18'D'4]1[c'G'2]8. <d16r8d> [c'G'4]c[1>'D7'4<]1 [s'G']4_8
```

なお、C・D・E をルート音とするコードは、現在のオクターブ値が採用されますが、F・G・A・B をルート音とするコードは、1 オクターブ低い側にシフトすることで、コード全体の音域を極力現在のオクターブ指定内に納めようとしています。またコードにおいては、その調性が既に考慮されて譜面上に記譜されているので、調性指定 \ と音部記号 ? の指定はあえて反映しないようになっています。

2.1.18 音符の連結

例えば、4 分音符とそれに続く 3 連符の初音との間がタイになっている場合、加算音長でそのタイを記述することはできません。そのような場合は & 指定を利用します。& 指定は、連続する同音階の音符を切れ目なく演奏させる指定です。

```
d4 & ( d m s )4 初音がタイ接続されている場合
( s m d )4 & d4 終音がタイ接続されている場合
```

連符内に記述することもできます。

```
d4 ( & d r m f s & s & s f m r d & )1 d4
```

この & 指定は加算記号での表記よりも本来のタイの記述に近いと言えます。和音やコードの連結も可能であり、その場合は互いの構成音の内音階の等しい音を連結します。

```
[ d m s ] & [ d f l ] ドの音だけがタイでつながる
[ d m s ] & m       ミの音がタイでつながる
s & [ d m s ]       ソの音がタイでつながる
```

ただし、和音内の要素音同士に連結記号 & を記述しても効果は現れません。和音内で音長を連結したい場合は、加算記号 ^ で表現して下さい。

```
[ d8 & d32 m2 s2 ]4   エラーにはならないが連結はしない
[ d8^32    m2 s2 ]4   8分音符 + 32分音符の長さが与えられる
```

和音内に & が記述された場合、& 記号の直前の音符が、その和音の次に続く音符や和音と連結対象となります。以下は和音同士の連結例です。

```
[ d m s ] & [ d m s ]   ドミソすべての音が連結する
[ d m s & ] [ d m s ]   ソの音だけが連結する
[ d m & s ] [ d m s ]   ミの音だけが連結する
[ d & m s ] [ d m s ]   ドの音だけが連結する
[ & d m s ] [ d m s ]   エラーにはならないが、どの音も連結しない
```

音の強弱 v が異なる音符を連結した場合は、前音の強弱値が採用されます。

& の前後の音程が異なる場合は連結はしませんが、前音のスタッカート打消して通常の音長となります。この効果は & が添えられた音だけに働くため、スタッカート付き旋律の一部を一時的にスラーにして演奏する場合に便利です。

```
d4/// r m f& s l c <d>
```

2.1.21 グループ感を与える (p.24) で解説する止音調整が指定されている場合も、上記の通常音長化が働きます。ただし音長を通常よりも伸ばす側に指定している場合は & を添えても音長は変化しません。

2.1.19 再現表記の活用

同じ和音やコードが繰返されたり、リズムだけが変化する場合、思いの外記述量が増えてしまいます。

```
[l<df>]4/ [l<df>] [l<df>] [l<df>] [l<dm>] [l<dm>] [sc<r>] [sc<r>]
<'F' 'F' 'F' 'F'> 'Dm' 'Dm' 'C' 'C' 'Dm' 'Dm' 'Dm' 'Dm' 'D'1
```

このような場合は“再現表記”を活用すると簡潔に記述することができます。この表記はカンマ、を記述することで行います。カンマ部分には直前で鳴らした和音・コードの記述が再現されます。上記の例を再現表記を用いて表すと、以下ようになります。

```
[l<df>]4/ ,, [l<dm>] , [sc<r>] ,   和音
<'F' ,, , > 'Dm' , 'C' , 'Dm' ,, , 'D'1   コード
```

再現表記は休符や単一音を対象にしないので、休符や単一音が間で演奏されても直前の和音・コードを再現することができます。また連符内でも記述可能です。

```
>s8.< [dms]16 _8. [dms]16 >s8.< [dms]16 ([dms][dms][dms])4
```

```
>s8.< [dms]16 _8. ,16 >s8.< ,16 (,,,)4
```

再現表記は、添える音長数字を対象としないので、音長を変化させる場合でも利用することができます。また、省略音長を併用すれば、より簡潔な記述になります。更に初音にゼロ音長を指定することでリズムパターンを完全に分離できます。

```
'Cm'8 'Cm'8 'Cm'16 'Cm'8 'Cm'16 'Cm'8 'Cm'8 'Cm'8 'Cm'8
'Cm'8 ,8 ,16 ,8 ,16 ,8 ,8 ,8 ,8   再現表記を活用
'Cm'8, ,16 ,8 ,16 ,8,,,          更に省略音長も活用
```

'Cm'0 ,8,,16,8,16 ,8,,, リズムパターンを分離

最後のリズムパターンの分離は 2.3.8 演奏パターンの部品化 (p.49) で解説するマクロ記述で効果を発揮します。

2.1.20 アクセントを付ける

音楽には、特にリズムを司る重要な要素としてアクセントがあります。そして、アクセントは拍子と密接な関係があります。拍子とは本来、同じ間隔で記譜を区切るという意味以上に、周期的に発生する強弱の波を表す意義があると言えます。“4/4 拍子と 8/8 拍子は、数学的には約分で同じ結果になるが、物理的には音楽の周波数が異なる”とでも申しませうか。従ってアクセントは、音楽の流れの中で、聴く者に拍子(周波数)を感じさせる重要な役割を果たすこととなります。

Muse 記法には拍子の概念はありませんが、このアクセントを簡便に指定する表記を用意しています。小文字の *w* に -127~+127 の数値(アクセント値)を添えた記述を行うことで、その次に鳴る音のみ、音の強さ *v* に変化を与えることができます。同様の演奏表記は、*v* 指定そのものでも可能ではありますが、*w* を使用することで元の強さに戻す手間が不要となります。下記記述は、前半でアクセント無し、後半でアクセントを付けた例です。アクセントを付けることで、演奏を一段と躍動的なものにすることができます。

```
ベートーベン「トルコ行進曲」
%105 v60
<s8mmm| smm<m| r16d>c1 s8f | mfs4> ; アクセント無し
w+40 <s8mmm|w smm<m|w r16d>c1 w s8s+|w l2> ; アクセント有り
```

w に添えるアクセント値は省略することが可能であり、一層コーディングの見通しを良くします。省略した場合は直近の *w* で指定された増分値が採用されます。増分値を変えたい場合は、再度 *w* に数値を添えて指定して下さい。なお、*w*0 と記述するとアクセント指定が無効となります。

和音やコード、連符の直前に *w* を指定した場合は、それら音群全体にアクセントが付きます。一方、和音や連符の中の 1 音のみに *w* を指定すれば、その音だけにアクセントを付けることもできます。両者の記述を施した場合は、音群全体の指定が優先しますが、和音や連符の中にも *w* がある場合は、それ以降のアクセント値はその値に変更されます。

2.1.21 グループ感を与える

人間の行う演奏は、楽譜通りではない微妙なタイミングのずれが起ります。そして人はそれを意図的に行なうことで、より高度な音楽表現を与えます。これらのノリ(グループ感)は、逆に電子音楽には不得意な領域です。タイミングが正確であるが故に、楽譜通りにデータ作成すると律儀で杓子定規な演奏になってしまうからです。

出音タイミング調整 *p* と、止音タイミング調整 *q* は、各音の発音に対し微妙な味付けをするための指定です。この指定をしておくことで以降の音符に対して、音の出だしや音を止めるタイミングをずらすことができます。この指定方法には、対象とする音符の音長を基準とした相対指定と、ずらす音長を直接与える絶対指定とがあります。相対指定は、*p* や *q* に +, - の符号を添え、1~100 の数値を与えることで、本来の音長を基準にして 1%~100% のズレが起こります。絶対指定は、*p* や *q* に加算記号 ^ や減算記号 ~ を添えて、その後に音長表記を与えることで、その音長分のズレが一律に起ります。

- 出音タイミング調整

音が出るタイミングを調整する指定です。音の出だしを早めたり遅くしたりすることで、突っ込みやタメのニュアンスを与えることができます。特に、リズムを刻むパートに活用すると、楽曲全体にグループ感を与えられます。またソロパートに活用すると、ソロを浮き立たせる効果も出ます。

p+10 音長の 10% 分、出音を早めます。
p-15 音長の 15% 分、出音を遅くします。

p^32 32 分音符分、出音を早めます。
 p~16. 付点 16 分音符分、出音を遅くします。
 p0 調整を行いません。デフォルトです。

- 止音タイミング調整

音が鳴り止むタイミングを調整する指定です。短めにしたり長めにしたりすることで、軽快な感じや粘りのある感じを与えることができます。原理的にはスタッカート指定と同じ効果がありますが、よりきめ細かく指定でき、各音符への指定をする必要がないので記述量が減ります。

q+10 音長の 10% 分、音が長く鳴ります。
 q-15 音長の 15% 分、音が短めになります。
 q^32 32 分音符分、音が長く鳴ります。
 q~16. 付点 16 分音符分、音が短めになります。
 q0 調整を行いません。デフォルトです。

以上の調整を図示すると以下ようになります。

出音 止音
 音が出る = = = = = 音が止る

p0	q0					
p+	p-	q-	q+	相対指定系列	
p^	p~	q~	q^	絶対指定系列	

スタッカート指定 / が付いている音符には q 指定は作用しません。つまり q 指定よりもスタッカートの方が優先度が高いことを意味します。

止音タイミング調整は、更に“固定音長指定”が可能です。q 後に直接音長を記述した場合にこの指定となります。この指定がなされていると、それ以降の音符がいかなる音長であっても、“音が出ている長さ”はこの指定音長となります。

(例) q8 d4 r2 m4

4 分音符	2 分音符	4 分音符
d	r	m
8 分音符	8 分音符	8 分音符

この指定は、和音による占有音長記述を簡便化する方法と解釈することもできます。上記の例を和音記述で表現すると以下ようになります。

[d8]4	[r8]2	[m8]4
q8	d4	r2 m4

この指定を解除する場合は q0 を記述します。

2.1.22 アルペジオの記述方法

ピアノの譜面によく見かける記号で、和音音符の左側に縦の波線が添えられたものがあります。これはアルペジオ（分散和音）奏法の一つで、和音を同時に奏するのではなく、多少ずらしながら奏でいく指定です。この奏法は、ギターの演奏ではその楽器の特性上自然に発生するパターンです。このパターンを記述するために、和音やコードの“発音遅延”指定があります。和音やコードの音長にコロンを添え、その後にはずらす音長を与えることで表現します。

```
\+++
[ml<dm>]2:32 _4 [<dm>]2:32 _4 [l<rf>]2:32 _4 [s<rm>]2:32
```

'G'1:64 'E7'2:64 'G7/D':64 'C'1:32 ,1:32 ,1:8.

発音遅延の順番は以下の通りです。

- 和音の発音遅延はその記述音順に従う。
- コードの発音遅延は低音部から高音部に向けて起こる。

64 分音符よりも短い発音遅延をしたい場合は、微分音長 *i* を利用して下さい(2.1.23 最小分解能と微分音長 (p.26) を参照のこと)。

なお、コロンに添える遅延音長は、通常の省略音長と独立の省略が可能です。よって一度定義すれば、後はコロンだけの記述で同様の発音遅延が起ります。コロンを記述しない場合は、アルペジオが起らず通常の音長が再現されます。

'F7-9/D'4:32 ,: ,: ,:

コードの場合は、遅延音長の末尾にマイナス記号を添えることで、発音の順番を逆向きにすることができます。ただし、あくまでもコード記述にのみ有効で、和音の発音順番には影響を与えません^[5]。このマイナス記号による発音方向の切替えを利用すれば、ダウン・ストロークとアップ・ストロークを交互に繰り返すギターの奏法などを簡便に記述することができます。

'C'4:32 'C'4:32- 'C'4:32 'C'4:32-
'C'4:32 ,:- ,: ,:- 上記の省略記法

コードと和音を組合わせて表記してもアルペジオを実現することができ、コードに音を追加していくことが可能です。この際も、上記の発音順番に従います。

'G7'1:16
['G7'c<rf>]1:16
[>>c<r<'G7'c<rf>]1:16

2.1.23 最小分解能と微分音長

Muse は音長に関して、64 分音符の 1/60 の最小分解能を持っています。64 分音符以下の微分音長は、減算記号 \sim を使うことで、

64.~64

のように記述できますが、よりきめ細かく指定するために微分音長記号 *i* があります。この指定は *i* に 1~960 の数値を添えて、その音長を指定する方法です。*i*60 が 64 分音符と同じ音長を表現し、*i*30 で 64 分音符の半分の音長を表現します。したがって、Muse の最小分解能は *i*1 となります。

*di*60 *d*64 と同等
*di*30 *d*64.~64 と同等
*di*15 *d*64..~64. と同等

アルペジオの発音遅延部分や、微妙に発音をずらして演奏効果を出す場合に、微分音長を利用すると効果的です。

[dms]4:*i*10 発音遅延に利用した例
*d*4~*i*10 微妙に音長を延した例

微分音長は、通常の音長数字 1, 2, 4, 8, 16, 32, 64 と同じ局面で記述することができます。また、音長数字と微分音長とを混在させた加減乗算も可能です。ただし、微分音長自体には付点を添えることはできません。

*di*10'5~*i*3 微分音長を含む演算

di45. NG (付点は付けられない)

2.1.24 エコーをかける

生の楽器演奏の模倣として電子音楽をとらえる限り本物を越えることはできません。しかし、電子音楽だからこそその演奏効果も存在します。エコーは、MIDI 界ではディレイと呼ばれており、このような効果は電子音楽の得意とする所です。今までの Muse 記法を組み合わせ、エコーを表現する例を以下に示します。

```
v127 [(dms<d>)1]8 v-10, v-10, v-10, v-10, v-10, v-10,  
v-10, v-10, v-10, v-10, v-10, v-10, _2.
```

上記の例は和音と連符を組合わせて利用し、発音をずらしながら相対強弱指定で少しずつ小さな音にしていくことでエコーを実現しています。再現表記を活用することで見通しの良い記述になっていますが、2.2.3 繰り返し記述と複写機構 (p.31) を利用することで更に簡素に記述できます。

```
v127 [(dms<d>)1]8 {v-10,}12 _2.
```


2.2 アンサンブル演奏

2.2.1 メンバーとフィンガーの概念

まず、Muse における演奏シチュエーションをご説明します。演奏者は 16 名おり、それぞれ A さん、B さん O さん、および Z さんというニックネームで呼ばれます。これら演奏者を今後“メンバー”と呼ぶことにします。目の前に 128 種類の楽器が準備されており、それぞれのメンバーが自由な楽器で演奏できます。演奏中の楽器の持替えもできますし、同じ種類の楽器を複数のメンバーが持つこともできます。ただし、Z さんだけは生れながらのドラマーで、パーカッション以外は演奏できません^[6]。

各メンバーには決った色が割当てられており、演奏の際にどのメンバーがどのように弾いているかをメインウィンドウのピアノ鍵盤上で確認できます。自動演奏の際の配色を考慮してメンバーを選ぶのも一興です。メニューの「機能 (G)」配下にある「メンバー情報 (M)」にて各メンバーの色を確認して下さい。

次に A さん 1 人に着目してみます。彼は 10 本の指を持っており各指は、0~9 の数字が割当てられています。この指を今後“フィンガー”と呼びます。生身の人間は 1 本の指で 1 音を奏でるのが精一杯ですが、この Muse メンバーは皆、超絶技巧の持主で、1 フィンガーで和音やコードを演奏することができます。実は前章 2.1 ピアノ独奏 (p.13) で紹介した演奏は、すべて A さんの 1 本の指で行っていたのです。

Muse はいきなり音符を書き始めると、初めてのフィンガー宣言 # が来るまでの間は #A0 であると解釈します。ピアノ独奏では、この解釈を前提にコーディング例を示していました。

音符を書く前に # に続けて、このメンバーとフィンガーを記述することで、どの演奏者のどの指で弾くのかを指定します。このメンバーとフィンガーをセットにした # 指定を、今後“フィンガー宣言”と呼びます。すべての音はどれかのフィンガーに属しており、次のフィンガー宣言が現れるまで同じフィンガー音と解釈されます。では、A さんの 2 本の指と、K さんの 1 本の指で演奏してもらいましょう。

```
; ドボルザーク「新世界より」第 2 楽章
%60 \=
```

```
#A1 o5 v120
```

```
|m8.s16s4m8.r16d4      |r8.m16s8.m16r2      |
|m8.s16s4m8.r16d4      |r8.m16r8.d16d2      |
|l8.<d16d4>c8s14        |l8<d>cs12            |
|l8.<d16d4>c8s14        |l8<d>cs12            |
|m8.s16s4m8.r16d4      |r8.m16s8.m16r4r+    |
|m8.s16s4<d8.r16m8m>   |<r8.d16r8>l<d2>      |
|<r8.d16r8>l8<d2>      |<r4.d8r4>l<d2._4    |
```

```
#A2 o4 v120
```

```
|m8.s16s4m8.r16d4      |r8.m16s8.m16r2      |
|m8.s16s4m8.r16d4      |r8.m16r8.d16d2      |
|f8.l16l4s8mf4         |f8lsmf2             |
|f8.l16l4s8mf4         |f8lsmf2             |
|m8.s16s4m8.r16d4      |r8.m16s8.m16r4r+    |
|d8.m16m4m8.f16s8s     |f8.f16f8fm2         |
|f8.f16f8fm2           |f4.f8f4fm2._4      |
```

```
#K1 o3 v120
```

```
|'C'2'C'                |'G7'2'G7'           |
|'C'2'C'                |'Dm7'4'G7\nlb''C'2 |
|'F'2:64'F':64          |<'F':64'F':64>     |
```

```

|'F':2:64'F':64      |<'F':64'F':64>      |
|'C'2'C'              |'G7'2'G7'          |
|'C'4'G\nlb''Am\nlb''C'|'G7'2'C'          |
|'G7'2'C'              |['G7'c]1:16'C'2._4 |

```

フィンガー宣言 # はたとえ断続的に現れても、前回のそのフィンガーの音から継続して演奏されます。したがって、上述の演奏は以下のようにスコア風に記述することもできます。いわば、パート譜をスコアに書き換えたといった所です。

; ドボルザーク「新世界より」第2楽章（スコア風の記述方法）

%60 \=

#A1 o5 v120

#A2 o4 v120

#K1 o3 v120

```

#A1 |m8.s16s4m8.r16d4      |r8.m16s8.m16r2      |
#A2 |m8.s16s4m8.r16d4      |r8.m16s8.m16r2      |
#K1 |'C'2'C'                |'G7'2'G7'          |

```

```

#A1 |m8.s16s4m8.r16d4      |r8.m16r8.d16d2      |
#A2 |m8.s16s4m8.r16d4      |r8.m16r8.d16d2      |
#K1 |'C'2'C'                |'Dm7'4'G7\nlb''C'2 |

```

```

#A1 |l8.<d16d4>c8s14        |l8<d>cs12            |
#A2 |f8.l16l4s8mf4         |f8lsmf2              |
#K1 |'F':2:64'F':64        |<'F':64'F':64>      |

```

```

#A1 |l8.<d16d4>c8s14        |l8<d>cs12            |
#A2 |f8.l16l4s8mf4         |f8lsmf2              |
#K1 |'F':2:64'F':64        |<'F':64'F':64>      |

```

```

#A1 |m8.s16s4m8.r16d4      |r8.m16s8.m16r4r+    |
#A2 |m8.s16s4m8.r16d4      |r8.m16s8.m16r4r+    |
#K1 |'C'2'C'                |'G7'2'G7'          |

```

```

#A1 |m8.s16s4<d8.r16m8m>   |<r8.d16r8>l<d2>      |
#A2 |d8.m16m4m8.f16s8s     |f8.f16f8fm2          |
#K1 |'C'4'G\nlb''Am\nlb''C'|'G7'2'C'          |

```

```

#A1 |<r8.d16r8>l8<d2>       |<r4.d8r4>l<d2._4     |
#A2 |f8.f16f8fm2          |f4.f8f4fm2._4       |
#K1 |'G7'2'C'              |['G7'c]1:16'C'2._4 |

```

2.2.2 演奏タイミングを強制的に合わせる

テンポ値が添えられていない単独の % 指定は、タイミングを合わせるという意味で非常に強力な指定です。この指定を行うと、それまで記述していた音符群は、全メンバー、全フィンガーを通じて、時間的に同じスタートラインに整列します。その整列タイミングは % 指定が与えられた時点で、全体を通じて最も先まで進んでいる音符位置が採用されます。図解すると以下ようになります。

例えば、以下のような記述をしたとします。

```
#A0 | α          |
#A1 | β                      |
#B0 | γ          |

#A0 | δ          |
#A1 | ε          |
#B0 | ζ          |
```

フィンガー宣言はたとえ断続的に現れても、前回のそのフィンガーの音から継続して演奏されるため、上記は実際の演奏では以下ようになります。

```
#A0 | α          || δ          |
#A1 | β                      || ε          |
#B0 | γ          || ζ          |
```

今度は、 γ と δ の間に % 指定を行ったとします。

```
#A0 | α          |
#A1 | β                      |
#B0 | γ          |

% ;   ここで % 指定を行った

#A0 | δ          |
#A1 | ε          |
#B0 | ζ          |
```

すると、% が記述された部分で全メンバー・全フィンガーのタイミングが強制的に整列するので、最も先まで音符のある β まで休符が挿入され、実際の演奏は以下ようになります。

```
#A0 | α          |   休符   | δ          |
#A1 | β          |           || ε          |
#B0 | γ          |   休符   | ζ          |

                この位置にタイミングが整列する
```

この強制タイミング合せの効果は、あくまでもテンポ値の添えられていない単独の % が指定された場合にのみ起り、テンポ値を与えた場合はその限りではありません。タイミングを合せしかもテンポ値も変えたい場合は、

```
%
%120
```

というように % を 2 つ記述して下さい。1 つ目でタイミングがそろい、2 つ目でテンポが変わります。

明に記述していない休符が与えられるということを意識しておけば、逆にこの強引な処理を積極的に使用し、一気に全メンバーのタイミングを合わせることが可能となります。例えば、たまにしか演奏に参加しないメンバーがある場合、延々と休符を記述して出番を待つかわりに、出番の直前で % を書けば事足りる訳です。

ただし、前半 (α, β, γ) に記述した内容が、整列タイミングを越えて後半 (δ, ϵ, ζ) に波及する場合があります。それは、以下の 3 つの場合です。

1. 前半の音符が連結 & で終了している場合
強制休符が挟まれなければ、前半の音と後半の音がきちんと連結されます。
2. メンバー属性指定で与えた遅延音長が、タイミング整列位置を越えている場合
後半に属性の効果が雪崩れ込み、遅延速度は後半のテンポに従います。
3. 和音記述において、占有音長より長い音長を和音内音符に与えた場合

後半まで音が持続されます。後半の音長は後半のテンポに従います。

2.2.3 繰返し記述と複写機構

音楽はその特性上反復が多く出現します。伴奏やリズム、そして歌の1番と2番の旋律などです。その様な場合、同じパターンを繰返させる指定を使うと記述量を減らすことができます。それは任意の部分の中括弧で括り、その後ろに2以上の数値を添えることで指定します。この指定により、添えた数値の回数分、中括弧内の演奏を繰返します。なお、数値を省略した場合は1と見なされます。また、ゼロを添えようと中括弧内を演奏させない指定となります。

```
#A0 { dms }3 ; ドミソを3回繰り返す
#A0 { dms } ; ドミソを1回繰り返す
#A0 { dms }0 ; 何も演奏しない(複写対象には成る)
```

複数のメンバーをゴツゴツ括って繰返させる事もできます。

```
{ #A1 | m8.s16s4m8.r16d4 | r8.m16s8.m16r2 |
  #A2 | m8.s16s4m8.r16d4 | r8.m16s8.m16r2 |
  #K1 | 'C'2'C' | 'G7'2'G7' | }3
```

中身の無い中括弧を記述すると、その直前に指定された中身のある中括弧の内容がその分に展開されます。この“複写機構”とも言える記法は、少し離れた位置に同一パターンがある場合に活用すると便利です。また中括弧は入れ子の記述も可能ですから、工夫すると記述量がぐっと減ります。

```
#A0 dms ccc dms rrr | dms ccc dms rrr | dms ccc dms rrr
#A0 { {dms} ccc {} rrr }3
```

中括弧以外の部分にoやvの指定をすれば、ユニゾン演奏として同一の旋律をオクターブ上や下で鳴らし、更に各メンバーの音量に変化を与えるような記述も簡便にできます。

オクターブ違いのユニゾン表現の例

```
#A0 o4 drmf slc_ cccc mrd_
#B0 o5 drmf slc_ cccc mrd_

#A0 o4 { drmf slc_ cccc mrd_ }
#B0 o5 { }
```

同一旋律で音の強さを変化させる例

```
#A0 v80 fsfs ccc_ <d>cls mrd_
#B0 v50 fsfs ccc_ <d>cls mrd_

#A0 v80 { fsfs ccc_ <d>cls mrd_ }
#B0 v50 { }
```

2.2.4 パーカッションの演奏方法

パーカッションはドラムメンバー Z さんの0~9の10本のフィンガーにかかっています。他のメンバーはパーカッションを叩くことはできません。逆にドラムメンバー Z さんはパーカッション専任です^[6]。

ドラムの種類は、ドラムメンバーに記述する音名に対応しています。よって、ドラムメンバーにも音階や音長が存在するため、和音記述や連符記述なども可能です。以下に、次々に音程を変えることで、いろいろなドラム種を鳴ら

す例を示します。

```
#Z0 v127 o1
d+2
r4 r+ m8 f f+4 s8 s+ l l+ _16 c < d8 d+4
r4 r+ m8 f f+4 s8 s+ l l+ _16 c < d8 d+4
r4 r+ m8 f f+4 s8 s+ l l+ _16 c < d8 d+4
r4 r+ m8 f f+4 s8 s+ l l+ _16 c < d8 d+4
r4 r+ m8 f f+4 s8 s+ l l+ _16 c < d8 d+4
r8 r+8
```

各音程に対して、どのようなドラムが割当てられているかは「機能 (G)」 「ドラムの試聴 (Z)」のメニューで実際に試聴してみてください^{[7][8]}。

ドラムメンバーは、他のメンバーとは異なる特性を持っているため、指定しても無効になる属性や、そもそも指定不可能な属性があります。

- 調性指定 \ 無効 2.1.4 調性の指定方法 (p.14)
- 音部記号 ? 無効 2.1.15 音部記号について (p.20)
- コード記述 ' ' 不可 2.1.17 コードによる伴奏 (p.21)
- 移調指定 T 無効 2.2.10 移調の仕方 (p.36)

2.2.5 フィンガー単位に指定する値

この節でご紹介することは、既にピアノ独奏で出てきた機能がほとんどですが、フィンガーという概念を得たので再度確認します。以下の 12 種類の値は、各フィンガー毎に独立して存在しています。

- 音名タイプ x
- オクターブ o
- 音の強弱 v
- アクセント w
- 音部記号 ?
- 出音タイミング p
- 止音タイミング q
- 移調楽器 y 2.3.14 移調楽器をフィンガー毎に割り当てる (p.55)
- 和音/コード再現 ,
- 連符再現 ()
- アルペジオ音長 :
- 省略音長

これらの値を“フィンガー属性”と呼びます。フィンガー属性がセットされる先は、直前のフィンガー宣言 # で指定されたフィンガーになります。各フィンガーで独立なこれらの値は、16 メンバー × 10 フィンガー すなわち 160 の個別指定ができることになります。他のフィンガーで異なる値を指定をしても、各フィンガーで独立に値を保持するため、何度も指定し直す必要はありません。

2.2.6 メンバー単位に指定する値

各メンバーには、それぞれ独立にいろいろな値を指定することができます。これらの値を“メンバー属性”と呼びます。メンバー属性はメンバー単位に設定するため、フィンガー毎に異なる値を指定することはできません。1 メンバー内のフィンガーですべて同値になります。なお、メンバー属性の指定文字はすべて大文字なのが特徴です。

- 音色 P

- 音量 V
- ステレオ S
- 移調⁹⁾ T
- 残響 R
- 揺らぎ W
- コーラス Q
- ピッチ U
- ペダル Y
- 出力変位特性 R=
- 変調特性 W=
- 周波数特性 Q=
- コントロール X

さて各属性値の詳細は後節に譲るとして、ここではこれらの値を各メンバーに指定する方法をご紹介します。それには@にメンバー記号A~0, Zを添え、まずメンバーを宣言した後、メンバー属性の値を記述していきます。すなわち、以下のような形式で指定します。

```
@E P23
```

この例は、Eメンバーの音色Pを23番(ハーモニカ)に指定した場合です。この@Eの部分とP23の部分は、離れていても構いません。以降@Eの部分で“メンバー宣言”、P23の部分で“メンバー属性”と呼びます。メンバー属性が記述された際、その属性値が反映されるのは、あくまでも前もってメンバー宣言で指定しておいたメンバーが対象になります。なお、メンバー宣言もメンバー属性も曲の途中で何度でも指定し直せます。

メンバー宣言におけるメンバー記号A~0, Zは、列挙記述が可能であり、複数のメンバーに対して一気に指定できるのが特徴です。例えば、A・B・Cさんを同じ楽器、音量、移調にしたければ、以下のように1行で指定できます。

```
ドビュッシー「亜麻色の髪の乙女」
%66 \=
```

```
@ABC P101 V120 T0 ;      ここで一気に指定 (PVSTRWQUY の記述順は自由)
```

```
#A0 ?0 o4 v120
#B0 ?6 o4 v120
#C0 ?0 o4 v120
```

```
#A0|<s4^8m16d>l8<d16m|s8m16d>l8<d16md8d16>l|<d8>c16ls2^ |2l8<d|
|r4.s8md16m|r8sm4l4^1|_2
#B0|_2.|_4 _ _8 <r> |l2.^ |2 <_4|m2f4 |m4d4[d>f<]4^1 |
#C0|_2.|_4 _ _8 'F'8|'C'2.^2|_4 |'G'2' Am'4|'G\n1b''Em\n1b''Am'4^1|
```

また@に添えるメンバーは省略可能であり、省略するとその時点のフィンガーが属するメンバーを指定したことになります。

```
#A0 d r m @A P23 f s l
```

```
#A0 d r m @ P23 f s l ;      メンバー A が省略されていると解釈される。
```

メンバー属性を音符の間に挟み込むことで、その属性値を反映させるタイミングを、指定することができます。挟み込む例を以下に示します。4つの例とも、同じ演奏をします。

```
@A #A0 d m s V80 d r m V90 d f l V120 d ; @メンバーを明示する場合
#A0 @ d m s V80 d r m V90 d f l V120 d ; @メンバーを省略する場合
#A0 d m s @V80 d r m @V90 d f l @V120 d ; 逐一@を添える場合
```



```
*TEXT"エレクトロニック" P25 #Z0 o2 (drmfslc<dr)2._4
*TEXT"アナログ" P26 #Z0 o2 (drmfslc<dr)2._4
*TEXT"ジャズ" P33 #Z0 o2 (drmfslc<dr)2._4
*TEXT"ブラシ" P41 #Z0 o2 (drmfslc<dr)2._4
*TEXT"クラシック" P49 #Z0 o2 (drmfslc<dr)2._4
```

ドラムセットによる音色の変化がどのようなものなのかは「機能 (G)」 「ドラムの試聴 (Z)」のメニューで実際に試聴してみてください。標準と異なる音色には、ボタンに黒丸 が付きます^[11]。

2.2.8 クレッシェンドとデクレッシェンド

@ によるメンバー宣言の後に V に添える 0~127 の数字で音量を指定します。音量は 127 が最大で、0 で無音状態となります。

```
@G P055 ; ここでのメンバー宣言が次行の V 指定に効果する
#G0 V50 d4ms<d> V90 d4ms<d> V127 d4ms<d> _2
```

v 指定は各音の発音時点の音量を示しますが、V 指定は発音している最中でも変えることができます。この特徴を引出す方法として“遅延効果”指定があります。これは V の音量値の後にコロンを添え、そこに音長を与えることで表現します。この指定は“与えられた音長分の時間をかけて指定の音量に到達せよ”という指定です。この指定で、なめらかなクレッシェンド・デクレッシェンドが可能です。

```
グルーバー「きよしこの夜」(賛美歌)
%50 @L P071 V20
```

```
#L0 V127:4. _4. V20:4. _4. ; 音量調整用のフィンガー
#L1 v127 o4 s8.116s8 m4.
#L2 v127 o3 'C'2.
```

```
#L0 V127:4. _4. V20:4. _4.
#L1 s8.116s8 m4.
#L2 'C'2.
```

```
#L0 V127:4. _4. V20:4. _4.
#L1 <r4r8> c4.
#L2 'G7'2.
```

```
#L0 V127:4. _4. V20:4. _4.
#L1 <d4d8> s4.
#L2 'C'2. _2
```

V 指定が 1 つのメンバーの全フィンガーに効果する性質を利用して、音量を司る専門のフィンガーを立ち上げると、綺麗にコーディングできます。上記の例は L メンバーの音量全体を #L0 フィンガーで調整しています。

また V の後に符号 +, - 付の数値を指定すると、現在の音量値を相対的に変化させることができます。ただし相対変化の結果が 0~127 の範囲を越える場合は、0 あるいは 127 の限界値がセットされ、文法エラーとしての検出はしません。

```
#B0 @V80 d r m V+100 d r m ; V180 となるので、V127 がセットされる
```

ピアノやマリンバなどの減衰系楽器に V の遅延効果を使用してクレッシェンドやデクレッシェンドをかけると、発音後に音が大きくなるといった不自然な演奏になるのでご注意ください (2.2.17 強弱と音量の関係 (p.41) を参照のこと)。

2.2.9 ステレオ効果の利用

@ によるメンバー宣言の後に S に添える -64 ~ +64 の数値でステレオ効果を指定します。-64 で左側のみ、+64 で右側のみ、0 で中央から鳴ります。この効果を利用すれば、左側から伴奏とパーカッションが鳴り、右側から旋律が聞えるといった広がりのある演奏を記述することができます。

以下の例は左から右に音が移動する記述してみました。S 指定が音符の合間に入り込んでいますが、先に 2 行目で宣言してある @A によりこのステレオ効果は、A メンバーに作用することになります。

```
@A P018 V127 ; 以降の S 指定は、@A により A メンバーに作用する
#A0 v127 o2
```

```
S-64m S-62f S-60f+ S-58s S-56s+ S-54l S-52l+ S-50c S-48<d S-46d+
S-44r S-42r+ S-40m S-38f S-36f+ S-34s S-32s+ S-30l S-28l+ S-26c
S-24<d S-22d+ S-20r S-18r+ S-16m S-14f S-12f+ S-10s S-8s+ S-6l
S-4l+ S-2c S0<d S+2d+ S+4r S+6r+ S+8m S+10f S+12f+ S+14s
S+16s+ S+18l S+20l+ S+22c S+24<d S+26d+ S+28r S+30r+ S+32m S+34f
S+36f+ S+38s S+40s+ S+42l S+44l+ S+46c S+48<d S+50d+ S+52r S+54r+
S+56m S+58f S+60f+ S+62s S+64s+

S+64s+ S+62s S+60f+ S+58f S+56m S+54r+ S+52r S+50d+ S+48d> S+46c
S+44l+ S+42l S+40s+ S+38s S+36f+ S+34f S+32m S+30r+ S+28r S+26d+
S+24d> S+22c S+20l+ S+18l S+16s+ S+14s S+12f+ S+10f S+8m S+6r+
S+4r S+2d+ S0d> S-2c S-4l+ S-6l S-8s+ S-10s S-12f+ S-14f
S-16m S-18r+ S-20r S-22d+ S-24d> S-26c S-28l+ S-30l S-32s+ S-34s
S-36f+ S-38f S-40m S-42r+ S-44r S-46d+ S-48d> S-50c S-52l+ S-54l
S-56s+ S-58s S-60f+ S-62f S-64m
```

S 指定は発音している中でも変えることができます。この特徴を引出す方法として“遅延効果”指定があります。これは S のステレオ値の後にコロンを添え、そこに音長を与えることで表現します。この指定は“与えられた音長分の時間をかけて指定のステレオ位置に到達せよ”という指定です。この指定でなめらかな音の左右移動が可能となります。以下の例は、まず左から右に波が押寄せ、次に右から左にヘリコプターが飛ぶ情景を記述したものです。

```
#L0 o2 @ V127 S-64 ; まずは左位置
P123 S+64:1 d1^4 ; 波の音が右側へ向って移動
P126 S-64:1 d1^4 ; ヘリコプターが左に向って移動
```

ドラムメンバーは初期状態で、各ドラムにステレオ値が与えられています。S 指定はそれらの位置を基準にして、左右にシフトさせます。従って、何も指定しなくてもそれなりの空間配置になっています。

2.2.10 移調の仕方

@ によるメンバー宣言の後に T に添える -95 ~ +95 の数値でキーの変更ができます。マイナスで低音側へ、プラスで高音側へそれぞれ半音単位にシフトします。この機能は、コード記述も含めて指定のメンバー全体に効き目があります。12 の倍数を指定すればオクターブのシフトとしても利用できます。この移調指定は特に移調楽器の譜面を入力する際に効果を発揮します。移調楽器に関する移調量の指定値は 3.2.10 移調 (p.68) を参照して下さい。

```
フォーレ「シチリアーノ」
%80 \=
@HMO V100 S0 T0
@H P001
```

```

@M P074
@O P072

#H0 v127 o5
_2^8m8|l4<d8m4l8 |<d8.>c16l8c4m8|m8.r16f+8m8.r16f+8 |m4.^4>m8 |
      |l4<d8m4s8 |c-8.l16s8l4>18|18.s+16c8l8.s+16<d8|>14.^4_8 ||

#M0 v90 o6 ?6
_2^4 |f8l<d>f1<d>|r+f1dm+s |>l<mf>l<mf |dm+sd4. |
      |f8l<d>f1<d>|rs-cdf1 |>c<flds+c |fd>lf4. ||

#O0 v60 o4
_2^4 'Am' 'F#m7-5' 'D7' 'E' 'Am7' 'Bb'4. 'A' 'Dm7\nlb''E' 'Am'2.

```

@HMO T+3 ; ここで半音3つ分、高音側へ移調

```

#H0 v127 o5
_2^8m8|l4<d8m4l8 |<d8.>c16l8c4m8|m8.r16f+8m8.r16f+8 |m4.^4>m8 |
      |l4<d8m4s8 |c-8.l16s8l4>18|18.s+16c8l8.s+16<d8|>14.^4_8 ||

#M0 v90 o6 ?6
_2^4 |f8l<d>f1<d>|r+f1dm+s |>l<mf>l<mf |dm+sd4. |
      |f8l<d>f1<d>|rs-cdf1 |>c<flds+c |fd>lf4. ||

#O0 v60 o4
_2^4 'Am' 'F#m7-5' 'D7' 'E' 'Am7' 'Bb'4. 'A' 'Dm7\nlb''E' 'Am'2. _2.

```

なお、この移調指定はドラムメンバーに対しては無効です。

2.2.11 楽器の音色をアレンジする

Muse では P 指定により 128 の楽器から音色を選べますが、それぞれの音色に対して更に微妙な味付けを行うことができます。この味付けをメディア界ではエフェクトと称しています。Muse のエフェクトは以下の 3 種類があります^[12]。

- 残響 R
音を止めた際後に残る余韻の強さです。特に減衰系の楽器で効果が顕著です。音はそれが鳴り続けている間でも常に残音を発生していますから、音の累積が起こり音量が上がる効果もあります。メディア用語ではリバースと呼ばれています。
- 揺らぎ W
一般に言われるピブラートです。特に通奏系の楽器で効果が顕著です。メディア用語ではモジュレーションと呼ばれています。
- コーラス Q
同じ楽器でも大人数で演奏すると、それぞれの微妙な音色の違いや奏法のズレで全体の音に深みや厚みができます。この指定は、いわばその人数に比例する値と考えて下さい。0 でソロの音色、127 で大合奏という訳です。当然のことながら、奏者が多いほど音量も増加しますが、ステレオ S の発音位置がぼやける効果もあります。舞台全体に広がるというイメージです。

以上の 3 つの指定は、すべて 0~127 の値を添えてその量を記述します。これらのエフェクト指定は、ドラムメンバーにも効果があります。各数字と実際のエフェクトの関係は、メニューの「機能 (G)」 「楽器の試聴 (P)」や「ドラムの試聴 (Z)」にて、実音で確認して下さい。

一日の終り (フランス古典)

```
%80 \= @M P66 T0
```

```
#M0 o4 ?0
```

```
@M R0 W0 Q0
```

; まずはエフェクトなし

```
#M0 s4.f8ms<dr m2d r8d>c<rd4>l s2._4
```

```
@M R127 W127 Q127
```

; 次にエフェクトを付けてみる

```
#M0 s4.f8ms<dm r2>l c4.l8sc<dr d2._4
```

この節で紹介した3つの指定は、すべて“遅延効果”を与えることができます。指定方法は、音量VやステレオSと同様に、コロンの後に遅延音長を添えます。通奏系の音に対し、次第にビブラートを強くしていったり、次第に演奏参加人員を増やしていくといった効果を表現できます。

```
@N P72 R0 W0 Q0
```

```
#N0 o4 d2& W127:1^2 d1^2 _4 ; 後半で次第に揺らぎを高めていく
```

```
@K P53 R0 W0 Q0
```

```
#K0 o4 Q127:1^2 d1^2 _4 ; 次第に参加人員を増やす
```

2.2.12 なめらかなグリッサンド

トロンボーンやバイオリンは、その楽器特性から半音よりも細かい音程の音を演奏することができます。Museでは、メンバー属性としてピッチ指定が可能であり、半音の1/10精度の微分音程を指定することができます。@によるメンバー宣言の後にUに添える-240~+240の数値でピッチを指定します。指定数値の10が半音分の音程を意味するので、+/-240によって、上下2オクターブ、合計4オクターブ分のピッチシフトが可能です。

```
@A U+120 ; Aメンバーのピッチを1オクターブ上げる指定
```

他のメンバー属性と同様、指定したピッチは次の指定があるまで効果を維持します。標準のピッチに戻す場合は、数値0を指定して下さい。

```
@A U0 ; 標準のピッチに戻す指定
```

更にU指定もVやSと同様に“遅延効果”指定を行うことができます。これはUのピッチ量にコロンを添え、そこに音長を与えることで表現します。この指定は“与えられた音長分の時間をかけて指定のピッチに到達せよ”という指定ですから、なめらかなピッチ変化、すなわちグリッサンドを演奏することができます。エレキギターのチョーキングや管楽器のしゃくり上げなどにも活用でき、音楽表現が多彩になります。

```
@A U-120:4 ; 4分音符分の時間をかけて1オクターブ下げる指定
```

以下は、なめらかなグリッサンドを演奏する例です。

森のくまさん (アメリカ童謡)

```
%80 @G P58 \=
```

```
#G1 s8f+s m4& @U-120:4 m4 _8 @U0 mr+md4 & @U+120:4 d4 _8 @U0
```

```
~~~~~ ~~~ ~~~~~ ~~~
```

ピッチ下降 標準に戻す ピッチ上昇 標準に戻す

ここで紹介したピッチ指定はドラムメンバーにも適用が可能であり、ドラムの表情を変化させることができます。メニューの「機能(G)」 「ドラムの試聴(Z)」にて、実音で確認して下さい^[13]。

2.2.13 ダンパーペダルの操作

ピアノに付いているダンパーペダルの操作は @ によるメンバー宣言の後に Y0 あるいは Y1 で指定します。Y0 はペダルを離す指定、Y1 はペダルを踏む指定です。ペダルが踏まれている間、音の減衰が抑制され、伸びやかに響きます。Y1 が連続して指定された場合、一度素早くペダルを離し、速やかに踏み直します。したがって踏みかえの場合は、Y0Y1 と記述する必要はありません（音源によっては Y1 の記述でペダルの踏み変えが効かない場合があります。その際は適度な間隔を置き、予め都度 Y0 指定をして下さい）。

以下は、ダンパーペダルを使用しない場合（離しっぱなし）と、ダンパーペダル操作をする場合の比較例です。特に低音部の響きを聴き比べて下さい。

ベートーベン「月光ソナタ」

```
@A P1 V127 Q0 R0 W0 S0 ; 以降の Y 指定は @A により A メンバーに作用する
%60 \++++
```

; ペダルを使用しない場合

```
#A1 o4 ?6 v127 [>l4<l]1 [>s4<s] [>f4<f]2[>r4<r] [>m<m], _2
#A2 o4 v80 (>s<dm)4()()()()()(>l<dm)()(>l<r=f)()
(>sc+<f)(>s<dm)(>s<dr+)(>fc+<r+) _2
```

; ペダルを操作した場合

```
#A1 o4 ?6 v127 [>l4<l]1 [>s4<s] [>f4<f]2[>r4<r] [>m<m], _2
#A2 o4 @ v80 Y1 (>s<dm)4()()() Y1 ()()()() Y1 (>l<dm)() Y1 (>l<r=f)()
Y1 (>sc+<f)(>s<dm) Y1 (>s<dr+) Y1 (>fc+<r+) _2 Y0
```

これらの位置にダンパーペダルの操作を挿入した。

なお Y 指定は音の減衰を抑制する指定として、ドラムも含め、ピアノ以外の楽器にも活用が可能です。ただし、通奏系の楽器に指定すると、次に Y0 が来るまでの間、音長を越えて鳴り続けるのでご注意ください^[14]。

2.2.14 全体に影響を及ぼす値

今まで、メンバー単位の設定値 @PVSTRWQUY、フィンガー単位の設定値 ?xvwo<> を紹介しましたが、この節では全メンバー・フィンガーに一気に影響を与える設定値について紹介します。それは既に紹介したテンポ % と調性 \ の 2 つです。これらの設定を切替えると、それ以降に記述するすべてのメンバー・フィンガーはそれに従うこととなります。以下に転調の例を示します。

ガーシュイン「ラブソディー・イン・ブルー」

```
%80
```

```
@DF P001 V127 S0 T0
```

```
\-- ; 始めは変口調
```

```
#D0 ?0 o5 v127
```

```
(mfs)4|l-2(s-f-s-)4(f-s-f-)4|m8r-dd-^r-8r=m^|d=8>l-s-^f-8m4|
```

```
#F0 ?6 o5 v127
```

```
(drm)4|[f-2]4[<d>1]4[1+4d=2]4s=|[f-1=]4.[f-1-]8^2|[f-1=]1||
```

```
\+++++ ; ここで嬰口調に転調
```

```
#D0 o5 v100
_16slsc8c_16slsc<d>cl|_slsc8cc2|_16drdm8m_16drdmfmr|_drdm8mm2|

#F0 ?6 o5 v80
[m<d>]4[l<d+>] [m<r>] [l<d>] | [m<d=>] [c<r>] [<d+>l]8[cs] [lf] [ms] |
[f=>l<]4[f+r] [s>l<] [f+r] | [f=>l<] | [sm] [f+r]8[md] [r>c<] [d>l<] |_2.
```

移調楽器の多彩なアンサンブルで構成されるオーケストラの楽曲は、各楽器毎に異なる調性で記譜されています。このような楽曲を記述する場合は、メンバー毎に調性の指定ができると便利です。これを行うには \ の後にメンバー記号を添え、その後に +, -, = 記号を与えて下さい。例えば、メンバー A のみに ♯ を 3 つ付けたい場合は、以下のようにします。

```
\A+++
```

\ に添えるメンバー記号 A ~ 0 は、列挙記述が可能であり、ドラムメンバーを除く複数のメンバーに対して一気に指定できるのが特徴です。上述の例示楽曲で指定した \-- は \ABCDEFGHJKLMNO-- の省略形と見なすことができます。

2.2.15 リタルダンドとアツチェレランド

テンポを滑らかに連続変化させるには、テンポ指定 % における“遅延効果”を利用します。遅延効果は、% のテンポ値にコロンを添え、そこに音長を与えることで表現します。これは“与えられた音長分の時間をかけて指定のテンポに到達せよ”という指定です。これを使うと、リタルダンドやアツチェレランドなどのテンポ変化が可能となります。

バッハ「小フーガ ト短調」

```
\--
```

```
@FG P020 S0 T0 V120
```

```
%72 ; 始めはアンダンテ
```

```
#F0 v120 o4 ?0 s4<r>c4.l8|sclsf+lr4|
```

```
#G0 v120 o4 ?0 _1|_1|
```

```
%132:1^1^1^1 ; 4小節かけてアレグロにアツチェレランド
```

```
#F0 s8rlrcl16sl8r|s8r16sl8r16lc8l16slr<rd>|
```

```
#G0 _1|_1|
```

```
#F0 clscsf+lsrslc<drm=|f=m=rf=m=rd+m=r8>l<rm=|
```

```
#G0 _1|r4lf4.m=8|
```

```
#F0 f16sfs(slslslsls)8.f32s116slclsfm=|flsld+lsrlsld+ls1|
```

```
#G0 r8fm=rd+m=>l4<|r8>l<m=>l<fm=16rm=8>l<|
```

```
%72:1^1^1^1 ; 4小節かけて再びアンダンテにリタルダンド
```

```
#F0 frd+rsrd+r1rd+rsrd+r|>l8<f>s<m=>fl<rf|
```

```
#G0 r8>l16<rm=8>l16<m=f8m=16rm=16>l<ls|fm=rfm=rd+m=r>l<rm=fslc=|
```

```
#F0 m-l_m-rs_r|d16>c<drdlsl>c<sf+s>lf+m=f+|s2
```

```
#G0 <d>c-<drd>c-l<d>c-lc-<d>c-lsc-|l8sf+rs4__2|
```

休符と速度 % のみで構成された、テンポ指定専用のフィンガーを設定すると、短い区間で肌理細かく演奏速度を増減させるテンポ・ルバート（緩急法）の調整が容易になります。

```
#F0 %100_1 %200_1 %300:1_1 %150_1 %80:1_1
#G0 dms<d> dms<d> dms<d> dms<d> dms<d>
```

2.2.16 ノンレガートな奏法の指定

減衰系の楽器ではあまり目立ちませんが、通奏系の楽器で同一音程を刻んで演奏する際、各音の拍がぼやけた感じになる場合があります。このような場合、各音の後ろにごく短い休符を挿入することで、次音の拍が明瞭に識別できるようになります。この“ノンレガート”の表記は、音長の後ろにスラッシュ / を 1 つ付け、続けていくつかのピリオド . を添えて行います。ピリオドは 3 つまで添えることができ、その個数で音を区切る度合を表します。占有音長を維持したまま音の長さを短くする訳ですから、文法上はスタッカート的一种と考えられます。

また、スタッカート // やスタッカティシモ /// にも、ピリオドを 1 つ添えることができ、それらの間を取った音長を表現します。これらの音長短縮規則を時間軸で示すと以下のようになります。

```
発音開始  = = = = =          占有音長

          ////   ///   ///.   //   //.   /   /. /... /...
```

ノンレガート・マルカートにおける、省略音長の継承などの文法上の扱いは、スタッカートとまったく同じです。以下にノーマルなタンギングから、次第にノンレガートの傾向を大きくし、マルカートから更にスタッカートへと移行していく例を示します。下記の例は、各フレーズ毎に TEXT コマンドを使って文字列を表示させています。TEXT コマンドに関しては 2.3.2 カラオケとして利用する (p.44) をご覧ください。

```
#B0 @P72 W10 Q10 T0 ?0 o4
*TEXT"通常のタンギング" d4 ddd d8 ddddddd (ddd)4 ()()() d1 _
*TEXT"弱めのノンレガート" d4/...ddd d8/...ddddddd (ddd)4/...()()() d1 _
*TEXT"強めのノンレガート" d4/. .ddd d8/. .ddddddd (ddd)4/. .()()() d1 _
*TEXT"マルカート" d4/. ddd d8/. dddddddd (ddd)4/. ()()() d1 _
*TEXT"メゾスタッカート" d4/ ddd d8/ dddddddd (ddd)4/ ()()() d1 _
*TEXT"スタッカート" d4// ddd d8// dddddddd (ddd)4// ()()() d1 _
*TEXT"スタッカティシモ" d4/// ddd d8/// dddddddd (ddd)4/// ()()() d1 _
*TEXT"スタッカティシシモ" d4////ddd d8////ddddddd (ddd)4////()()() d1 _
```

ひとつかたまりのフレーズの最終音にノンレガートやマルカートの指定を行うことで、次のフレーズとの分離性を強調することができます。

2.2.17 強弱と音量の関係

Muse には音の大小を表現する指定として、強弱 v と音量 V の 2 つがあります。両者の違いの 1 つはその指定単位にあります。強弱 v はフィンガー単位に指定する値であるのに対し、音量 V はメンバー単位に指定する値です。例えば、小文字 v はそれぞれの指の強さ、大文字 V はそれぞれの演奏メンバーの体力 (パワー) とでも申しましょうか。

以下で v と V の片方を一定にして、もう片方の値を次第に変化していく実験を行います。曲の前半 α では旋律と伴奏が共にクレッシェンドしていきます。後半 β は旋律がデクレッシェンドしていきますが、伴奏は最後まで音量が変わりません。

```
; ベートーベン 交響曲第 9 番より “喜びの歌”
%104 \+
@G P055 ; 以降の V 指定は @G により G メンバーに作用する
```

```
; α. v を一定にして V を大きくしていく (全フィンガーに効く)
#G0 v127 V25 o5 c4c V30 <dr V40 rd V50 >c1 V60 ss V90 lc V110 c4.18 14_
#G1 v127 o4 'G'1 'D' 'G' 'D'
```

```
; β. V を一定にして v を小さくしていく (1つのフィンガーにだけ効く)
#G0 V127 v110 c4c <dr v90 rd v60 >c1 v50 ss v40 lc v30 14.s8 s4
#G1 v127 'G'1 'D' 'G' 'D'2 'G'2 _1
```

概念的には総音量は v と V の積で決まります。したがってどちらか一方がゼロならば無音状態となります。ただし強弱が $v0$ の場合、そのデータは MIDI 出力の対象になりません。このことは $v0$ が $V0$ と異なり、MIDI 音源に負荷を与えることなく鍵盤描画を行うことを示しています。よって強弱 $v0$ を利用すると、楽曲演奏とは関係のない無音の鍵盤アニメーションを、もたれることなく実現できます。

もう1つの大きな違いは、 v 指定が各音の発音時点の音量を与えるのに対し、 V 指定は発音している最中でも変えることができるという点です。

以上の違いを表にまとめると以下ようになります。

	指定する単位	効果の特性	ゼロ指定時
強弱 v	フィンガー毎	発音開始時点のみ	MIDI 出力しない
音量 V	メンバー毎	発音中も効果する	MIDI 出力する ^(注)

(注) 共にゼロ指定の場合は $v0$ が優先され、MIDI 出力されません。

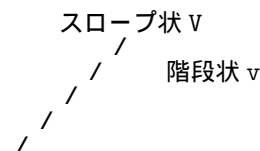
音量指定 V は、発音中も効果するという特徴があるため遅延効果を利用することで、なめらかなクレッシェンド・デクレッシェンドが実現できることは、既にご紹介しました (2.2.8 クレッシェンドとデクレッシェンド (p.35))。

実は、強弱指定 v にも遅延効果の記述が可能です。 V での書式と同様にコロンの続く遅延音長で表現します。下記の例は、遅延効果の特性を比較するものです。各タイプに TEXT コマンドを使って文字列を表示させています。TEXT コマンドに関しては 2.3.2 カラオケとして利用する (p.44) をご覧ください。

```
%60
@H P50 R127

*TEXT"音量 V の遅延効果 (スロープ状に変化)"
#H0 v127 @V20 @V127:1'6 o5 d1^1 r m

*TEXT"強弱 v の遅延効果 (階段状に変化)"
#H0 @V127 v20 v127:1'6 o5 d1^1 r m
```



音量 V がスロープ状に音が大きくなるのに対し、強弱 v は発音のたびに音が大きくなるという、いわば階段状の変化を示します。ピアノやマリンバなどの減衰系楽器は、発音後に音量を上げることが物理的に不可能ですから V の遅延効果よりも v の遅延効果を利用の方が自然です。

また、音源や選択した楽器によって異なりますが、音量 V が単純なボリュームとしての変化に対し、強弱 v は波形レベルの変化が起ります。例えば生楽器のピアノを弱く弾いて録音し、それを大音量で再生しても強い音には感じられません。このように強弱 v は音そのものの表情に変化が起ります。これは波形の変化であり、この点は波形加工の効果が強弱 v によって左右されることと相関があります (2.3.17 波形を加工する (p.58) を参照のこと)。

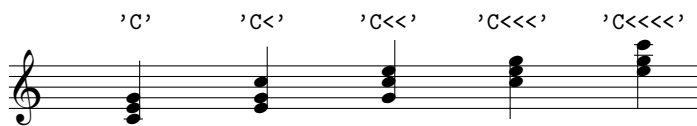
2.2.18 コードの転回形

旋律が音程の移り変りを表現するのに対し、コードは複数の音の混ざり具合、ブレンド感を表現します。したがってコードを進行させていく際は、あまり音域を変化させず、一定の音域に納めて旋律の邪魔をしない方が心地よいアンサンブルとなります。

Muse は旋律の音域にコードを混入させないようにするため、F・G・A・B の高音側をルート音とするコードは、1 オクターブ低い側にシフトさせていますが、これだけでは音域の跳躍を抑止することはできません。根本的にこの跳躍感を解決するためには、コード各音の並びを組替える“転回”という方法を取ります。

転回はコードネームの直後に < あるいは > 記号を添えて指定します。< でコード内の最低音を 1 オクターブ上げる上昇方向の組替えを行い、> でコード内の最高音を 1 オクターブ下げる下降方向の組替えを行います。また、同一記号を列挙することで転回操作の階数が指定できます。列挙は最大 4 つまでです。

上昇方向の転回例



以下に転回指定の例を示します。基本形の演奏では、伴奏コードが飛び跳ねていますが、転回を施すことでコード進行がスムーズになり、旋律と伴奏が互いの役割を任せて果すようになります。

```
%120
@D P1 V100 W0 R80 Q0
@A P49 V100 W0 R80 Q0

*TEXT"基本形のコード伴奏"
#D0 v120 o5 |f4l2f4 |m4c2m4 |r4c-r-2 |s2f |_1
#A0 v80 o4 |'F'1 |'Em'1 |'Gdim'1 |'Csus4'1 |_1

*TEXT"転回形のコード伴奏"
#D0 v120 o5 |f4l2f4 |m4c2m4 |r4c-r-2 |s2f |_1
#A0 v80 o4 |'F'1 |'Em>>'1 |'Gdim'1 |'Csus4>'1|_1
```

これらの部分で転回形を用いた

Muse でのコード記述はルート音の指定から始めて本節の転回の指定まで、7 つものパラメータがあります。省略可能なパラメータがほとんどなので、実際に記述する際はそれ程の困難もないのですが、あえて最も複雑な指定を行った場合を以下に示します（必須なのは、ルート音と音長のみ。音長も省略音長が可能）。

コードネーム テンション 音長

' C# dim >>> (+9,13) /Gb ' 2 :16

ルート音 転回 ベース音 アルペジオ
(分数コード)

なお転回の操作は、追加音であるテンションとベース音には影響を与えません。これらの追加音は、基本形におけるルート音からの相対音程で算出されます。

2.3 より高度な記述

2.3.1 コマンドによる制御

この節では、“コマンド”という概念の機能を紹介します。コマンドは*に続くコマンド名の後に、ダブルコーテーションでくくられたパラメータを添えるという形式を持っています。

* コマンド名 "パラメータ文字列"

パラメータ文字列は、キャンセル文字に対する唯一の例外領域です。スペース、全角文字、半角カタカナ、半角縦棒、セミコロン ;、感嘆符 ! も文字列として解釈されます。ただし改行とタブだけはキャンセルされます。

コマンドは以下の 12 種類が用意されています。始めの 4 つは、鍵盤上部に文字列が表示される機構を持っており、それぞれ表示色が異なります。

*HEAD	曲のタイトル表示	(青)	
*TEXT	文字列の表示	(茶)	2.3.2 (p.44)
*MARK	位置決め可能な文字列表示	(紺)	2.3.3 (p.45)
*STOP	一時停止と文字列表示	(赤)	2.3.4 (p.45)
*FONT	上記 4 コマンドの文字フォント指定		2.3.5 (p.46)
*WAVE	音声ファイルの再生		2.3.6 (p.46)
*COLR	メンバー色のカスタマイズ		2.3.10 (p.52)
*DRUM	ドラマーに関する転向や残響の制御		2.3.11 (p.53)、2.3.12 (p.53)
*ROOM	演奏会場の設定		2.3.13 (p.54)
*FING	フィンガー属性の一括指定		2.3.15 (p.56)
*DATA	エクスクルーシブのダイレクト記述		2.3.19 (p.59)
*POOL	上記エクスクルーシブの共通な前半部分を指定		2.3.19 (p.59)

HEAD コマンドは 1 つの Muse ファイルに唯 1 つ記述するもので^[15]、通常曲のタイトル等を記載します。この HEAD コマンドのパラメータ文字列は、ロード直後に表示エリアに描画されます。HEAD コマンドは、どの位置に書いても構いません。その例として以下に HEAD コマンドを記述しますが、この文字列が Readme.txt をロードした直後に表示されるはずですが、

*HEAD" Muse 文法解説のための演奏例 (Readme.txt 連動)" "

2.3.2 カラオケとして利用する

TEXT コマンドを打っておくと、そのタイミングで鍵盤上部にその文字を表示させることができます。文字列の表示色は茶色です。1 度表示した文字は次に本コマンドが指定されるまで維持されます。消去したい場合はパラメータを空にしたコマンドを与えて下さい。

*TEXT""

TEXT コマンドをうまく利用すれば、ちょっとしたカラオケ・システムが作れます。通常、音符の合間に TEXT コマンドを挟み込めば良いのですが、以下の例では、休符だけで構成された歌詞用フィンガーを設定しました。このように専用フィンガーを設定すると、後述するマクロ機構を活用すれば、1 番と 2 番で同一の旋律を繰り返しても、歌詞は 1 番・2 番で変えることもできます。

カラオケでは、ほんのちょっと早めに歌詞を出すのがコツです。このフレーズの指定には、音長の減算記号 ~ を利用すると調整が容易です。

静かな湖畔 (スイス民謡)

```
%120 \-
@JK P69 V127 S0 T0
#J0 o4 v127
f8.f16f8.s16 18.11618.116 s8.f16s8.116 f8.f16d4
18.11618.c16 <d8.d16d8.d16> c8.116c8.<d16> 14.<d8>
14.<d8> 14.<d8> 18.<d16>18.<d16> 12 _1

#K0 o3 v127

#K1 ; 歌詞専用のフィンガー
*TEXT" 静かな湖畔の 森かげ近く" _1_1~8 ; 8分音符早めに出す
*TEXT" 起きてはいかがと かつこが鳴く" _1_1
*TEXT" カッコー" _2
*TEXT" カッコー カッコー" _2
*TEXT" カッコー カッコー カッコー" _8^8.
*TEXT" カッコー カッコー カッコー カッコー" _8^8.
*TEXT" カッコー カッコー カッコー カッコー カッコー" _2^16
```

*TEXT で最後に表示した文字列は、演奏が終了しても表示したままになります。もし演奏終了時点で消去したければ *TEXT"" を最後に記述しておいて下さい。

2.3.3 楽曲中の位置決めを容易にする

複数の楽章や楽曲を 1 つの Muse データに納める場合、各曲の頭出しが容易に行えると便利です。前述の STOP コマンドを使用すると、シークバーの左右の三角ボタンで頭出しが可能になりますが、常に演奏がそこで一時停止してしまいます。

MARK コマンドは演奏を一時停止させずに、頭出しを可能にするコマンドです。文字列を表示する機構としては TEXT コマンドと同等ですが、表示色は紺色であり、STOP コマンドのようにシークバーの左右の三角ボタンでの位置決め対象になります^[16]。

```
; チャイコフスキー「弦楽四重奏曲第 1 番二長調」
*MARK "第 2 楽章 アンダンテ・カンタービレ"
%58 \--
@B P50 R127
#B0 c8<rr>c <m4r/.. d4>s8f<d>c f2/ f8rs4 r<d/.. d8r>ls/. f2 _2.
```

なお、上記の例はノンレガートな奏法指定を使ってフレージングを行っています。参考にして下さい。

2.3.4 途中で強制的に停止させる

STOP コマンドは、2 つの機能に大別されます。一つは、このコマンドに到達した時点で演奏が一時停止する機能。今一つは、このコマンドを記述した所を曲の最後と解釈し、そこで演奏を終了する機能です。この 2 つの区別は、パラメータに文字列内容を与えるか否かでいきます。

1. 一時停止の機能

```
*STOP "マウスをクリックして下さい"
```

この記述がある所で演奏が一時停止する。停止とともにパラメータ文字列が赤字で表示される。

2. 演奏終了の機能

```
*STOP ""
```

この記述がある所で演奏は終了する（この場合でもダブルコーテーションは必要です）。

前者の例は、シークバーの左右の三角ボタンをクリックすると、STOP ポイントに位置決めされるため、強力なデバッグ支援機構として活用できます。この例は、Readme.txt の各節表題で利用しています。参考にして下さい。

後者の例は、ある位置以降のデータを一時的に無効にしたい場合に活用できます。また、後ほど紹介するマクロと組み合わせることで威力を発揮します (2.3.8 演奏パターンの部品化 (p.49) を参照のこと)。

2.3.5 表示フォントを切替える

テキスト表示領域に文字列を表示させるコマンドは、HEAD・TEXT・MARK・STOP の 4 種類がありますが、これらの表示フォントはデフォルトで「MS P ゴシック」となっています。上記 4 種類のコマンドを発行する前に、FONT コマンドを発行することで、それらの表示フォントを変更させることができます。書式は以下の通りです。

```
*FONT" フォント名 , 修飾 "
```

この FONT コマンドが指定されると、新たな FONT コマンドが発行されるまで、指定されたフォントが維持され続けます。フォントには、以下のように太字と斜体を組み合わせた修飾が可能です。

```
0 : 標準  1 : 太字  2 : 斜体  3 : 太字 + 斜体
```

修飾を省略した場合は標準の修飾が採用されます。逆に、フォント名を省略し修飾だけを指定した場合は、直前のフォントに対して修飾を施した状態になります。フォント名と修飾の組み合わせで表現できる文字形状は、メニューの「機能 (G)」 「フォントの確認 (N)」にて表示イメージを確認することができます。

再生するパソコンに存在しないフォント名が指定されると、そこでの修飾指定も含めて無視され、その時点で有効となっていたフォント指定が維持されます。この特性を生かすと、再現環境に応じたフォントの優先度指定が可能となります。例えば以下のように連続したフォント指定を記述することで、特殊なフォントが存在する環境ではそのフォントを採用させ、存在しない場合は一般的なフォントを採用させることができます^[17]。

```
*FONT"Times New Roman"   一般的に存在するフォント
*FONT"HG 正楷書体-PRO"   少々特殊なフォント
*FONT"ほにゃ字"         かなり特殊なフォント
```

2.3.6 音声ファイルを再生させる

Muse は、任意のタイミングでオーディオ系のファイルを再生させる機構を持っています。通常、MIDI 音源と WAVE 音源は独立していますから、MIDI では表現できない効果音をミキシングすることが可能となります。コマンド名は WAVE で、パラメータには Wave ファイルに代表される音声を含むオーディオ系のファイル名を記述します^[18]。

ファイル名はパス記述が可能です。相対パスにする場合は、記述する Muse ファイル自体からの相対パスにします。しかしパス記述は、Muse データを配布することを考慮すると、使用を避けた方が無難です。Muse ファイルと音声ファイルは極力同一フォルダに配置し、ファイル名だけの指定にすることを推奨します^[19]。

なお、下記の例は曲の途中から演奏しているので V 指定を活用してフェードインを実現しています。メンバー宣言で E および F のメンバーを指定しているため、1ヶ所の V で、メンバー EF を同時に処理しています。参考にして下さい。

```
シヨパン「幻想即興曲」
%80 \=
@EF                ;   メンバー宣言 EF
P1 S0 T0 W0 R100 Q0 ;   メンバー EF に対し、一気に属性指定
#E0 o4 v127 V30 V127:2 ;   メンバー EF を同時にフェードイン
#F0 o4 v127 ?6
```

```
#E0 (lclclcl)8.s+16l8.c16|s8_<c4 & (c8lsm-d>l8.<s16>)2|
#F0 m8c<r+d>|mc<ms>|l<d-md>|

#E0 r2~16c16l8.s+16l8.c16|s4_1(<d>c1)|s2(lclcl)8s<mr>|
#F0 c8<m>smc<r+lr+|mscs rflf|>m<msc <r>csm>|

#E0 <m2s2|f4m[r]4~8(mrdr)8m8.d16|>s2l2|^2(c-<d>c-<d>c-)8l8<r8m8|
#F0 l<dml<d>lmd|cmlmsmlm|>msc<m>rfl<r>|fl+<df1+fm>f|

#E0 f4mrm|d4.>f+32s1s<m4.r8|r1|[d1>s1m1]2
#F0 c<r[m+d]s[r]c[ms]r|[>l<d]mlm[>l<r]msm|[>l<r]msc<r>csm[m1>l1]2|

*WAVE "Clap.mp3"
_1_1_1 ; WAVE 音が鳴っている間、休符を入れておく。
```

上記例では曲の終了時に拍手のウェーブ音を MP3 で入れています。最後の和音の長さは全音符ですが、和音自体の占有音長を 2 分音符にして音が鳴り終る前に拍手がかぶるように工夫しています（余談ですが、私は曲が最後まで鳴り終らない内に拍手をするのが嫌いです。ちゃんと最後の響きまで堪能すべきであり、鳴り終わってからの静寂も、少なくとも 1 小節分ぐらいは味わいたいものです。実際、Muse データでも最終音の余韻を響かすために最後に適度な休符を与えることをお勧めします）。また、拍手が鳴っている時間だけ最後に休符を入れて、Muse 演奏自体が終了しないようにしています。Muse 演奏が終了した時点（シークバーが右端に達した時点）で WAVE 音源も OFF になりますから、最後まで WAVE 音を鳴らしたい場合この最後の休符は必須です。WAVE 音を途中で強制的に止めたい場合は、パラメータなしのコマンド *WAVE"" を指定して下さい。

2.3.7 マクロで各種の繰返しに対応する

音楽には同じパターンが何度も出現することが多々あります。その様なパターンの入力を何度も何度も行うのは単調で苦痛な作業であり、かえって入力ミスも増えます。そこで Muse では、既に入力したパターンを中括弧 { } でくくって領域を定義し、その領域に任意の名前を付けることで、再度同じパターンを使用しなくなった場合に、その名前での入力だけで済ます機構を設けました。この様な機構は一般に“マクロ”と呼ばれています。ここでは、領域を指定するマクロ記述を“定義マクロ”と呼び、それを任意の位置に再現するマクロ記述を“展開マクロ”と呼ぶことにします。それぞれの書式は以下の通りです。

```
定義マクロ：$ マクロ名 {     ここに内容が入る     } 繰返回数
展開マクロ：${ マクロ名 } 繰返回数
```

マクロ名は自由な命名が可能です。ただし、スペース、タブ、改行、全角文字、半角カタカナ、半角縦棒 | の、キャンセル文字群は無視されます。セミコロン ; および感嘆符 ! は、コメント記述文字として解釈されるので使えません。

マクロは、今まで紹介したあらゆる指定記述とは本質的にレベルが異なり、演奏時の音出しや文字表示を指示するものではありません。あくまでも、定義マクロの領域内に入っている記述が展開マクロの位置に、展開マクロで指定した繰返し回数分、あたかも存在するかのように解釈されるだけです。

定義マクロで指定される“繰返回数”は、マクロの定義域に含まれません。あくまでも、中括弧 { } で囲まれた内容が、展開マクロ側で利用されます。なお、定義側も展開側も繰返し回数を省略すると 1 と見なします。

一般にマクロというと、定義マクロ部分は実行されず定義に徹するのが普通ですが、Muse では、定義マクロ部分も通常の記述部分と同様に演奏されます。この思想により、1 度書いた部分を気軽にマクロ定義でくくっていただけます。逆に定義マクロを定義のみとし演奏させたくない場合については、次節で紹介する演奏パターンの部品化を参照して下さい。では講釈はこの位にして、とにかく記述例を示してみることになります。

クラリネットをこわしちゃった (フランス童謡)

```
%280 \-
```

```
@K V080 P49 S0 T0 Q00 R40 W0
```

```
@M V120 P70 S0 T0 Q10 R60 W0
```

```
@N V120 P72 S0 T0 Q20 R60 W0
```

```
@Z P1
```

```
#K0 o6 v80
```

```
#M0 o4 v80
```

```
#N0 o4 v120
```

```
*TEXT"ここから、冒頭のマクロの定義開始です"
```

```
$(Macro1{
```

```
  #K0|'F'1&|'F'  |'F'&  |'F'2_4|
```

```
    |'F'1&|'F'  |'F'&  |'F'2_4|
```

```
  #M0|f4fff|fffs|l2.s4 |f2d4_  |
```

```
    |f4fff|fffs|l2.s4 |f2d4_  |
```

```
  #N0|f4fff|fffs|l2.s4 |f2d4_  |
```

```
    |l4l1l1|l1lc|<d2.>c4|l2f4_  |
```

```
*TEXT"これは、セーニヨ部分のマクロ定義です"
```

```
$(Macro2{
```

```
  #K0|'F'1          |'D7>>'  |'Gm7'&  |'Gm7'2_          |
```

```
    |'C7>>'1&  |'C7>>'  |'F'&  |'F'2_          |
```

```
    |'F'1&  |'F'4_2.  |'F'1&  |'F'1          |
```

```
    |'F'1&  |'F'  |'D7>>'  |'Gm'4'D7>>\n1b''Gm'2|
```

```
    |'C7>>'1&  |'C7>>'&  |'C7>>'  |          |
```

```
  #M0|l2l4.//18// |f+4rmf+  |s2s4.//s8//|s2_          |
```

```
    |s4//ss4.//f8|m4drm  |f2f4.//f8//|f2_          |
```

```
    |l2.s4  |f4__  |l2.s4  |f4__<d>          |
```

```
    |l4l8l14<d> |l4l8l14<d>|<rd>c1  |s1cc          |
```

```
    |s4s8ss4c  |s4s8ss4c  |<d>c1s  |          |
```

```
  #N0|<d2d4.//d8//|r4d>c1  |c2c4.//c8//|c2_          |
```

```
    |c4//cc4.//c8|<d4>c1s  |l2l4.//18//|l2_          |
```

```
    |<d2.>c4  |l4__  |<d2.>c4  |l4__<d>          |
```

```
    |l4l8l14<d> |l4l8l14<d>|<rd>c1  |s1cc          |
```

```
    |s4s8ss4c  |s4s8ss4c  |<d>c1s  |          |
```

```
}
```

```
#K0 |'F'2_ |
```

```
#M0 |f1  |
```

```
#N0 |f1  |
```

```
}
```

```
*TEXT"冒頭のマクロを再現します"
```

```
$(Macro1}
```

```
*TEXT"マクロを抜けました"
```

```
#K0|'F'1&|'F'  |'F'&  |'F'  |
```

```
  |'F'1&|'F'  |'F'&  |'F'  |
```

```
  |'F'1&|'F'  |'F'&  |'F'  |
```

```
#M0|f4fff|ffff  |ffff  |fffs|
```

```
  |l2.s4|f2d  |f4fff  |ffff|
```

```
  |f4fff|fffs  |l2.s4  |f2d  |
```

```
#N0|f4fff|fff |fff |fffs|
|12.s4|f2d |14111 |1111|
|14111|111c |<d2.>c4|12f |
```

```
*TEXT"セーニョに戻ります"
```

```
#{Macro2}
```

```
*TEXT"最後に、コーダを演奏します"
```

```
#K0|'F'4_'C7>>'_|'F'__2|
```

```
#M0|f4//_<d>_ |<f>__2|
```

```
#N0|f4//_<d>_ |<f>__2|
```

```
*TEXT""_1'2
```

```
$Drum{ ; ドラムのマクロ (次節でも使用します)
```

```
#Z0 $Set{v60 [o4r+]0 ,4,8,,4,}17
v127_2o312 ${Set} v127_2o3d+2 ${Set}25
v127_2o2r+2 ${Set} v127_2o31+2 ${Set}8
o4 v127 [1+<1>]1 1+,1+ ,, 1+1+1+1+ [1+<1>],
${Set}9 v127_2o5s2 ${Set} v127_2o5f+2 ${Set}7 ,4_,_,__2
```

```
}
```

定義マクロの { } の中には、今まで出てきたあらゆる記述記号を包含できます。また、音符の連結 & までを記述しておき、展開マクロの次の音と連結させることも可能です。このようにマクロはかなり柔軟な記述ができるので、楽典に山の様に存在する繰返し記述（反復記号・1 カッコ・2 カッコ・セーニョ・ダルセーニョ・コーダ・ダカーポ・フィーネ等）を、これ一発で表現できます。また繰返し回数は、特にパーカッションで威力を発揮します。上記例は 1 小節分のパターンを \$Set で定義しておき、あとは必要な小節分回数で展開しています。

定義マクロは入れ子が何階層でも記述できます。例えば、上記の例で、\$Macro1 の中に、\$Macro2 が含まれていすし、ドラムのマクロ \$Drum の中にも、\$Set が含まれています。ただし、定義マクロの中にそれ自身の展開マクロを記述すると無限ループが生成するため、文法エラーとなります。

2.3.8 演奏パターンの部品化

マクロは繰返し記号への対応という使い方以外にも演奏パターンの部品化という活用もできます。あらかじめ旋律やリズムのデータを部品化し、それをマクロとして定義しておきます。そして、実際の演奏は展開マクロで行うという構成です。ただし、Muse のマクロはそのままの記述では実際の演奏を行ってしまいますので、部品として扱うには、その時点では定義するだけで演奏はしない、という方法が必要になってきます。この構造を実現するには 2 つの方法があります。

1. STOP コマンドを利用する方法

演奏させるデータの末尾に *STOP"" を記述し、そこで演奏を終了させるようにしておきます。そして、定義マクロは *STOP"" の後に記述するようにします。

2. 繰返し数ゼロを利用する方法

定義マクロの繰返し数にゼロを指定することで、定義マクロ自体は演奏しないため、定義マクロをどこにでも自由に記述することができるようになります。

以下は、前節の曲を *STOP"" を利用した部品構造で書き直した例です。

```
クラリネットをこわしちゃった (マクロを部品として利用)
```

```
%280 \-
```

```
@K V080 P49 S0 T0 Q00 R40 W0
```

```
@M V120 P70 S0 T0 Q10 R60 W0
```

```

@N V120 P72 S0 T0 Q20 R60 W0
#K0 o6 v80
#M0 o4 v80
#N0 o4 v120

*TEXT"部品形式で書き直しました（今演奏しているのは部品 1 です）"
${Part1}

*TEXT"ここで、部品 2 に入ります"
${Part2}
#K0 |'F'2_|
#M0 |f1 |
#N0 |f1 |

*TEXT"部品 1 に戻りました"
${Part1}

*TEXT"再度、部品 2 を演奏します"
${Part2}
#K0 |'F'2_|
#M0 |f1 |
#N0 |f1 |

*TEXT"次は、部品 3 です"
${Part3}

*TEXT"またしても、部品 2 に戻りました"
${Part2}

*TEXT"最後に、コードの部品 4"
${Part4}

${Drum} ; 前節で定義してあるドラムセットを再利用しています

_1 *TEXT"以上です"

*STOP"" ; ここで演奏終了（以下は部品データ）

; 部品 1
$Part1{
    #K0|'F'1&|'F' |'F'& |'F'2._4|
        |'F'1&|'F' |'F'& |'F'2._4|
    #M0|f4fff|fffs|12.s4 |f2d4_ |
        |f4fff|fffs|12.s4 |f2d4_ |
    #N0|f4fff|fffs|12.s4 |f2d4_ |
        |14111|111c|<d2.>c4|12f4_ |
}

; 部品 2
$Part2{
    #K0|'F'1          |'D7>>'      |'Gm7'&      |'Gm7'2_      |
        |'C7>>'1&      |'C7>>'      |'F'&        |'F'2_        |

```

```

|'F'1&      |'F'4_2.  |'F'1&      |'F'1      |
|'F'1&      |'F'      |'D7>>'     |'Gm'4'D7>>\n1b''Gm'2|
|'C7>>'1&   |'C7>>'&   |'C7>>'     |          |
#M0|1214.//18// |f+4rmf+  |s2s4.//s8//|s2_      |
|s4//ss4.//f8|m4drm  |f2f4.//f8//|f2_      |
|12.s4      |f4___    |12.s4      |f4__<d>   |
|1418114<d> |1418114<d>|<rd>c1     |slcc      |
|s4s8ss4c   |s4s8ss4c |<d>c1s     |          |
#N0|<d2d4.//d8//|r4d>c1  |c2c4.//c8//|c2_      |
|c4//cc4.//c8|<d4>c1s  |1214.//18//|12_      |
|<d2.>c4     |14___    |<d2.>c4     |14__<d>   |
|1418114<d> |1418114<d>|<rd>c1     |slcc      |
|s4s8ss4c   |s4s8ss4c |<d>c1s     |          |
}

; 部品 3
$Part3{
#K0|'F'1&|'F'      |'F'&      |'F'      |
|'F'1&|'F'      |'F'&      |'F'      |
|'F'1&|'F'      |'F'&      |'F'      |
#M0|f4fff|ffff     |ffff     |fffs|
|12.s4|f2d     |f4fff     |ffff|
|f4fff|fffs     |12.s4     |f2d |
#N0|f4fff|ffff     |ffff     |fffs|
|12.s4|f2d     |14111     |1111|
|14111|111c     |<d2.>c4|12f |
}

; 部品 4
$Part4{
#K0|'F'4_'C7>>'_|'F'__2|
#M0|f4//_<d>_   |<f>__2|
#N0|f4//_<d>_   |<f>__2|
}

```

2.1.19 再現表記の活用 (p.23) で紹介した “リズムパターンの分離” とゼロ音長を利用して、リズムのマクロ化を実現することもできます。定義マクロで、再現表記、だけで発音する和音やコードのないリズムだけを記述しておき、マクロを展開する際にゼロ音長で、発音する和音やコードを指定するという方法です。

1. STOP コマンドを利用する方法

```

'Cm'0 ${Rhythm}      ; Cマイナーのコードをリズムパターンで演奏
[dms]0 ${Rhythm}     ; ドミソの和音をリズムパターンで演奏
*STOP""              ; ここで演奏が停止する
${Rhythm}{,4,4,8,8,4} ; リズムパターン定義

```

2. 繰返し数ゼロを利用する方法


```

$Rhythm{,4,4,8,8,4}0 ; リズムパターン定義 (ここでは演奏しない)
' Cm'0 ${Rhythm}      ; Cマイナーのコードをリズムパターンで演奏
[dms]0 ${Rhythm}     ; ドミソの和音をリズムパターンで演奏

```

2.3.9 スコア譜とパート譜の混在

テンポ値が添えられていない単独の % 指定を利用すると、必然的にスコア風の記述をすることになります。しかし例えば楽曲部分だけをスコア譜で記述し、歌詞の部分はパート譜で記述したいといった場合も多々あります。このような場合は % の後にメンバー記号 (A~0, Z) を添えて下さい。添えられたメンバーに対してのみ、強制タイミング合わせの機構が働くようになります。これにより、スコア譜記述とパート譜記述を混在させることができます。

% に添えるメンバー記号は、複数列挙することが可能です (例: %ABC)。その場合、列挙したメンバー内で、最も先まで音符のある位置まで休符が挿入され、それらのメンバーだけが、その位置に整列します。以下は、ABC メンバーをスコア譜で、D メンバーをパート譜で記述する例です。

```

#A0 | α          |
#A1 | β          |
#B0 | γ          |

%ABC ; ここで ABC のタイミングを合わせる

#A0 | δ          |
#A1 | ε          |
#B0 | ζ          |

#D0 | η          |
#D0 | θ          | この部分はパート譜で記述

```

すると、メンバー ABC にだけに整列のための休符が与えられ、実際の演奏は以下のようになります。

```

                                     メンバー ABC はこの位置に整列。
#A0 | α          |   休符   | δ          |
#A1 | β          |           || ε          |
#B0 | γ          |   休符   | ζ          |
#D0 | η          | || θ          |
                                     メンバー D には強制休符は与えられない。

```

単独の % 指定は %ABCDEFGHIJKLMNOZ の省略形と見なすことができます。

2.3.10 メンバー色をカスタマイズする

Muse は初期状態で各メンバーにデフォルトの表示色 (メンバー色) が与えられていますが、COLR コマンドを活用するとこの各メンバー色を自分の好みの色に変えることができます。具体的な指定は、COLR コマンドのパラメータで行います。

```
*COLR" A(64a088) E(550c2d) "
```

上記の例は、メンバー A と E に対して初期の色を変更する記述です。このようにメンバー記号 (A~0, Z) に括弧でくくられた色コードを添えた記述を、カスタマイズしたいメンバー分、列挙していくという形式で指定します。色コードの指定は、RGB の各輝度を 00~ff までの 16 進数表記で行います。この 6 桁が色コードで、これはホームページ作成などで使用される HTML の指定方法と同様です。上記例のメンバー A は、赤が 64 の輝度、緑が a0 の輝度、青が 88 の輝度を指定しています。パラメータ中に同じメンバーの指定がなされた場合は、最後に指定した色コー

ドが採用されます。また、タブやスペース・改行は任意に挿入できます。更に、メンバー ID は列挙が可能であり、一気に複数のメンバーを同一色にすることができます。

```
*COLR" ABCD(28a9bb)
      K(b08a74)
      IHF(006ce5) "
```

この COLR コマンドは、1 つの Muse ファイルに唯一つ記述します。記述する位置はどこでも構いませんが、その設定は曲頭から効果します。

2.3.11 ドラマーの転向について

ドラムセットはそのメンバーの音色 P で指定できますが、ドラムを担うメンバーが Z メンバー 1 人では、同時に異なるドラムセットを鳴らすことができません。そのためには、もう 1 人ドラマーをアサインする必要があります。このような場合に DRUM コマンドを利用すると、A~0 までのメンバーの内、誰か 1 人をドラマーとして転向させることができます。

この指定は DRUM コマンドのパラメータに、転向させるメンバーを与えて行います。与えられるメンバー記号は A~0 の 15 メンバーの内いずれか 1 つです。例えば、メンバー Hさんをドラマーに転向させる場合は、以下のように記述します。

```
*DRUM "H"
```

DRUM コマンドも、HEAD コマンドと同様に、1 つの Muse ファイルに唯一つ記述します。ファイル中のどの位置に書いても構いませんが、複数記述することはできません。このことは、ドラマー転向に関して次の制約があることを表しています。

- ドラマーへの転向は 1 人だけで、しかも 1 回だけである。
- 演奏の途中で、楽器を持替えるように転向させることはできない。

どの位置に DRUM コマンドを記述したとしても、指定メンバーは最初からドラマーとして演奏に参加します。

また、この DRUM コマンドを使用すると、上記とは逆にドラマー専任の Z メンバーを、A~0 メンバーと同様のメロディー奏者に転向させることができます。その記述は、パラメータに Z- を与えることで行います。

```
*DRUM "Z-"
```

以上の 2 つの指定を併記する事もできます。下記は H メンバーをドラマーにし、Z メンバーをメロディー奏者にする例です。

```
*DRUM "H Z-"
```

2.3.12 ドラムロールの響きを向上させる

スネアロールやシンバルロールなどを演奏させると、まるで機関銃を放ったかのようなブツブツした音になってしまう場合があります。これは、同音連打に対する音源の設定によります。同音連打とは、同一音程の音を連続して鳴らすことを言い、ドラムでは、ロール奏法としてよく出現します。通常 MIDI 音源では、同時発音数不足を警戒し、特にドラムメンバーに対して同音連打の際に直ぐに前の音を止めてしまう設定になっています。このためにブツブツした機関銃音になってしまうのです。

既に 2.3.11 ドラマーの転向について (p.53) でご紹介したドラム転向コマンド *DRUM を使って、このロール残響の制御を行うことができます。ロール残響は、0, 1, 2 の 3 つのレベルで指定します。

- 0 : 同一音の重なりを一切認めないレベル (ドラムメンバー Z のデフォルト)
- 1 : 適度な重なりを認め、同時発音数不足もそれなりに押えるレベル
- 2 : 完全に重なりを認めるレベル

これらのレベル値を、*DRUM のパラメータで記述したドラムメンバー記号に添えることで響きの制御を行います。例えばドラム転向させた H メンバーをレベル 2 の響きに指定したい場合は、以下のように記述します。

```
*DRUM "H2"
```

ドラム専属の Z メンバーにもこの指定を行いたい場合は、パラメータ部分に同一形式で列挙記述します。もちろん Z メンバー単独での指定も可能です。

```
*DRUM "H2 Z1"      ドラム転向とロール残響を一気に指定
*DRUM "Z2"         Z メンバーにロール残響を指定
```

なお、レベル値を添えない場合は、音源自体のデフォルトのレベルが採用されます^[20]。

2.3.13 演奏会場を設定する

ROOM コマンドを活用すると、演奏を行う会場の音場空間を調整することができます。音場とは、演奏会場の広さ、壁面の反響度合、反射音のうねりや減衰など、音が発せられて以降の空間内の響き具合を指します。これらの指定は、以下の書式を用いて行います。この ROOM コマンドは、1 つの Muse ファイルに唯 1 つ記述します。記述する位置はどこでも構いませんが、その設定は曲頭から効果します。

```
*ROOM" R  $\alpha$  ( $\beta$ ,  $\gamma$ ,  $\delta$ ,  $\epsilon$ ,  $\zeta$ ,  $\eta$ ) Q  $\theta$  ( $\iota$ ,  $\kappa$ ,  $\lambda$ ,  $\mu$ ,  $\nu$ ,  $\xi$ ,  $o$ ) "
```

R に添えられた値 α とそれに続く括弧でくられた 6 つの値 $\beta \sim \eta$ は残響に関する指定です。また、Q に添えられた値 θ と括弧内の 7 つの値 $\iota \sim o$ はコーラスに関する指定です。R のセットと Q のセットはどちらを先に書いても構いません。これら合計 15 個の値の意味を以下に示します。

α . 残響パターン値

0~7 の数字であらかじめ準備されているパラメータの組合せを選びます。会場の壁に当って戻る音や会場空間内に残る音に対する性質を決めます。続く括弧内の値を大きくすると、以下のような傾向を示します (β を除く)。値の範囲は β, γ が 0~7 で、 $\delta \sim \eta$ が 0~127 です。

- β . Type 響きのタイプを 0~7 の値で指定する。反響アルゴリズムの指定。
- γ . LowPass 壁面における高音の吸収率が上がり、丸みのある残響となる。
- δ . Level 残響の音量が上がる。
- ϵ . Time 残響が長く残る。
- ζ . Feedback 反響の繰返し回数が増える。
- η . Delay 音の反射時間がかかり、広い会場の表現となる。

θ . コーラスパターン値

0~7 の数字であらかじめ準備されているパラメータの組合せを選びます。会場空間内で混ざり合ったり、重なり合ったりする音の性質を決めます。続く括弧内の値を大きくすると、以下のような傾向を示します。値の範囲は ι が 0~7 で、 $\kappa \sim o$ が 0~127 です。

- ι . LowPass 高音域が減衰し、丸みのあるコーラスとなる。
- κ . Level 重なり合う音の音量が上がる。
- λ . Feedback 音を重ねていく量が増える。
- μ . Delay 重なり合う音のタイミングがずれうねりを生成する。
- ν . Rate うねりの周期が速くなる。
- ξ . Depth うねりの振幅が深くなる。
- o . SendRvb コーラス音の残響が増す。

ROOM コマンドの設定は 2.2.11 楽器の音色をアレンジする (p.37) で解説をした、残響 R とコーラス Q の値にそれぞれ乗ずる形で効果が出ます。したがって、R0 Q0 の状態では、まったく効果が現れません。逆に R や Q の値を調節すれば曲中でも音場の効果量に変化を与えることができます。また、メンバー毎に効果量のバランスを取るといった記述や、遅延効果を活用するという記述も可能となります。

ROOM コマンドは 15 個ものパラメータを持ちますが、それぞれ省略が可能です。括弧内のパラメータを省略すると、 α や θ で指定したパターン値が採用されます。パターン値そのものが省略された場合は R4 (HALL-2)・Q2 (CHORUS-3) がデフォルト値として採用されます。

```
*ROOM" R3(4,0,64,64,0,0) Q2(0,64,8,80,3,19,0) "
*ROOM" R3(4,0,64,64,0,0) "      コーラス設定を省略
*ROOM" R3(4,0,64,,0) "        Time と Feedback を省略
*ROOM" R3(4,0,64) "          Time 以降を省略
*ROOM" R3() "                6 つのパラメータすべてを省略
*ROOM" R3 "                  同上
*ROOM" R(1,3,,54,,108) "     残響パターン値を省略 (R4 と見なす)
```

「演奏会場の設定ダイアログ」にて Fix チェックが行われている場合は、本コマンドは効果せず、ダイアログ上の設定が優先されます。

2.3.14 移調楽器をフィンガー毎に割り当てる

様々な移調楽器が登場するオーケストラやブラスバンドのスコアを、実音に読み替えることなくそのまま Musing したい場合、調性 \ と移調 T を適切に組み合わせる必要があります。しかしこれらの指定は全域属性やメンバー属性であるため、1 つのメンバー内に異なる移調楽器を記述したい場合は、フィンガー ID を切り替える度に調性 \ と移調 T を指定し直さなければなりません。そんな局面で便利なのが移調楽器 y コマンドです。このコマンドはフィンガー属性であり、フィンガー毎に異なる移調楽器を指定できます。そして 1 回指定すれば、次に指定し直すまで効果が持続します。

このコマンドは、移調楽器の指定を簡便に行うために“調性”と“移調”を一括で行う形式になっています。y に続けて +, - を必要な数だけ添えて調を指定します (この部分の書式は調性 \ と同じです)。更に続けて、スラッシュを添えて -95 ~ +95 の数値を記述します。マイナスで低音側へ、プラスで高音側へそれぞれ半音単位にシフトします (この部分の書式は移調 T と同じです)。

```
y++++/-5
       $\beta$ . 移調 (T-5) の指定と同等
       $\alpha$ . 調性 (\++++) の指定と同等
```

実際の Musing イメージは、以下のようになります。

```
@A P72 ; クラリネット
#A0 y---/-3 d r m d r m ;   A 管
#A1 y+/-2 s l c s l c ;    Bb 管
#A2 y+++/-3 s l c s l c ;  Eb 管
```

y コマンドを記述したフィンガーでは、調性 \ や移調 T よりも優先してその値が採用されます。逆に上記 α や β を省略すると、調性 \ や移調 T が有効となります。

```
y/-5      調性は全域属性 \ が採用される
y++++/    移調はメンバー属性 T が採用される (スラッシュは省略可能)
y/        調性も移調も \ や T が採用される (スラッシュは省略可能)
```

調性 \ については、2.1.4 (p.14) および 2.2.14 (p.39) を、移調 T については、2.2.10 (p.36) を参照して下さい。

2.3.15 フィンガー属性を一気に指定する

楽章毎に別けて作成したデータを、1つの組曲や交響曲として編成しようとしたり、複数の楽曲を1つのアルバムとして綴じようとする場合、いくつかの Muse データを束ねて1つの Muse データにすれば良いわけですが、その際、以下の点に気を付けなければなりません。

1. HEAD, COLR, DRUM, ROOM コマンドが複数出現しないようにする。
2. マクロ名が重複しないようにする。2.3.7 (p.47) 参照
3. 連結する曲間で各属性の初期化を行う。

特に3におけるフィンガー属性の初期化は、16メンバー×10フィンガー、最大160のフィンガーに対して行わなければなりません。この手間を省くため、FING コマンドがあります。この指定は FING コマンドのパラメータに ?, x, o, v, w, p, q, y のフィンガー属性を与えることで、全メンバー・全フィンガーのフィンガー属性を、一括して指定することができます。パラメータ内に記述する属性は、どのような順番でも構いませんし、その中の一部のコマンドだけを指定することも可能です。

```
#A0 ?0 x0 o4 v100 w0 p0 q0 y/
#A1 ?0 x0 o4 v100 w0 p0 q0 y/
#A2 ?0 x0 o4 v100 w0 p0 q0 y/
:
:
#B0 ?0 x0 o4 v100 w0 p0 q0 y/
#B1 ?0 x0 o4 v100 w0 p0 q0 y/
#B2 ?0 x0 o4 v100 w0 p0 q0 y/
:
:
```

*FING "?0 x0 o4 v100 w0 p0 q0 y/"
(これ一発で OK)

またこのコマンドは、英米系や独系で音名を記述する方にとって、各フィンガーに逐一 x1 や x2 の指定をする手間を省きます。例えばデータの冒頭で以下の1行を記述すれば、再度指定のない限り、以降すべてのフィンガーで英米系音名を利用できます。

```
*FING "x1"
```

FING コマンドはデータ中の任意のタイミングで何度でも発行できます。記述しない属性はその時点の値が維持されます。FING コマンド中に v+10 などの相対強弱指定をした場合は、その時点の各フィンガー強弱値を基準にして個々に相対変化が起ります。

FING 指定内のアクセント w に関しては、それ以降効果させるアクセントの値をセットさせる機能として働きます。FING 記述直後の音符に直接アクセントを与えることはありません。

以下のように FING コマンドとメンバー属性の一括指定を活用すると、数行ですべての属性を初期化することができます。これは別々に作成された複数の Muse データを連結したい時に便利です。

```
%125 \= *FING "?0 x0 o4 v100 w0 p0 q0 y/"
@ABCDEFGHIJKLMNOZ
P1/0 V127 S0 U0 Y0 T0 R40 W0 Q0 R=64.64.64 W=64.64.64 Q=64.64
```

上記例示の音色指定 P1 に添えたスラッシュ・ゼロの意味については、2.3.16 音色バリエーションを指定する (p.57) を参照して下さい。

全メンバー・全フィンガーに対して一括指定をするのではなく、特定メンバー内のフィンガーにのみセットしたい場合は、メンバー ID に続く括弧内にフィンガー属性を記述します。メンバー ID は列挙して記述できます。

```
*FING " x1 ABC(v40) Z(w+20 q+10) "
```

Z メンバー内フィンガーに w+20 と q+10 をセット
ABC メンバー内のフィンガーに v40 をセット
全てのフィンガーに x1 をセット

2.3.16 音色バリエーションを指定する

試聴画面にある 128 の音色をベースにして、各音色毎に派生的な音色を指定することができます。これをバリエーションと呼びます。バリエーションの指定方法は、音色のメンバー属性 P の後にスラッシュを添え、更に 0~127 のバリエーション番号を与えることで行います。バリエーションを添えない場合、たとえ楽器 P を持替えたとしてもバリエーション番号は以前の値が採用されます。標準に戻したい場合は /0 を明示的に添えて下さい。以下に、いくつかのバリエーション例を示します。

Draw オルガン P17 の場合
P17 / 1 厳粛なオルガン
P17 / 8 ディチューンしたオルガン
P17 /16 イタリア製の 60 年代風オルガン
P17 /24 濁りのあるオルガン

ナイロンギター P25 の場合
P25 / 8 ウクレレ
P25 /16 フレットノイズ付のギター
P25 /32 バリエーションの効いたギター

三味線 P107 の場合
P107/ 1 津軽三味線

琴 P108 の場合
P108/ 8 大正琴
P108/16 カヌーン

和太鼓 P117 の場合
P117/ 8 大太鼓

小鳥 P124 の場合
P124/ 1 犬ワン
P124/ 2 奔馬
P124/ 3 ピョッ
P124/ 4 猫にゃ〜ん
P124/ 5 ヴォ〜ン

上記のようにバリエーション番号は楽器毎に歯抜けで存在しており、また音源のモデルによって存在したりしなかったりする番号もあります^[21]。明確な規則性もないため、ここでそのすべてを紹介することはできません。保有の音源マニュアルやヘルプファイルを参照して下さい。バリエーションは、MIDI 用語では“バンクセレクト”と呼ばれており、GS 音源の場合は Bank MSB で、XG 音源の場合は Bank LSB で解説がされています。

楽器の試聴ウィンドウは、音の定義がなされている /1, /8, /9, /16, /24, /32 の 6 個のバリエーションが選べるようになっています。ただし、すべての楽器でこの 6 個のバリエーションが可能とは限りません。指定できるバリエーションは楽器毎に異なり、多分に歯抜けになっています。Muse は、存在しないバリエーション番号を指定すると標準バリエーション /0 の音色を採用します。ただし、音源によってはこの標準化機構が働かず未定義バリエーションは発音しない機種があります。

逆に、高級な音源では上記 6 個以外のバリエーションも存在し、組合わせて千を超える音色の出せるものもあります。それらの音色は実際の Muse コーディングで確認して下さい。なお、バリエーションはドラムメンバーに指定しても効果しません。

2.3.17 波形を加工する

Muse は、音の波形そのものを加工する指定ができます。この波形加工は音楽的機構と言うよりは楽器の製作に近く、音そのものを作り込むシンセサイザーの領域です。加工ができる特性は以下の 3 種類であり、それぞれ 2~3 個の値を持っています。指定の書式は、残響 R, 揺らぎ W, コーラス Q のメンバー属性記号にイコールを添え、その右辺にピリオドで区切った値を列挙します。各値のデフォルト（標準）はすべて 64 であり、0~127 の範囲で指定します。

1. 出力変位特性 R = 値 1. 値 2. 値 3
発音されてから停止（残響）までの、時間的な出力レベル変位を制御する。
2. 変調特性 W = 値 1. 値 2. 値 3
揺らぎ（ビブラート）の強さや周期などの制御を行う。
3. 周波数特性 Q = 値 1. 値 2
倍音の発生量制御や、特定周波数帯の共鳴の調整を行う。

これら 3 つの特性指定はメンバー属性の一種であり、直前のメンバー宣言 @ で、それらが効果するメンバーが規定されるのは、通常のメンバー属性と同様です。したがって、メンバー単位の制御が可能です。ただし属性記号は同じでも、通常の R, W, Q とは完全に独立の機構です。

指定する右辺の各値は独立に省略が可能であり、省略すると以前の値を保持します。ただし省略時も区切りのピリオドは規定数だけ記述しなければなりません。

```
Q = 25.          値 2 を省略
W = .100.       値 1 と値 3 を省略
R = 20. .50     値 2 だけ省略
```

各特性に与えるそれぞれの値が、どのような性質を持っているかの詳細は、3.3 波形加工 (p.69) を参照して下さい。

この特性指定には“遅延効果”の記述が可能です。これは、上記記述の後にコロンを添え、そこに音長を与えることで表現します。例えば 2 分音符分の時間を経て、ある特性値に向けて連続的に波形を変化させる場合、以下のように記述します。周波数特性の場合は 2 つの値を持っていますが、それぞれの値が現在の値から指定された値に向かって、独立にかつ一括して変化します。

```
Q=10.40:2
```

なお、音の変化確認を容易にするため、波形のイメージ図を添えた確認ダイアログを用意してあります。「楽器の試聴」や「ドラムの試聴」の右下にある [波形] ボタンで起動する「波形の加工」ダイアログを利用して下さい。また、波形加工の効果は、強弱 v によって大きく変化します。本ダイアログには強弱調整のスクロールバーも付いていますので、その差を実音にて確認して下さい。本ダイアログ上で発音を行うと該当の特性パラメータ文字列がクリップボードにセットされます。データ作成中にこの機能を使うと、試聴した波形パラメータを簡便に入力できません^[22]。

2.3.18 コントロールの指定

Muse は X 指定により、MIDI 音源のショートメッセージをダイレクトに記述する方法が備わっています。X に機能を表現するコントロール番号を添え、その後をイコールで区切って、効果の度合いを表現するコントロール値を与えます。

```
X7=95      コントロール番号 7 に、コントロール値 95 を与えた。
```

コントロール番号は、0~129 まで準備されています。

前半の 0~127 は、MIDI 規格で定められている番号と同一の数字です。これら番号の意味や機能に関しては、MIDI に関する書籍やマニュアル、該当サイトなどを参照して下さい。イコールの右側に指定するコントロール値の範囲は 0~127 ですが、その意味は機能によって異なります。なお、X0~X127 のコントロールに関しては、遅延指定が可能です。他の遅延指定と同様、末尾にコロンと音長を添えて指定します。例えば以下の様に記述すると、2 分音符分の時間を掛けてボリュームが 100 から 60 に変化します。

```
X7=100 X7=60:2
```

後半の X128 と X129 は Muse で割り付けた番号で、特殊なコントロール指定になります。

- アフタータッチ X128

既に鳴っている音に対して指定することで、更に効果を与える機能です。その効果がどのようなものかは、音源によって様々に定義されています。アフタータッチには、2 つの指定方法があります。

```
X128=40          指定メンバー全体に作用 (チャンネルプレッシャー)
```

```
X128=40.65      指定メンバーの内、特定の音程に作用 (キープレッシャー)
```

1 つ目は @ によるメンバー宣言で指定されたメンバーに 40 の強さのプレッシャーを与える例です。2 つ目は、更にその内の “65 の音程” の音のみにプレッシャーを与える例です。この “65 の音程” という数字表現の音程は、MIDI 用語では “ノート番号” と呼ばれており、60 を中央のドとして、半音単位に 1 ずつ増減させることで音程を指定します。高い音ほど大きな値を取ります。

一部の音源ではアフタータッチの初期状態が無効となっています。これを有効にするには、エクスクルーシブというロングメッセージを音源に送出する必要があります。この送出方法に関しては次節を参照して下さい。

- ピッチベンド X129

U 指定と同じ機能ですが X 指定では、より細かな微分音程を指定できます。与えられるコントロール値は、-8192~+8191 の範囲を持ちます。0 の指定以外は、+、- の符号を添えて記述します。

```
X129=+5432
```

Muse デフォルトの ±2 オクターブのピッチ範囲で、16383 の分解能を持つことになり、約 0.3 セントという肌理の細かい指定が可能となります。

X 指定の機能は、今までのメンバー属性 PVRQ.....などを完全に包含しています。しかし X 指定での記述は「フィンガー情報」や「メンバー情報」、「譜面モニタ属性表示」などには反映されません。X 指定は、音源の性能をより引き出したい方のための一種裏技的な機能であることをご承知おき下さい。

2.3.19 エクスクルーシブの指定

Muse は、DATA コマンドにより、MIDI 音源のロングメッセージをダイレクトに記述する方法が備わっていません^[23]。*DATA のパラメータに、16 進コードをカンマ区切りで列挙することでエクスクルーシブを表現します。16 進コード並びの様々な用法は、MIDI に関する書籍やマニュアル、インターネットなどを参照して下さい。

エクスクルーシブは、F0 で始まり F7 で終わるデータ列です。DATA コマンドでは、この始末端のコードを自動付与しますので、F0 と F7 のコードは記述しないで下さい。例えば以下は、XG 音源の初期化エクスクルーシブの記述です。

```
*DATA "43, 10, 4C, 00, 00, 7E, 00"
```

パラメータ内のタブや改行、半角スペースや全角スペースなどは無視されます。また、無駄なカンマが連続した場合、最小限のカンマに解釈されます。

```
*DATA " , , 43 , , 10 , , , 4C , 00 , , , , 00 , 7E , , , 00 , "
```

```
*DATA " 43 , 10 , 4C , 00 , 00 , 7E , 00 "
```


チェックサムの自動計算機構も備わっています。総計の対象となるデータを括弧でくくることで、閉じ括弧の次にチェックサムが挿入されるようになっています。以下は、GS 音源の初期化エクスクルーシブの記述例です。なお、括弧前後のカンマは付けても付けなくても構いません。

```
*DATA "41, 10, 42, 12 (40, 00, 7F, 00)"
```

```
*DATA "41, 10, 42, 12, 40, 00, 7F, 00, 41"
```

Muse のメンバーという概念は、エクスクルーシブでは「チャンネル」という 16 進数の番号で扱われます。Muse のメンバー チャンネル番号の対応関係は以下の通りです。

GS 系の場合

A	1	B	2	C	3	D	4	E	5	F	6	G	7	H	8
I	9	J	A	K	B	L	C	M	D	N	E	O	F	Z	0

XG 系の場合

A	0	B	1	C	2	D	3	E	4	F	5	G	6	H	7
I	8	J	A	K	B	L	C	M	D	N	E	O	F	Z	9

エクスクルーシブを記述していると、その前半の数値はすべて音源固有の値であり同じものを何度も記述していることに気づきます。この入力を簡素化するために、POOL コマンドがあります。パラメータの書式は DATA と同一です。POOL コマンドで前半部のデータを指定しておく、続く DATA コマンドでは後半部からの記述だけで済むようになります^[24]。指定した POOL コマンドの内容は、次に POOL コマンドが発行されるまで維持されます。内容をクリアしたい場合はパラメータが空の *POOL"" を記述して下さい。以下に POOL コマンドを使用した記述例を示します。

```
*POOL" 41,10,42,12 "      POOL コマンドで前半部を定義
```

```
*DATA" (40,00,7F,00) "    *DATA"41,10,42,12,(40,00,7F,00)"
```

```
*DATA" (40,01,30,03) "    と同等となる。
```

```
*DATA" (40,01,34,46) "    .
```

```
*DATA" (40,01,33,48) "    .
```

```
*DATA" (40,02,00,00) "    以下同様
```

```
*DATA" (40,02,02,40) "    .
```

```
*DATA" (40,02,03,00) "    .
```

演奏を開始する際、Muse は必要最低限の音源初期化のメッセージを送出しています。しかし、DATA コマンドを使って自ら肌理細かく音源初期化を記述したい場合、この自動送出手はかえって邪魔になります。そのような場合は、パラメータを空にした、*DATA"" を記述して下さい。この記述がある場合、演奏開始時に自動的に送っていたほとんどのメッセージ送信を行わなくなります。具体的には、GS リセットを出力しなくなり、DRUM や ROOM コマンドも無効となります。また、ピッチベンドのレンジ設定も行ないません。なお、*DATA"" は、1 つの Muse ファイルに唯 1 つ記述します。記述する位置はどこでも構いませんが、複数記述することはできません。

2.4 補足

2.4.1 コマンドの実行タイミングについて

* コマンドは、マクロのような比較的上位レベルの指定に誤解されがちですが、実はかなり下位レベルにあり、その実行されるタイミングはフィンガー宣言 # に依存しています。すなわち、音符と同等のレベルにあります。* コマンドを記述する時は、どのフィンガーに対して書いているのかを意識して下さい。タイミング上は* コマンドは音長ゼロの一種の音符であると考えた方が良いでしょう。その意味では 2.2.6 メンバー単位に指定する値 (p.32) で説明した、メンバー宣言やメンバー属性も実行されるタイミングは、任意のフィンガーに依存する訳ですから、音長ゼロの特殊な音符と解釈することができます。この考え方は % によるテンポ指定でも当てはまります。テンポを変化させるタイミングは、やはりフィンガー内の記述位置に依存しています。

2.4.2 メンバー色の表示優先度について

鍵盤の色は、その音階の音を発生した瞬間に表示され、音が消えた時に戻ります。では、同時に複数のメンバーが同じ音を発生した場合は、一体どのメンバー色で表示されるのでしょうか？ それは、Muse ファイルの記述順番に依存します。いくら同時といっても、Muse ファイル上は 1 メンバーずつ記述する訳ですから、順番があります。表示はその順番で行われるため、一番最後のメンバー色が結果として残ることになります。この原理は譜面モタでの重ね書きにおいても適用されます。従ってできるだけ前面に表示したい色のメンバーは、Muse ファイル上では一番後に記述するのがコツです^[25]。

2.4.3 連符・和音・コードの入れ子関係

連符や和音は複数の音を列挙する記述形式を持っています。そして、その音の集合体が 1 つの占有音長を持っているので、それ自身 1 つの音であるとも解釈できます。同様に、コード記述も一度に複数の音が鳴りますが占有音長という概念から 1 音と見なせます。したがって連符やコードという 1 つの音が和音の中に記述されたり、和音やコードという 1 つの音が連符の中に記述できて良いはずですが、Muse は考え得るこれらのハイブリッドな記述に対応していません。

連符の中に各タイプの音を並べた例
 ([dms] [df1] [cls] d r m 'C#m7')4

和音の中に各タイプの音を並べた例
 [d8 r2 m4 (mf<s>)16 'Eb'1]4

ハイブリッドな和音記述が可能なことで、Muse でサポートしているコードに音を重ねて、より複雑なコードパターンを比較的容易に作成できます。ただし、連符同士の入れ子、和音同士の入れ子はできません。

(d r m (df1)) ; NG
 [d r m [df1]] ; NG

和音同士の入れ子は論理的に意味がありませんが、連符同士の入れ子は譜面に記載されている場合があります。その場合は次のような記述で対応して下さい。例えば、6 連符の最初の 1 音が 3 連符であった場合、Muse 記法で強引に記述すると以下ようになります。

((α α α) β β β β β)

しかし、先程述べたように Muse では連符の入れ子はできません。そこで、 α と β の音長比を整数で求めます。上記の場合は、 α が 1 に対して β は 3 となります。ここで、連符内の音長は、実音長ではなく単にその連符内の配分比

率を表現するだけであるという性質を利用します (2.1.11 連符の表現 (p.17) 参照)。この性質を利用すれば、上記の連符の入れ子を外すことができます。

$$(\alpha_1 \alpha_1 \alpha_1 \beta_1^{1^1} \beta_1^{1^1} \beta_1^{1^1} \beta_1^{1^1} \beta_1^{1^1})$$

更に乗算記号と省略音長の継承を利用すると、以下のように簡素に記述できます。

$$(\alpha_1 \alpha \alpha \beta_1^3 \beta \beta \beta)$$

2.4.4 メンバー属性の競合について

Muse は、メンバー属性をどのフィンガーからでも指定することができるため、複数のフィンガーを使って同一時刻に異なるメンバー値を記述することができます。しかし同一のメンバー属性は、1メンバー内で一度に1つの値しか取り得ません。

```
#A0 d r m @V30 d r m
#A1 m f s @V90 m f s
```

ここで競合が起きている (実際には一方の値しか取れない)

このような記述を“メンバー属性の競合”と言います。Muse では、競合が起きた場合データ上、後に記述された側の属性値を優先します。上の例では @V90 の方が効果します。また、フィンガー情報のダイアログの右端欄に表示されている音量 V の値は、時刻 (タイミング) 上の最終値ではなく、データ記述上の最後の属性値となります。下記の例を参照して下さい。

```
#A0      d r m @V30 d r m ;      時間的には V30 が最終値であるが、
#A1 @V90 m f s      m f s ;      ダイアログには V90 が表示される
```

更に、遅延効果における開始の属性値は、その時点のタイミング上の音量ではなく、このダイアログ上の音量が採用されるので注意が必要です。このように、複数のフィンガーに同一のメンバー属性を記述すると、競合を起したり遅延効果の制御が把握しづらくなるので、あえて使用する場合はご留意下さい。

2.4.5 エクスポート MIDI ファイルの定格性

通常の MIDI 形式でテンポ変化 (リタルダンドやアツチェレランド) を行おうとすると、階段状の変位でしか扱えません。階段の幅を細かくして誤魔化すことはできますが、厳密に言うと線形変位ではありません。そこで Muse は、本当になめらかな速度変化を実現するため画一的な MIDI 出力を行わず、数値計算によるテンポ制御を採用しています。これにより、人間的にテンポを揺らがせる演奏を行うことができる訳です。しかしそのための代償として、Muse から出力された MIDI ファイルは、他のシーケンサーや譜面変換ソフトでは処理しきれない端数音符が多く出現してしまいます。この節では、この現象を一時的に回避するための手段をご紹介します。

それは“テンポ指定コマンド % を一切使用しない”ことで実現します (タイミング合せのための単独 % はこの限りではありません)^{26]}。要は、デフォルトのテンポ %125 で首尾一貫して記述されている状況においてのみ、Muse の数値計算処理と、従来の定格化 MIDI の表現が一致するわけです。

むろん、テンポ指定を施さないと演奏データとしては意味を成さない場合が多いのですが、例えば譜面変換がしたい場合は実用上の問題はありませんし、他のシーケンサーに読込ませたい場合は、そこでなんらかの手を加える事を目的としているはずですから、テンポの指定をその1つと考えれば問題はないとも言えます。Muse は、MIDI データを演奏データとしてではなく、音源ドライブのための結果データとして捉えています。演奏情報はあくまでも Muse データ側が担うべき、という思想です。

注

- [1] 音長にはゼロの指定も可能ですが、通常の音名に添えると演奏されない音になります。音長のゼロ指定は 2.1.16 和音とタイミングコントロール (p.20) や 2.1.19 再現表記の活用 (p.23) で、初めてその効果を発揮します。
- [2] 後述する連符にかかるタイや、マクロ範囲をまたがるタイなどを記述する必要がある場合は“音符の連結”という概念を用います。2.1.18 音符の連結 (p.22) を参照のこと。
- [3] 相対変化の結果が 0 ~ 127 の範囲を越える場合は、0 あるいは 127 の限界値がセットされ、文法エラーとしての検出はしません。
- [4] 実音での入力ができる方はこの指定を使う必要はありません。へ音記号だろうがハ音記号だろうが、すべからく ?0 すなわちデフォルトの状態を入力して下さい。
- [5] 和音内のコードにも逆アルペジオ指定が効きますが、和音部分には無効です。また、和音内連符にはアルペジオ指定そのものが効きません。
- [6] 実はドラマーの役割を変更する事もできます。詳しくは 2.3.11 ドラマーの転向について (p.53) を参照して下さい。
- [7] 「ドラムの試聴」ダイアログでは、発音した音程の文字列や、ドラムセットに連なる一連のメンバー属性をクリップボードにセットします。データ作成中にこの機能を使うと、試聴したドラム音を簡便に入力できます。
- [8] 音源によっては、o1 列と o6 列のドラムが部分的に鳴らないものがあります。逆に「ドラムの試聴」の音域を越えて発音できる音源もあります。
- [9] ドラムメンバーは、移調属性 T が無視されます。2.2.4 パーカッションの演奏方法 (p.31) を参照のこと。
- [10] 「楽器の試聴」ダイアログでは、ボタン押下で発音した音色やエフェクト指定の文字列をクリップボードにセットします。データ作成中にこの機能を使うと、試聴した音色や効果を簡便に入力できます。
- [11] MIDI 音源によっては Ethnic や SFX のドラム音色が使用できない機種があります。この場合、直前に選んでいたドラムセットが維持され、結果としてボタンフェースと異なる音になります。逆に音源が上位機種の場合、「ドラムの試聴」で選べるドラムセット以外にも多くの音色を持っています。詳細は 3.2.2 音色 (p.66) を参照して下さい。
- [12] MIDI 音源によっては、エフェクトの効かない仕様の機種があります。パソコンには複数の音源が搭載されている場合があるので、メニューの「音源 (V)」で音源を切替えてみて下さい。より高性能な MIDI 音源が見つかる可能性があります。
- [13] ピッチ変化は、鍵盤表示や譜面モニターには反映しません。
- [14] ダンパーペダルを踏んでいる間は、同時発音数が累積されていきます。踏みっぱなしにしておくと、演奏モタレの原因になるのでご注意ください。適当な間隔を持って Y1 で踏み直すが、Y0 で適宜離すことを推奨します。
- [15] COLR, DRUM, ROOM も、1 つの Muse ファイルに唯一記述するコマンドです。
- [16] STOP や MARK コマンドが存在する場合、テキスト表示領域をマウス右ボタンでクリックすると、その位置を示す逆三角形のゲージマークを ON/OFF できます。このゲージマークは、STOP が赤、MARK が紺で表示されます。更にゲージマークが出ている状態でマウス左ボタンをクリックすると、その位置におけるゲージ間の演奏時間がポップアップします。
- [17] 指定したすべてのフォントが再生するパソコン上に存在しない場合は、デフォルトの「MS P ゴシック」が採用されます。
- [18] 指定できるファイルは、wav/mp3/wma/avi/mpg/wmv/asf 等で、動画系のデータでも音声部分のみを再生します。ただし、Windows Media Player の DLL バージョンが古いと再生しない場合があります。また映像系のデータを指定した場合、多少再生開始に時間がかかる場合があります。
- [19] 指定されたパスで音声ファイルが見つからない場合や、再生不能なファイルが指定された場合でも、エラーにはならず演奏は進行します。
- [20] 一般的な音源のデフォルトは、Z メンバーがレベル 0、その他のメンバーはレベル 1 に設定されているようです。レベル 2 にする場合は同時発音数不足にご注意下さい。
- [21] バリエーション番号はあくまでもベース音から派生しているので、異なる音源で再生しても似た音が出ますが、配布する MIDI ファイルを作成する際は、音源による再現性の差異に留意が必要です。特に、存在しないバリエーションが指定された際、音源によっては音が出なくなるものがあるので注意が必要です。
- [22] 波形加工のパラメータ群はとても感度が高く、僅かな値の変化で大きく音形が変わりますし、MIDI 音源によって楽器やドラムの種類による変化の具合がかなり異なりますので、かけすぎに注意して下さい。また、音源によっては効果が出るパラメータの範囲が 0 ~ 127 より狭い場合もあります。
- [23] DATA および POOL コマンドは、指定された数値をそのまま音源に送出します。7F を越える値はチェックしていますが、それ以外の論理的な整合チェックは、Muse 側では行いません。記述には細心の注意を払って下さい。なお、DATA や POOL で記述したエクスクルーシブの内容は、「メンバー情報」や「フィンガー情報」などのダイアログには反映されません。
- [24] チェックサムの括弧は、POOL と DATA の間をまたぐ事はできません。
- [25] 同時発音数の少ない音源では、それを越える音を出そうとすると音切れが起りますが、この場合もできるだけ後に記述した音が優先的に発音されます。
- [26] エクスポートされた MIDI は、音源や諸々のパラメータを初期化するために強制的に 1 小節分 (4 分音符 4 つ分) のマージンが曲頭に付加されます。ただし、セットアップ小節のテンポは 800 と高速なので、実時間にして 0.3 秒程度です。

参考：速度標語の目安

%40	Grave	重々しくおそく
%45	Largo	幅広くおそく
%52	Lento	おそく
%58	Adagio	ゆるやかに
%72	Andante	歩く速さで
%80	Andantino	やや速く
%92	Moderato	中位の速さで
%100	Allegretto	やや快速に
%132	Allegro	快速に
%160	Vivace	活発に
%184	Presto	急速に

参考：4分音符以外の速度指定における変換式

今、以下のような速度指定が譜面に書かれていたとする。

$$N \text{ 分音符} = m$$

この時、%指定に添える数値 α は以下のように算出できる。

$$\alpha = 4 \times m \times \frac{1}{N}$$

N が付点8分音符の場合は $1/N$ の部分を以下のように算出する。

$$\frac{1}{N} = \frac{1}{8} + \frac{1}{16} = \frac{3}{16}$$

8分音符 付点

3.1.2 タイミング

% α

α . メンバー記号 (省略可能)

指定メンバー内の全フィンガーのタイミングを強制的に揃える。メンバー記号は列挙が可能であり、そのメンバー内で整列処理される。整列タイミングは、対象フィンガーの中で最も先まで進んでいる音符。テンポは以前のものが維持される。メンバー記号が省略された場合、すなわち単独の%の場合は、全メンバー・全フィンガーでタイミングが揃う。

3.1.3 調性

\ α β (初期値 \=)

α . メンバー記号 (省略可能)

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O の 15 の英字の中から列挙。ただし、ドラムメンバーは指定することができない。ここで指定されたメンバーにのみ、 β の調性が与えられる。指定されないメンバーの調性はその時点のものが維持される。省略した場合は、ドラムメンバーを除く全メンバーの指定と見なす。

β . 調号

1~7個の+あるいは-の並び。+が \sharp 、-が \flat を表し、その並びの数が \sharp あるいは \flat の数に対応する。八調の指定は\=と表現する。再度、本指定が行われるまで、この調性が維持される。

記述した位置から作用を開始する。ただし、ドラムメンバー、およびコードに対しては機能しない。なお、x2 指定

下において、音名 b は調性の影響を受けない。また音名 h は、フラット系の調性の影響を受けない。

3.2 メンバー属性

3.2.1 メンバー宣言

$\textcircled{\alpha}$ (初期値 \textcircled{A})

α . メンバー記号 (省略可能)

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, Z の 16 の英字の中から列挙。省略した場合は、その時点のフィンガーが属するメンバーが採用される。以降の P, V, S, R, W, Q, U, Y, T, R=, W=, Q= の効果対象メンバーを指定する。

Z はドラムメンバー専任であり、移調 T の指定は無効となる。

3.2.2 音色

$P \alpha \beta$ (初期値 $P1/0$)

α . 音色番号

1 ~ 128 の番号で音色を指定する。再度、本指定が行われるまで同一メンバー内でこの値が維持される。ドラムメンバーに対しては、ドラムセットの選択として機能する。

参考：上位機種におけるドラムセット

P	ドラムセット	P	ドラムセット	P	ドラムセット
1	Standard1	26	TR-808	50	Ethnic
2	Standard2	27	Dance	51	Kick & Snare
3	Standard3	28	CR-78	53	Asia
9	Room	29	TR-606	54	Cymbal & Claps
10	Hip-Hop	30	TR-707	57	SFX
11	Jungle	31	TR-909	58	Rhythm FX1
12	Techno	33	Jazz	59	Rhythm FX2
17	Power	41	Brush	128	CM-64/32L
25	Electronic	49	Orchestra		

β . バリエーション番号 (省略可能)

スラッシュに続く 0 ~ 127 の番号でバリエーションを指定する。省略した場合は、標準の /0 が採用される。音源側で定義されていないバリエーション番号が指定された場合は、標準バリエーションが採用される (音源によっては無音となる)。バリエーションは各楽器毎に効果する番号が異なる。再度、本指定が行われるまで同一メンバー内でこの値が維持される。ドラムメンバーに対しては機能しない。

3.2.3 音量

$V \alpha \beta$ (初期値 $V127$)

α . 音量値

0 ~ 127 の値。0 で無音、127 で最大となる。+、- 符号付の数値を指定すると、現音量値を基準にした相対変化となる。相対変化の結果が 0 ~ 127 の範囲を超える場合、限界値がセットされる。再度、本指定が行われるまで同一メンバー内でこの値が維持される。

β . 遅延音長 (省略可能)

上記音量に到達するまでの時間を、コロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音

長の対象にはならない。コロンのみの記述はできない。スタッカート指定 / は使用できない。

3.2.4 ステレオ

$S \alpha \beta$ (初期値 S0)

α . ステレオ値

-64 ~ 0 ~ +64 の値。-64 で左のみ、0 で中央、+64 で右のみが鳴る。再度、本指定が行われるまで同一メンバー内でこの値が維持される。

β . 遅延音長 (省略可能)

上記ステレオ値に到達するまでの時間を、コロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音長の対象にはならない。コロンのみの記述はできない。スタッカート指定 / は使用できない。

ドラムメンバーに対しても機能するが、プリセット・パンからの相対である。

3.2.5 残響

$R \alpha \beta$ (初期値 R40)

α . 残響値

0 ~ 127 の番号で残響の強さを指定する。再度、本指定が行われるまで同一メンバー内でこの値が維持される。

β . 遅延音長 (省略可能)

上記残響値に到達するまでの時間を、コロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音長の対象にはならない。コロンのみの記述はできない。スタッカート指定 / は使用できない。

3.2.6 揺らぎ

$W \alpha \beta$ (初期値 W0)

α . 揺らぎ値

0 ~ 127 の番号で揺らぎの強さを指定する。再度、本指定が行われるまで同一メンバー内でこの値が維持される。

β . 遅延音長 (省略可能)

上記揺らぎ値に到達するまでの時間を、コロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音長の対象にはならない。コロンのみの記述はできない。スタッカート指定 / は使用できない。

3.2.7 コーラス

$Q \alpha \beta$ (初期値 Q0)

α . コーラス値

0 ~ 127 の番号でコーラスの大きさを指定する。再度、本指定が行われるまで同一メンバー内でこの値が維持される。

β . 遅延音長 (省略可能)

上記コーラス値に到達するまでの時間を、コロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、

省略音長の対象にはならない。コロンのみの記述はできない。スタッカート指定 / は使用できない。

3.2.8 ピッチ

$U \alpha \beta$ (初期値 $U0$)

α. ピッチ量

-240 ~ 0 ~ +240 の値。0 を標準ピッチとした絶対変化量。10 が半音分のピッチ量。-240 で 2 オクターブ下、+240 で 2 オクターブ上のピッチとなる。0 の指定でピッチを標準にする。再度、本指定が行われるまで同一メンバー内でこの値が維持される。

β. 遅延音長 (省略可能)

上記ピッチに到達するまでの時間を、コロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音長の対象にはならない。コロンのみの記述はできない。スタッカート指定 / は使用できない。

3.2.9 ペダル

$Y \alpha$ (初期値 $Y0$)

α. ペダル制御値

0 あるいは 1 の値。 $Y0$ でペダルを離し、 $Y1$ でペダルを踏む。 $Y1$ が連続した場合は、一度素早くペダルを離れた後、再度速やかに踏む。再度、本指定が行われるまで同一メンバー内でこの値が維持される。

3.2.10 移調

$T\alpha$ (初期値 $T0$)

α. 移調値

-95 ~ 0 ~ +95 の値。半音単位に移調する量を指定する。12 の倍数を指定することでオクターブ単位の上下となる。記述された音程に対し、- で低音側、+ で高音側に移調する。再度、本指定が行われるまで同一メンバー内でこの値が維持される。

なお、本指定はドラムメンバーに対しては機能しない。

参考：主な移調楽器と移調量の関係

D \flat	D	E \flat	E	F	G \flat	G	A \flat	A	B \flat	B
T+1	T+2	T+3	T+4	T+5	T+6	T+7	T+8	T+9	T+10	T+11
T-11	T-10	T-9	T-8	T-7	T-6	T-5	T-4	T-3	T-2	T-1

C Piccolo (+12), Celesta (+12), Xylophone (+12), Glocken (+24), FrenchHorn (-12), ContraFagotto (-12), Contrabass (-12), BassFlute (-12)

D Clarinet (+2), Trumpet (+2), FrenchHorn (-10)

E \flat Clarinet (+3), Trumpet (+3), FrenchHorn (-9), AltoSax (-9), BaritoneSax (-21)

E Trumpet (+4), FrenchHorn (-8)

F Trumpet (+5), EnglishHorn (-7), FrenchHorn (-7)

G AltoFlute (-5), FrenchHorn (-5)

A \flat Clarinet (+8)

A Oboe d'amore (-3), Clarinet (-3), Trumpet (-3), FrenchHorn (-3), Cornet (-3)

B \flat Clarinet (-2), BassClarinet (-14), SopranoSax (-2), TenorSax (-14), FrenchHorn (-2), Trumpet (-2), Cornet (-2)

3.2.11 コントロール

$$X \alpha = \beta \gamma \delta$$

α. コントロール番号

0~129 の値。0~127 は、MIDI 音源のコントロール番号と等しい。128 はキープレッシュャを、129 はピッチベンドの指定となる。

β. コントロール値

コントロール番号が 0~128 の場合は、0~127 の範囲を持つ。コントロール番号が 129 の場合は、-8192~+8191 の範囲を持つ。

γ. ノート番号 (β が 128 の場合のみ指定可能)

ピリオドに続くノート番号の記述で、ポリフォニック・キープレッシュャを指定する。ノート番号は、0~127 の範囲を持つ。60 が中央のド。省略した場合は、チャンネル・プレッシュャとなる。

δ. 遅延音長 (省略可能)

β のコントロール値に到達するまでの時間を、コロンに続けて音長表現で指定する。α が 0~127 の場合のみ指定可能。本指定を行う場合は、それ以前に該当コントロールの開始値を、指定しておく必要がある。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音長の対象にはならない。コロンだけの記述はできない。スタッカート指定 / は使用できない。

3.3 波形加工

3.3.1 出力変位特性

$$R = \alpha \cdot \beta \cdot \gamma \delta \text{ (初期値 } R=64.64.64 \text{)}$$

α. アタック立上がり時間 (省略可能)

0~127 の値。64 が標準値。音の立上がりの速さを調節する。値を小さくすると、鋭く立上がってアタックが強くなり、値を大きくすると、緩やかに立上がりアタックが弱くなる。省略すると以前の値を採用する。

β. アタック減衰時間 (省略可能)

0~127 の値。64 が標準値。音が立ち上ってから持続レベルになるまでの時間を調節する。音色の特徴は、この時間内に凝縮されているため、長くすると太くて厚い音質となる。逆に短くすると、か細く弱々しい感じの音となる。省略すると以前の値を採用する。

γ. 残響減衰時間 (省略可能)

0~127 の値。64 が標準値。音を止めてから音が完全に消えるまでの減衰時間を調節する。コンサートホールの様な残響は、この減衰時間を長くすると実現する。ただし同時発音数に影響するため、過度に長くする場合は注意が必要。省略すると以前の値を採用する。

δ. 遅延音長 (省略可能)

上記特性に到達するまでの時間を、コロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音長の対象にはならない。コロンだけの記述はできない。スタッカート指定 / は使用できない。

3.3.2 変調特性

$$W = \alpha \cdot \beta \cdot \gamma \delta \text{ (初期値 } W=64.64.64 \text{)}$$

α. 周期 (省略可能)

0~127 の値。64 が標準値。ビブラートの振動周期を調節する。値を大きくすると細かい周期になり、値を小さくすると荒い周期となる。省略すると以前の値を採用する。

β. 振幅 (省略可能)

0~127 の値。64 が標準値。ビブラートの揺れ幅を調節する。値を大きくすると激しく、値を小さくすると弱くビブラートする。省略すると以前の値を採用する。

γ. 開始時間 (省略可能)

0~127 の値。64 が標準値。音が出てから実際にビブラートがかかり始めるまでの時間を調節する。値を大きくすると、なかなかビブラートに移行せず、値を小さくすると、直ぐにビブラート効果が現れる。省略すると以前の値を採用する。

δ. 遅延音長 (省略可能)

上記特性に到達するまでの時間を、コロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音長の対象にはならない。コロンのみの記述はできない。スタッカート指定 / は使用できない。

3.3.3 周波数特性

$$Q = \alpha \cdot \beta \gamma \text{ (初期値 } Q=64.64 \text{)}$$

α. フィルター (省略可能)

0~127 の値。64 が標準値。倍音成分の発生量を調節する。値を大きくすると倍音成分を多く通し、澄んだ (硬い) 音になり、値を小さくすると倍音成分が削られ、こもった (柔らかい) 音となる。例として、弱音器の装着などは値を小さくすることで実現する。省略すると以前の値を採用する。

β. 共鳴 (省略可能)

0~127 の値。64 が標準値。上記フィルタリング付近の共鳴量を調節する。この値を大きくする程、高音域の共鳴が目立つ音となる。省略すると以前の値を採用する。

γ. 遅延音長 (省略可能)

上記特性に到達するまでの時間を、コロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音長の対象にはならない。コロンのみの記述はできない。スタッカート指定 / は使用できない。

3.4 フィンガー属性

3.4.1 フィンガー宣言

$$\# \alpha \beta \text{ (初期値 } \#A0 \text{)}$$

α. メンバー記号

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, Z の 16 の英字の内 1 つ。

β. フィンガー値

0~9 の値。

以降の ? , x , o , v , w , p , q , $<$, $>$, $*$, $()$, $[]$, $'$ および、音符指定やコマンド記述が本宣言のメンバー内フィンガーに反映する。ドラムメンバーは音部記号 ? には反応しない。また、ドラムメンバーはコードを使用することはできない。自動演奏時の演奏色はメンバーにより決まる。

3.4.2 音名タイプ

$$x \alpha \text{ (初期値 } x0 \text{)}$$

α. 音名値

0~2 の値。

	ハ	ニ	ホ	ヘ	ト	イ	ロ	変ロ	休符
x0	d	r	m	f	s	l	c		-
x1	c	d	e	f	g	a	b		-
x2	c	d	e	f	g	a	h	b	-

ドラムメンバーに関しても、本指定の音名が採用される。再度、本指定が行われるまで同一フィンガー内でこの値が採用される。x2 指定において、音名 b は調性の影響を受けない。また音名 h は、フラット系の調性の影響を受けない。

3.4.3 強弱

$v \alpha \beta$ (初期値 $v100$)

α. 強弱値

0~127 の値。0 で無音、127 で最強となる。+、- 符号付の数値を指定すると、現強弱値を基準にした相対変化となる。相対変化の結果が 0~127 の範囲を超える場合、限界値がセットされる。再度、本指定が行われるまで同一フィンガー内でこの値が維持される。v0 の音も鍵盤表示されるが、MIDI 出力の対象にはならない。

β. 遅延音長 (省略可能)

上記強弱値に到達するまでの時間を、コロンの続けて音長表現で指定する。遅延は連続的に行われず、あくまでも発音時点での強弱変化である。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると遅延が起らず即座に効果する。本指定で与えた音長は、省略音長の対象にはならない。コロンのみの記述はできない。スタッカート指定 / は使用できない。

3.4.4 アクセント

$w \alpha$ (初期値 $w0$)

α. アクセント値 (省略可能)

-127 ~ +127 の値。強弱 v への加算結果が 0~127 の範囲を超える場合は限界値で律則される。再度、本指定が行われるまで同一フィンガー内でこの値が維持される。省略した場合は、直前に指定したアクセント値が採用される。

指定直後の音名に対してのみ、アクセント値分の強弱変化が起こる。和音やコードなどの直前に指定された場合、複数音に作用する。休符の直前にアクセントを記述することはできるが、無効である。

3.4.5 絶対オクターブ

$o \alpha$ (初期値 $o4$)

α. オクターブ値

- および 0~9 の値。- が最低オクターブで、値が増すほど高音側を表す。

次の絶対あるいは相対オクターブ指定がなされるまで、本指定のオクターブ値が維持される。

3.4.6 相対オクターブ

< または >

現在のオクターブ値を相対的に上下させる。< で高音側に、> で低音側にシフトする。次の絶対あるいは相対オクターブ指定がなされるまで、本指定のオクターブ値が維持される。

3.4.7 音部記号

$$? \alpha \text{ (初期値 ?0)}$$
 α . 音部値

0~6の値。ト音記号としての音階を、指定された音部記号の音階に変換する。

- ?0 ト音記号 (高音部記号)
- ?1 八音記号 (中音部記号) ソプラノ記号 (第1線がハ) 最下線
- ?2 " メゾ・ソプラノ記号
- ?3 " アルト記号 (第3線がハ)
- ?4 " テノール記号
- ?5 " バリトン記号 (第5線がハ) 最上線
- ?6 ヘ音記号 (低音部記号)

記述した位置から作用を開始する。ただし、ドラムメンバー、およびコードに対しては機能しない。再度、本指定が行われるまで同一フィンガー内でこの値が維持される。

3.4.8 出音調整

$$p \alpha \text{ (初期値 } p0)$$
 α . 調整値

+あるいは-に続く1~100の値で、音長に対する調整パーセントを与える。+の場合は音を早めに出し、-の場合は音を遅めに出す。^あるいは~に続く音長記述での調整もできる。^の場合は音を早めに出し、~の場合は音を遅めに出す。

記述した位置から作用を開始する。再度、本指定が行われるまで同一フィンガー内でこの値が維持される。

3.4.9 止音調整

$$q \alpha \text{ (初期値 } q0)$$
 α . 調整値

+あるいは-に続く1~100の値で、音長に対する調整パーセントを与える。+の場合は音を遅めに止め、-の場合は音を早めに止める。^あるいは~に続く音長記述での調整もできる。^の場合は音を遅めに止め、~の場合は音を早めに止める。上記符号を付けず直接音長を指定した場合は、固定的な指定音長となる。

記述した位置から作用を開始する。再度、本指定が行われるまで同一フィンガー内でこの値が維持される。本指定は、スタッカート指定がなされている音符には効果しない。

3.4.10 移調楽器

$$y \alpha \beta \text{ (初期値 } y/)$$
 α . 調号

1~7個の+あるいは-の並び。+が♯、-が♭を表し、その並びの数が♯あるいは♭の数に対応する。=は八調を表現する。省略した場合は、全域属性の調性\の値が採用される。なおx2指定下において、音名bは調性の影響を受けない。また音名hは、フラット系の調性の影響を受けない。

 β . 移調値

スラッシュに続けて、半音単位の $-95 \sim 0 \sim +95$ の値を指定する。記述された音程に対し、 $-$ で低音側、 $+$ で高音側に移調する。12 の倍数を指定することでオクターブ単位の上下となる。再度、本指定が行われるまで同一フィンガー内でこの値が維持される。省略した場合は、メンバー属性の移調 T の値が採用される。

なお、本指定はドラムメンバーに対しては機能しない。主な移調楽器と移調量の関係については、移調 T の項を参考のこと。

3.5 音符記述

3.5.1 音符

$$\alpha \beta$$

α . 音名表現

音階記号と臨時記号からなる。音階記号は、音名タイプ x に従った音名英字の内の 1 つ。臨時記号は、 $+$ が \sharp 、 $-$ が \flat 、 $=$ が \natural を表す。また、 $++$ でダブルシャープ、 $--$ でダブルフラットを表す。臨時記号は、常に調性 \backslash に優先して作用する。また、小節の概念はないので、各音への個別指定が必要。休符 $_$ も音名的一种と見なす。

β . 音長表現 (省略可能)

[音長値 + 付点] + [演算記号] + ... + [スタッカート]

[音長値]

0, 1, 2, 4, 8, 16, 32, 64 の 8 つの数値の内の 1 つ。ゼロ音長は、演奏を無効化する。ゼロ音長であっても、省略音長および再現表記の記憶が行われる。和音記述において、ゼロ音長は占有時間ゼロを表現する。省略時は、同フィンガーでの直前の音長値が採用される。

[付点]

ピリオド $.$ で表す。複付点の場合は $..$ とピリオドを 2 つにする。3 つ以上のピリオドは不可。音長値を記述せずに付点のみ記述することはできない。

[微分音長]

[音長値 + 付点] の位置に、 i に添えた $1 \sim 960$ の数値により表現。 $i1$ が最小分解能であり、64 分音符の $1/60$ の音長を表す。数値はその倍率を意味し、 $i60$ で 64 分音符の音長を表現する。微分音長には付点を添えることはできない。

[演算記号]

3 つの記号 (加算 $^$, 減算 $~$, 乗算 $'$) で表す。加減算記号の記述により [音長値 + 付点] を幾つでも連ねられる。乗算記号は、加減算記号より優先して演算される。乗算記号は、連続して記述できない。乗算記号直後の数字は乗算倍数であり、音長ではない。

[スタッカート]

スラッシュ $/$ とピリオド $.$ の組み合わせで表す。音長表現末尾にのみ記述可能。省略時、スタッカートなしと見なす。音長値と共に省略音長の記憶対象となる。 $/$ の列挙は最大 4 つまで。ピリオドは $/$ の後に 3 つまで。スタッカートやスタッカティシモにはピリオドが 1 つ付けられる。

参考: スタッカートの目安 音長短縮比

$/...$	弱めのノンレガート	(31/32)
$/..$	強めのノンレガート	(15/16)
$/.$	マルカート	(7/8)
$/$	メゾスタッカート	(6/8)
$//.$	長めのスタッカート	(5/8)
$//$	スタッカート	(4/8)
$///.$	長めのスタッカティシモ	(3/8)
$///$	スタッカティシモ	(2/8)
$////$	スタッカティシモ	(1/8)

3.6 群記述

3.6.1 連符

$$(\alpha)\beta$$

α. 音符並び (省略可能)

連続して発音すべき音符を連符の数だけ列挙する。音名表現と音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。スタッカート指定 / も各音で独立に反映する。各音に与えられた音長は、連符内の音長配分比率に反映する。和音やコード表現も記述できる。音符並びを省略すると、直前の連符を再現する。

β. 音長表現 (省略可能)

連符全体の音長を音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ここに与えられた音長内に納るように、連符内の音長を配分する。配分比率は、連符内の音符に与えられた音長比率で決定される。省略時は、同フィンガーでの直前の音長値が採用される。ゼロ音長を指定すると演奏されないが、連符再現パターンは記憶される。

和音やコードを包含することができる。逆に連符を和音の中に記述することもできる。ただし、連符の中に連符を包含することはできない。

3.6.2 和音

$$[\alpha]\beta\gamma$$

α. 音符並び (省略可能)

同時発音すべき音符を列挙する。音名表現と音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。各音の音長は同じである必要はなく、それぞれ指定の音長分鳴り続ける。スタッカート指定 / も各音で独立に反映する。連符やコード表現も記述できる。音符並びを省略すると、休符と同じ意味になる。

β. 占有音長 (省略可能)

和音自体の占有音長を音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ここに与えられた音長が経過した後、次に記述した音が発音を開始する。省略時は、同フィンガーでの直前の音長値が採用される。ゼロ音長を指定すると占有時間ゼロとなり、次音が即座に発音を開始する。

γ. アルペジオ (省略可能)

初音から順次ずらしながら演奏する場合、ずらす量をコロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。ゼロ音長を指定すると発音遅延は起らない。コロンのみの記述で、βとは独立のアルペジオ省略音長が採用される。スタッカート指定 / は無効である。末尾にマイナス記号を添える事ができるが、和音内の単音成分には影響を与えず、コード成分のみ発音順が逆になる。

連符やコードを包含することができる。逆に和音を連符の中に記述することもできる。ただし、和音の中に和音を包含することはできない。

3.6.3 コード

$$'\alpha\beta\gamma\delta\epsilon'\zeta\eta$$

α. ルート音表現

音階記号と半音記号からなる。音階記号は、C, D, E, F, G, A, B の 7 英字の内の 1 つ。半音記号は必要な時に限り記述する。#, b のどちらか一方。調性 \ と音部記号 ? の影響を受けない。ダブルシャープやダブルフラットは記述不可。オクターブは、通常の音符と同様 o の値が採用されるが、F, G, A, B の高音側が指定された場合、1 オクターブ下げて演奏される。

β. コードネーム (省略時はメジャーコード)

コードネームは、以下の 43 の内から選択。() 内は指定可能な別名。

3 重和音		4 重和音		5 重和音	
		6	m6	69	m69
M (省略)	m	7	m7	9	m9
		M7	mM7	M9	mM9
+5 (aug)	m-5	7+5	m7+5	9+5	m9+5
-5	m+5	7-5	m7-5	9-5	m9-5
		M7+5	mM7+5	7+9	m7+9
sus2		M7-5	mM7-5	7-9	m7-9
sus4		7sus4		9sus4	
		dim7 (dim)	dimM7	dim9	
		add9	madd9		

参考：コードに関する譜面上の表記例

M dim alt -5
+ aug maj M on / (分数コード)

γ. 転回形 (省略可能)

< または > を列挙することで、転回の階数を指定する。< で上昇方向に、> で下降方向に転回が起る。列挙できる階数は 4 つまでで、上昇と下降の混在はできない。転回対象には、 δ のテンションや、 ϵ のベース音は含まれない。

δ. テンション (省略可能)

以下の 7 つのテンションを丸括弧内に指定する。

(-9, 9, +9, 11, +11, -13, 13)

列挙の際は、カンマで区切る。ルート音からの相対度数で求めるので転回の影響を受けない。

ε. 分数コード (省略可能)

スラッシュ / に続けてベース音を指定する。本分数コードは転回指定ではなく、単にベース音の追加として機能する。ベース音の表記はルート音表現と同じ音階記号で指定する。ベース音のオクターブ位置はルート音より低くなる様に決定される。よって転回の影響は受けない。

ζ. 音長表現 (省略可能)

コード自体の占有音長を音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。コード内のすべての音はこの音長となる。省略した場合は、同フィンガーでの直前の音長表現が採用される。ゼロ音長を指定すると演奏されないが、再現パターンは記憶される。

η. アルペジオ (省略可能)

低音部から高音部へ順次ずらしながら演奏する場合、ずらす量をコロンに続けて音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。和音内に記述されたコードには指定できない。末尾にマイナス記号を添えると、逆順となる (高音部から低音部へ)。ゼロ音長を指定すると発音遅延は起らない。コロンをみの記述で、 ζ とは独立のアルペジオ省略音長が採用される。スタッカート指定 / は無効である。

和音や連符の中に記述することができる。ドラムメンバーには指定できない。

3.7 補助記述

3.7.1 連結

α & β

α β . 音表記

群記述を含むあらゆる音の表記 (単音・連符・和音・コード)

音符 α と同一音程の音符 β を連結し、音長を結合する。連結は同一のフィンガー内で実施される。和音やコード間の連結も可能であり、一致する音要素を連結する。ただし、和音内の音符同士の連結はしない。連符内の音符同士は連結対象となる。また、連符の初音の前や終音の後に記述すると、連符外の音との連結を行う。 α と β の音程が異なる場合、 α での音長を短くする効果はキャンセルされる。

3.7.2 再現

, α

α. 音長表現 (省略可能)

再現音の音長を音長表現で指定する。音長表現の形式については 3.5.1 音符 (p.73) 参照のこと。省略した場合は、同フィンガーでの直前の音長表現が採用される。

再現記号 , の位置に直前に記述した和音あるいはコードが再現される。再現対象は和音・コードであり、単音・連符は対象外である。本記述は連符内には記述できるが、和音内には記述できない。

3.8 コマンド記述

3.8.1 コマンド

* α " β "

α. コマンド名

以下の 12 名称のいずれか 1 つで機能を指定する。

HEAD	曲頭停止時のタイトル表示 (1 コマンドのみ有効)
TEXT	文字列の表示/消去
MARK	位置決め可能な文字列の表示
STOP	一時停止と文字列の表示/完全停止
FONT	上記 4 コマンドの文字フォント指定
WAVE	音声ファイルの再生
COLR	メンバー色のカスタマイズ (1 コマンドのみ有効)
DRUM	ドラマー転向やロール残響の指定 (1 コマンドのみ有効)
ROOM	演奏会場の設定 (1 コマンドのみ有効)
FING	フィンガー属性の一括指定
DATA	エクスクルーシブの出力
POOL	エクスクルーシブ前半部の記憶

β. パラメータ文字列

各コマンドの機能を補完する文字列。スペース、全角文字、半角カタカナ、半角縦棒 |、セミコロン ;、感嘆符 ! もパラメータ文字列として解釈される。ただし改行とタブは無視される。

HEAD	曲頭において青色で表示
TEXT	茶色で表示 *TEXT" で消去指定
STOP	赤色で表示し *STOP" で完全停止
MARK	紺色で表示 *MARK" で消去指定
FONT	" フォント名, 修飾 " で指定 *FONT" で初期状態
WAVE	再生ファイルパス名 *WAVE" で強制停止 音声系ファイルのパスは相対および絶対の指定が可能
FING	フィンガー属性 (? , x , o , v , w , p , q , y) を任意に列挙指定
DRUM	A ~ Z のメンバー記号でドラマー転向を指定 更に 0 ~ 2 の数字を添えて、ロール残響度を指定 Z- の指定によりドラマーを通常楽器奏者化
ROOM	Rx(x, x, x, x, x, x) と Qx(x, x, x, x, x, x) の形式で指定
COLR	ABC(rrggbb) DE(rrggbb) FH(rrggbb)... の形式で指定
DATA	xx, xx, ... の形式で指定 *DATA" で初期化無効
POOL	書式は DATA と同一 *POOL" で記憶クリア

以下のコマンドは、MIDI へのエクスポートでメタイベントが出力される。これら MIDI 出力コマンドのパラメータ長は、半角換算で最大 127 文字である。

HEAD	Sequence/Track Name
TEXT	Lyric
STOP	Lyric
MARK	Lyric
FONT	Marker
WAVE	Cue Point
DATA	System Exclusive

3.9 マクロ記述

3.9.1 繰返し

$$\{ \alpha \} \beta$$

α . 繰返し内容

中括弧にて任意の指定内容をくくった場合は、定義域として機能する。内容が存在しない場合は、その位置に直前の定義域内容を展開する。

β . 繰返回数 (省略可能)

定義する局面でも、展開する局面でも有効。省略した場合は 1 と解釈する。ゼロを指定すると、演奏を行わない。

3.9.2 定義マクロ

$$\$ \alpha \{ \beta \} \gamma$$

α . マクロ名

任意の英数記号文字で命名。その名称は展開マクロにて使用される。ただし、スペース、タブ、改行、全角文字、半角カタカナ、半角縦棒 | はキャンセル文字として無視される。また、セミコロン ;、感嘆符 ! はコメント記述文字として解釈される。

β . マクロ内容

中括弧にて任意の指定内容をくくる。

γ . 繰返回数 (省略可能)

定義マクロを繰返す回数。省略した場合は 1 と解釈する。繰返し回数は定義域に含まれず、定義内容は β の領域内となる。ゼロを指定すると、定義のみで演奏はしない。

定義マクロ自体も演奏対象となる。

3.9.3 展開マクロ

$$\$ \{ \alpha \} \beta$$

α . マクロ名

定義マクロにて命名した名称。

β . 繰返回数 (省略可能)

定義マクロを展開する回数。省略した場合は 1 と解釈する。ゼロを指定すると、展開を行わない。

著作権などについて

Muse はフリーソフトです。ただし、Muse プログラム、Muse 文法およびドキュメントに関する著作権は、開発者「加藤 一郎」が保有します。本ソフトウェアは、非営利に限り無許可で再配布、および使用が可能です。書籍などへの転載は基本的に快諾致しますが、その件に関するご一報を希望します。

作品ができたら加藤宛 (kato@c3-net.ne.jp) に送って頂けると幸いです。開発の励みにしたいと思います。お礼と言ってはなんですが、送付して頂いたら、MIDI ファイルを Muse ファイルに変換するソフト mid2mus.exe を、お贈りしようと考えています。Muse データを添付し、「mid2mus 希望」とお書添え下さい。もちろん無料です。その他、ご意見・ご感想・ご質問等も謹んでお受けいたします。

なお、本書に例示されている楽曲、および添付のサンプル曲 Sample*.mus については、日本音楽著作権協会 (JASRAC) の作品検索サービスによって、作曲に関する著作権が消滅していることを確認しております。また、カラオケの説明部分で使用している曲に関しては、作詞の著作者も不詳であることを確認しております。

収録楽曲一覧

- 2.1.5 ゆめをのせて (ロシア民謡)
- 2.1.6 ほたるの光 (スコットランド民謡)
- 2.1.7 別れの曲 (ショパン)
- 2.1.8 線路は続くよどこまでも (アメリカ民謡)
- 2.1.9 交響曲第 6 番「田園」(ベートーベン)
- 2.1.10 アルプス一万尺 (アメリカ民謡)
- 2.1.11 アイダ行進曲 (ヴェルディ)
- 2.1.12 おお牧場はみどり (チェコスロバキア民謡)
- 2.1.13 チゴイネルワイゼン (サラサーテ)
- 2.1.14 交響詩「我が祖国」より“モルダウ”(スメタナ)
- 2.1.15 交響曲「未完成」(シューベルト)
- 2.1.17 山の音楽家 (ドイツ民謡)
- 2.1.20 トルコ行進曲 (ベートーベン)
- 2.2.1 交響曲「新世界」(ドボルザーク)
- 2.2.6 亜麻色の髪の乙女 (ドビュッシー)
- 2.2.7 ピクニック (イギリス民謡)
- 2.2.8 賛美歌「きよしこの夜」(グルーパー)
- 2.2.10 シチリアーノ (フォーレ)
- 2.2.11 一日の終り (フランス古典)
- 2.2.12 森のくまさん (アメリカ童謡)
- 2.2.13 月光ソナタ (ベートーベン)
- 2.2.14 ラプソディー・イン・ブルー (ガーシュイン)
- 2.2.15 小フーガ ト短調 (バッハ)
- 2.2.17 交響曲第 9 番「喜びの歌」(ベートーベン)

-
- 2.3.2 静かな湖畔（作詞作曲不詳）
 - 2.3.3 弦楽四重奏曲第1番ニ長調（チャイコフスキー）
 - 2.3.6 幻想即興曲（ショパン）
 - 2.3.7 クラリネットをこわしちゃった（フランス童謡）

Sample1 愛の夢「第三番」(リスト)

Sample2 交響詩「中央アジアの草原にて」(ボロディン)

Sample3 四季（ビバルディ）

著作権が消滅したパブリックドメインな楽曲。それは人類共通の宝物ですね。

開発後記

《Ver1.0 開発後記》1999.03.01

『音楽は時間に描く絵画である』という言葉聞いたことがあります。今回のプログラム開発は、まさにタイミング制御との戦いでした。1000分の1秒でタイマーコールバックが起るので、極限まで瞬間々々の処理を軽くしなければならなかったのです。今まで自分のソフトでマシンをハングさせたことなどなかったのに今回は何度もリセット行為をしてしまいました。ショックでした。

MIDI エクスポートの機構も苦労しました。何が辛かったかって MIDI の仕様を記した資料が見つからないのです。書店には MIDI シーケンサーを使った曲作りの本は山のようにあるのに、MIDI フォーマットそのものの本は本当に僅かなのです。そんな折、情報源で役立ったのは WEB でした。まったくインターネットって奴は、目的を見据えた情報探索に関しては充実した結果を得ることができるものです。

Muse 文法の仕様は、かなりの自信作です。でも一番手が掛りました。Readme.txt のサンプル曲を書きながら、次から次へと発想が沸いてきて、何度も仕様を変更しました。それに伴い、Muse ファイルのコンパイル・アルゴリズムも、幾度となくコーディングし直しました。でも、とってもエレガントな出来だと思っています。自分で自分を褒めて上げたい気持ちです。

ウィンドウの Look & Feel も、とっても気に入っています。シークバーの制御がタイミング制御と密接に関連しているので大変神経を使いましたが、使う側にとっては制約の少ない GUI に仕上がったと思っています。初期の頃は一度演奏を止めないとシークすることができなかったのですから。マニュアルに一度書いた制約事項を消していく行為は、実に快感でした。

実は音を鳴らすプログラムは、今回で 4 回目のトライアルです。1 回目は 16 年前、Beep 音を機械語で ON/OFF し、その周期で音階を作りました。そして、BASIC のデータファイルから曲を演奏できるようにしたものでした。2 回目は 12 年前、初めて個人で手に入れたパソコン “FM-7” の BASIC で音源駆動を利用して開発しました。このプログラムは画面に五線紙が表示され、音符を配置して曲を演奏するものでした。思いの外、データ入力が大変な作業でした。3 回目は 6 年前、“CanBe” で FM 音源を駆動する部分をアセンブラで書き、この Muse のようにテキストファイルから曲を演奏させました。4 つの FM 音源の波形をエディット・ミックスし、音自体を作成してライブラリ化できるシンセサイザーまがいのシステムでした。MS-DOS のプログラムだったので、GUI はまったくありませんでしたが、Muse の原型はここにあります。

そして 4 回目の今回、ついに私が求めていたイメージの作品が完成しました。昨今、巷にはもっと高機能かつ高性能な音楽シーケンサーがあふれています。でも、16 年の遍歴を抱いたこの Muse は、私にとって非常に愛着の沸く作品なのです。

《Ver1.1 開発後記》1999.03.03

初版リリースからまだ 3 日しか達っていないのに、もうバージョンアップをしてしまう自分が恥ずかしいのですが、やっぱり直ぐに最新版をお披露目したい気持ちを押さえ切れません。今回のバージョンアップは、演奏中や試聴中でも音源の切替えを可能にしたことと、フィンガー拍数に [偶奇] ボタンを付けたことです。

もう 1 つ、実は内心とても不安だった CPU 負荷の問題にメスを入れました。1/1000 秒の精度を 1/100 秒程度に落したことで動作が安定したと思われます。しかし一人人間は、音楽を鑑賞するにあたりどの位の時間精度を必要とするのでしょうか？ 少なくとも、私は 1/100 秒で充分のようですが...

さて話は遡りますが、今回の Muse は初版リリース前に入念な試行体制を組みました。会社の同僚の中で音楽的な素養のある人に声をかけ、モニタを募ったのです。特に以下の 4 名の方々には、実際に曲を作成して頂くと共に、有益な示唆を数多く与えてくれました。前回紹介できなかったので、改めてこの場で感謝の意を表します。 田中淳一郎 氏 ・ 河合右輔 氏 ・ 百元正嗣 氏 ・ 大塚剛史 氏

《Ver1.2 開発後記》1999.03.05

これがフリーソフトでなかったら大変な話なのですが、またしてもバージョンを上げることに相成りました。少し堪えようと思ったのですが、Muse 文法記述に関わる部分なので、早くご披露しなければデータ互換が失われ、ユーザからの響響を浴びる可能性があります。

変更したのは x2 指定における音名 h の解釈です。私自身がハイグレードな音楽通ではないもので、仕様を整理するのに大変苦労しました。前バージョンの解釈ミスを指摘してくれた河合君と、今回の仕様検討に快く応じてくれた大塚君に心より感謝いたします。

考えてみると“文法”という奴はデータフォーマットであり、あんまりコロコロ変えられないということに気づきました。現在の Muse 文法にはかなりの自信があるのですが、それでも多少ドキドキします。もし致命的な欠陥や記述不能な譜面が出現したらなんでしょう...。でもまあ、フリーソフトですから、仕様は予告なく変更する場合があります。お気楽モード ON !

おまけと言ってはなんですが、データ作成のための支援機能を付けました。ドラムの試聴でボタンクリックすると、その音階文字列がクリップボードに出力されます。テキストエディタ側でペタッと貼付ければ、楽に入力できるという算段です。楽器の試聴ではその音色 P 指定を出力します。せっかちな私は、音色番号を確認せずにダイアログを閉じてしまう場合がままあったので、少なくとも私自身にとっては、ヒット機能です。

更にシークバーの移動時にツールチップで時刻を表示させるようにしました。考えてみれば、これだけ付録があればバージョンアップの価値はあるでしょう。

《Ver1.3 開発後記》1999.03.11

音楽というものは実に多様なパラメータの組み合わせです。Museのような簡易ツールは、とてもそのすべてを網羅できません。しかし、満点は取れなくとも及第点は目指しています。その意味で今まで心残りだったものは、連続的なテンポの変化と、滑らかな強弱の変化を表現することでした。前者の課題は時間制御の数値計算が実に難解ですが、後者の課題は頑張ればクリアできそうでした。そして、今回はその克服バージョンという訳です。

逆説的ではありますが、滑らかに音の強さを変化させるため、デジタル物量を投入してアナログに見せかけると言う処理になりました。所詮、計算機はそんなまがい物しかできないのです。特にクラシック系の曲は、クレッシェンド・デクレッシェンドをふんだんに使って曲想を表現します。実際の楽器を演奏する“人”の偉大さを改めて感じ入った次第です。すべてのパラメータをアナログ的に、パラレル・リアルタイムで処理しているのですから。

ついでとってはなんですが、ステレオ効果に関しても同じ概念で文法化できるので、同様の書式でサポートしてみました。今回の強化で Muse の表現力がグッと高まったと言えます。それにしても、上位コンパチの書式でよかったあ。

イントラネットに“Muse の殿堂”というタイトルのホームページを開設し、そこに皆から送ってもらった作品を展示することにしました。現在は前に紹介した方々の作品が並んでいます。これから、未知の方から作品が送られてくるようになって、殿堂がにぎやかになるといいなあと祈っている今日この頃です。

《Ver1.4 開発後記》1999.03.20

私は根っから単純なので複雑な文法は嫌いだし、そもそも覚えるべきコマンドが多いのも閉口します。そんな訳で本来の MIDI 音源の能力を引出すエフェクト指定を無視してきました。デフォルト固定でコーディングしていたのです。

しかし開発が一段落した今、ちょっと試みにエフェクトの効果実験をしてみるとこれが思いの外楽しいのです。128 の決められた楽器をただ受身で選ぶのではなく何やら自分なりの音色を創作できる小気味よさが感じられます。

という訳で、とうとう MIDI エフェクト命令まで実装してしまいました。と言っても、ごくごく基本的な 3 つのコマンドだけです。でも私にとっては充分すぎるくらい複雑です。なにせ私は、1・2・沢山 ... という大雑把な人間なので。

ちなみに今回のバージョンは必然的に「楽器の試聴」ダイアログに手を入れることになり、800 × 600 の解像度しか持たないノートパソコンにうまく納めるという改良も行いました。それとツールチップ発生時のウィンドウフォーカスの問題や、クレッシェンド指定時の対象チャンネル算出部の不具合なども直しました。

《Ver1.5 開発後記》1999.04.02

友達から続々と素晴らしい曲が届きました。そしてそれらのデータを聴いたり、見たりしているうちに、また新たな強化の思いが湧起ったのです。1 つは、多くのメンバーを用いたデータに対し、あるメンバーだけ取出して聴いてみたいという欲求です。丁度メンバー色一覧のダイアログが表示するだけで淋しげだったので、そこにメンバーの ON/OFF 機構を付けました。

デバッグにも役立ちそうで、なかなかの機構だと思います。

もう 1 つは、短い音長の連符の際、どうもテンポがもたれる様なので、割込み精度を 3 倍ほど高めました。これにより確かにもたれなくなった様です。果して私の耳は 1/100 秒の精度を持っていたのでしょうか？何はともあれ、今回のバージョンから Muse は、1/250 秒のタイミング精度で動作することになりました。

それと、もう 1 つ。同一の和音やコードのデータが続く時、なにやら入力が面倒そうなのに気づき、それらを統括的に解消するために、再現記法という概念を導入しました。この表記に連符も統合するか否かで 1 週間悩みました。結局、分離する方が記述性が高いことに気づき、() で連符の再現することにしました。

更に、もう 1 つ。ゼロ音長を和音以外にも拡大適用し、上記の再現記法と組み合わせることでリズムパターンをマクロで定義できるようにしました。しかし、このために群記述における省略音長の採用順序を変更しなければなりませんでした。これが一番苦労しました。そして悩みました。何故なら文法の互換を破ることになるからです。互換性を保つ方法もあったのですが、しかし、その奇怪な仕様では、どうしても私の審美感が許さなかったのです。

《Ver1.6 開発後記》1999.04.08

小さな 2 つの改良をしました。1 つは、ナインス系のコードを増強したこと。これにより、Muse が対応するコードは 29 種類に増えました。先日、近所の楽器店にふらりと寄ったら、当り前の話かも知れませんが一般の書店にはない譜面関係の書籍がずらりと並んでいるではありませんか！ Muse に入力することを考え、ワクワクしながら気に入った 1 冊を購入しました。その譜面に思いの外ナインス系のコードが多かったというのが、増強のきっかけです。

今 1 つの改良は、前回のメンバ ON/OFF 機構を、MIDI エクスポート時にも適用拡大したことです。このアイデアは鳥谷氏が e-Mail で提供してくれました。ユーザがある箇所で指定したパラメータが、全体に渡って統一的に作用するという思想は、私の好みにピッタリとフィットしました。

《Ver1.7 開発後記》1999.04.30

『遠いものは大きく、近いものは小さく見えるだけのこと。なあに、近づけばそれ程のことはないさ』...この物理的遠近法を無視した逆説的な論法は、宮崎駿アニメのあるシーンで語られる台詞です。私は妙にこの言葉が印象に残っており、何かに躊躇する時不思議と思い出すのです。

テンポの連続的変化の対応はとても難解で私には解けないし、解けても実装できないと思い込んでいました。しかし着手してみると結構楽しい思考プロセスの旅でした。『傾きの異なる 2 つの半直線を滑らかに接続する放物線』というイメージが、リタルダンドとアツチェレランドを解く鍵だったのです。近づいてみると、本当に大したことはありませんでした。

何故かロードに時間がかかる Muse データが送られてきました。むろんロードそのものに時間がかかるのではなくコンパイル処理に時間がかかるわけですが、一体どこにその元凶があるのか、CPU タイムダンプのデバッグライトではどうしても特定できません。今までこの方法で高速化をした経験が何度もあるのに。お陰で、プロファイラ解析の体験を初めてさせても

らいました。

結局、長～い休符がある場合に、次音の挿入位置を検索する処理に時間がかかっていることが判明しました。そこで、異なる2つの対応策を捻出しました。一つは、音の追加時は単純に列挙しておき、最後にパッチ的に時系列でクイックソートする方法。今一つは、休符に関してもダミーのノートを挿入追加し、あくまでもリアルタイムに時系列整合を保っておき、最後にダミーノートを解放する方法。

両方とも時間をかけて丁寧に作り込み、今までのデータで実験検証した後、結論としてリアルタイム処理の方を採用しました。これで10倍程の高速化が得られたはずですが、著名な小説家や画家の創作プロセスに喩えるのはおこがましいですが、Museも最終プログラムに至るまで、数多くの習作を神に捧げています。

繰返しモードのインターバル時点で、マウスによる演奏制御をすると最悪ハングが起るといふ障害にも遭遇しました。結構大手術でした。でもこの病も完治し、今回のバージョンで、Museはかなり箔が付いたと思います。

《Ver1.8 開発後記》1999.06.01

初版から丁度3ヶ月が経ちました。改版インターバルも長くなり、Museもだいが落ち着いて来たようです。今回は、大作の入力に挑戦することで得た経験をもとに、自らデバッグに便利と思われる機能を2つ追加しました。

1つは「メンバー色一覧」において印を一齐にON/OFFする機構。今一つは「フィンガー拍数」において最終属性値を表示する機構。両機能とも、体験に裏付けられているので必ず役に立つという自信があります。

考えるに、モノを作るということは、モノを使うという行為により完結するのだとしみじみ思うのです。思い起せば初版を作り上げて後、自ら入力を繰返すことで、Museはここまで進化してきました。今後もこの姿勢を崩さずに、事に臨んでいこうと思っています。

《Ver1.9 開発後記》1999.06.12

遅延効果の機構を実装した時期から、釈然としない部分がありました。それは、メンバー属性の宣言@がフィンガー宣言#の配下に記述されることが間々あるという点です。しかし記法としてはなんら矛盾していませんから、何がどうまづいのか具体的なイメージが描けない日々が続いていました。

けれども先日、帰宅途中の電車の中で突然その欠点が形になりました。マクロの中に@指定を施す場合、メンバー指定が定数なため、マクロの使い回しに足枷が生じ、柔軟性がないことに気づいたのです。その明確なデメリットに思いが到達した時と、新しい記法を思いつくのがほとんど同時でした。メンバーを省略可能とし、その際フィンガーの属するメンバーを採用するようにしさえすれば問題は解決します。プログラムの修正量はたった2行でした。

フィンガー配下の@指定で、再度メンバーを書く必要はないし、マクロの汎用性は高まるし、しかも文法互換は保たれるのです。今まで文法エラーではじかれていた記述が、新しい解釈で通るだけのことです。すべての側面で満足行く解に出会うのは、然う然うあるものではありません。

問題を解決するには、まずその問題の本質を厳密にイメージすることである。これが今回、私の得た教訓でした。

もう1つ、無名マクロの文法を追加しました。スコア風に入力する際、オクターブの異なる同一旋律のメンバーを書いていくのにマクロ名の命名が実に煩わしいと感じたからです。

嬉しかったのは、文法互換がここでも保たれたことです。結局、従来文法エラーと解釈する記述に対して、何らかの機能を付加する分には、上位互換が保存されるという指針を得ることができました。

《Ver2.0 開発後記》1999.06.20

ぜんぜん改版インターバルが短くなっていないことに気づきました。でもそれに気づくよりも先に、新しいアイデアに気づいてしまうのですから仕方ありません。まあ、考えてみればプロセスを楽しむ趣味という行為を満喫していると言えましょう。私の信条は『仕事はアウトプット、遊びはプロセス』なのです。

今回の改良は、GUI系と文法系にそれぞれ施しました。GUI系の改良は、シークつまみの位置をダイレクトに位置決めできるようにしたことです。デバッグを繰返す過程で、マウスによるスクロールが煩わしいことに気づいたのです。

もう1つ、マウス右ボタンを導入し、一連の操作を一気に行うようにしました。以前より、ツールバーにミニボタンを付けるなどしてリロードを簡便にできないかという要求を受けていましたが、単に専用のボタンを設けるだけでは、真の欲求を満たさないのではないかという懸念が払拭できず、実施する気になりませんでした。今回、デバッグ中に繰返す自分の操作手順を冷静に分析して、より高次元で解決する機構が実装できたと思っています。

文法系の改良は、先日実装した無名マクロにおいて、定義域の指定時でも繰返し回数を添えられるようにしたことです。考えてみると、今までMuseには単純な回数繰返しの指定がありませんでした。今回ようやく可能になった訳ですが、この文法の改造は、とてつもなく大変な作業でした。告白しますが、マクロがらみのアルゴリズムは、驚愕に値する複雑さです。

でもまあ、今回も文法互換が保てて本当によかった。胸をなで下ろしています。何故って、もう私のMuseデータライブラリは100曲に迫る勢いなのですから。

《Ver2.1 開発後記》1999.06.26

長い、長～い、それはそれは長いバイオリンの通奏を入力する機会があり、音長を加算記号で連ねていくのが思いの外面倒でした。無名マクロや連結記号で工夫すれば、まあ許容できる記述量に納めるのですが、こと遅延効果に至っては、そんな工夫もできないことに気づいたのです。

そこで音長を加算・減算に加えて、乗算を新設することにしました。最後に1つだけ大切に置いて置いた逆シングルコーテーション文字を、とうとう使用することになり、これで特殊文字はすべて使い尽してしまいました。しかし凶らずも、音長の演算子3つがすべて上付文字となり、その統一感に満足しています。

初版の頃から、ちょっと手抜きかなと思っていた、複付点とダブルシャープ・ダブルフラットに対応しました。共にたま～にしかお目にかからないし、読替えればさほど苦もなく記述できるので放置していたのですが、制約条件解除の快感を味わいたくなって手を出しました。

ダブルシャープ、ダブルフラットの対応は6行程度のif文を

追加するだけでできてしまったのですが、複付点の方はちょっとやっかいです。64 分音符に複付点を付けると、Muse の持つ内部分解能を越えてしまうのです。分解能を倍にするために、慎重にソースコードをトレースする必要がありました。また、対応ついでに従来から端数が出るので気になっていた“音長数字省略の付点のみ記述”を文法エラーとしてガードすることにしました。

一種の文法非互換となる訳ですが、今まで通っていた記述が、通らなくなるのだから、修正箇所と修正方法が明確であり、まあ問題ないでしょう。従来文法エラー記述に新しい解釈を加えることが問題ないように、従来解釈を文法エラーにするのも、ほぼ問題ないという法則が見えてきました。結局、非互換問題は文法エラーに関連しない所で、従来と異なる解釈にすり替わるときに起るようです。

本格的なクラシックの音楽を入れて、強弱設定を本気でやりました。特に、クレッシェンド・デクレッシェンドでメンバー音量 v を振ると、今どの値になっているのか、フィンガー音量 v と相まって、訳が分からなくなってきます。そこで、フィンガー拍数のダイアログに、メンバー音量 v も表示するようにしました。これで、一段とデータ作りは楽になったのですが、このダイアログ、なんだかやたらに賑やかな情報ウィンドウになってしまいました。

《Ver2.2 開発後記》1999.07.07

最近、Muse の処理がオーバーフローする程の超大作が送られてきており、ちょっとスペックが不足気味のマシンでは、曲がもたれてどうしようもない状態が出現しました。解析の結果、原因はスクロールバー処理と鍵盤カラー表示の重さと判明しました。前者は、ノンウェイトのメッセージポストに変更することで解消しましたが、後者は、ひとえに鍵盤表示のデータ構造に依存しています。

Muse は 16 名のメンバーが 10 本指を持っており、単純に考えると同時に 160 の音が出せるように思えます。しかし、1 本指でも和音が出せ、しかも文法上、何重の和音でも記述できる訳ですから、限界はないに等しいのです。この状況下でカラー表示の重なりを処理するのは至難の業と言えます。

実は Muse の鍵盤上に舞踏するカラー表示は、かなり難度の高い処理をしています。3D 処理と言ったら言過ぎかも知れませんが、重なった色を剥がした時、ちゃんと下の色を表出させるための機構は結構やっかいなものです。

今まで、単純な線形リストで管理していたのですが、今回専用の構造体を新設し自前でメモリ制御を行うという荒技を施しました。Windows32bit タイマーにおける Latency 問題は残るものの、演奏のもたつきはかなり改善されたと思います。

また、連続する同一のコントロールメッセージのように、意味のないコードを削除するクリーンアップルーチンを開発することで、より CPU の負荷を軽減しました。

開発して半年。やっとボリュームとエクスペッションの関係が理解できました。今まで、常識的判断で、音量調整 v はマスタボリュームメッセージを使用していたのですが、MIDI 界では、ソングボディ内にボリューム変化を与えてはいけないお作法になっていることを知りました。ボリュームは、まさに演奏時、個々のマシン音源特性に合わせたチャンネル音量バランスの調整用にとっておくものとのこと。お説御尤もと納得し、改修いたしました。

《Ver2.3 開発後記》1999.08.08

河合氏が所有している一般公開のホームページに Muse を掲載してくれました。恥ずかしながら、私は未だにデジタル・ホームレスでありまして、まあ居候させてもらってる、という所でしょうか。ありがとう河合君。お世話になります。ちょっとオーバーですが、とうとう Muse も全世界に船出したことになります。まったく見知らぬ人が Muse を楽しんでくれると思うとワクワクします。

今回の機能強化は、ピッチベンドを取込んだことです。トロンボーン奏者、大塚氏の強つての（それは初版の頃からの）願いだったのです。お待たせしました。例によって、MIDI 仕様を理解するのに、インターネットをフル活用させてもらいました。私も見知らぬ人の情報提供を甘受している 1 人な訳です。

ピッチベンドは、他のコントロールとは一線を画す難度の高い制御です。また、データ作りの上でも、活用はちょっとセンスを要します。そんなピッチベンドの指定を、Muse の記述ポリシーを崩さずに文法化できたことが、とっても誇らしい気分です。従来遅延効果の概念を利用することに気づいた時は、体が宙に浮く程嬉しかったのを覚えています。

この喜びに勢いを得て、ダンパーペダルの操作、すなわちホールド制御も実装してしまいました。確かに、ピアノ曲ではその響きが違います。結局、今回の改版は楽器特性の演奏手法に対する強化シリーズとなった訳です。

更に、ピアノ曲を多メンバーで入力してみたら、@ 指定の強制休符調整が無意味どころか、邪魔な機構であることに気づき、それを撤廃しました。これがあると、多メンバーを一気に指定できる @ 指定の長所が生かせないのです。またしても、数百行のプログラムを闇に葬ってしまいました。

文法互換も乱れましたが、メンバー属性のタイミングをフィンガー記述で一元化するという新しい概念のための代償です。幸いにも修正箇所は少なく、その場所も明確ですから、お許し頂きたいと思います。

何はともあれ、ピッチベンドの指定とホールドの指定を取込んだことで、Muse は初級シーケンサーから中級シーケンサーへと格が上がったと言って良いかも知れません。

《Ver2.4 開発後記》1999.08.15

Muse の開発に着手する 1 年以上も前に、いつか Muse の様なソフトを作ろうと思い MIDI フォーマットに関する本を購入していました。その本は、比較的専門的な内容で、いざ開発に着手しようとした際には、あまり役に立ちませんでした。その記述は、基本的なことがすべてわかっている人に対しての説明だったのです。

Muse 開発がここまで来た今、改めて読返してみました。そして、そのほとんどを理解できる自分に驚いています。忘れもしない今年の成人の日に、プロトタイプのコーディングを開始して以来、私は MIDI に関する諸々の知識を吸収し続けていたのだなあと、しみじみ感じるのでした。

さて今回、ドラムマップに拡張音が存在することをまさにその本から知りました。今まで純標準の 47 音のみのサポートでしたが、16 音増強し 63 音にしました。それに伴い、試聴ボタンの配列が変わり、デットスペースが発生したので、以前から気になっていた不足機能を強化するきっかけとなりました。それは、ドラムの試聴ダイアログでの、ドラム音に対するエフェクトの確認機構です。

自分の成長を自分自身で感じるという喜びを味わった、今回の改版劇でした。

《Ver2.5 開発後記》1999.08.25

“二大巨頭”という言葉は、両者が切磋琢磨してその対象を高めていくという、自然の摂理が生み出す必然なのかも知れません。古今東西、数々のテーマに於てこの現象が見られます。MIDI フォーマットの世界も例外ではありません。GS と XG がそうです。前回バージョンアップしたドラム拡張音ですが、残念ながら両者で異なる配列を持っていました。私のようなアプリケーションの開発者から言わせると、歩み寄って共通仕様を組んでくれなかったことが遺憾です。勝負は音色そのもので行って欲しかった。音色マップのような基本仕様は一緒にして欲しかった。だって、土俵が違ったら戦にならないじゃないですか。

と言うわけで、自衛手段の発令です。GS と XG のラジオボタンを設け、ドラムマップ自体を内部で整合させるという設計を捻出しました。これならば、Muse のコーディングデータは共通に使用できるし、どちらの音源でも同じ演奏が再現できる訳です。我ながら、なかなか優れた行司案だと感じてます。でも、各音源に指定するプロパティなので毎回指定するのは面倒です。そんなこんなで、とうとう初期化ファイル Muse.ini を立ち上げてしまいました。

もう1つ、かなり骨のある強化をしました。メンバ色一覧のダイアログに、演奏楽器を表示させるようにしたこと。このため、横型レイアウトを縦型に変更しました。従来のレイアウトはピアノ鍵盤の縦横バイメージに合っていて気に入っていたのですが、楽器名を横文字で表示するためにはしかたありません。

でも、レイアウトの変更などは鼻歌混じりの作業であり、問題は表示自体の制御でした。Muse は楽器の持替えができるので、この表示はリアルタイムで行わなければなりません。シークバー連動の表示制御や、最初に構える楽器の抽出など、悪戦苦闘しました。結局、文字列表示と基本的に同一の制御であることに気づき一気に霧が晴れたという所です。複雑だと思われる制御も、整理体系化すれば、見通しが良くなるということをもっと知りました。

《Ver2.6 開発後記》1999.09.22

今回は一種のバグ FIX バージョンといった感じです。連符内に記述された遅延効果を伴うメンバー属性において、その遅延音長が連符内配分長ではなく、実音長になっていたことに気づき、直しました。また、和音が連符内に入っている場合に配分音長の考慮欠落があったので、手を入れました。加えて、表示上の些細な問題ではありますが、フィンガー拍数、テキスト表示領域、楽器の試聴のそれぞれに改修を施しました。

さて、前回のバージョンから今回に至るまで、実は Muse にとって大きなイベントがありました。前回の改版の直後の 8/30 に、ものは試しと、かの有名なダウンロードサイト「窓の社」に公開を依頼してみたのです。そうしたら、その日の内にいともあっさりとしてニュースに掲載されました。その影響で、掲載後 12 時間で 1000 件を超えるダウンロードが加わりました。

この勢いに乗って翌 8/31、今度は「Vector」に投稿。9/3 に採用通知が届きました。更に 9/7、今度は「フリッポ」というサイトから掲載の依頼を頂きました。同日、(株)KK ベスト

セラーズ「Cyber Globe」という雑誌から CD-ROM 掲載の依頼があり、続いて 9/18 に(株)アスキー「テックウィン別冊」からも掲載の依頼を頂きました。

今思い返すと、この一連の動きは Muse にとって、船出のようなものかも知れません。海は広く、船は小さい。でもまずは沈むことなく滑り出しました。もう後には引けません。舞上がっちゃいけない、浮足だっちはいけない、と思いつつ、やはり嬉しいと言うのが正直な感想です。だって、人から認められるということが自分の存在意義の確認には最も効果的なのですから。

そして、なにより楽しみなのは、未知なる人から舞込んでくる e-Mail がグッと増えたことです。一体明日は、どんな人とコミュニケーションができるのだろう。考えただけで、ワクワクしてくるのです。

《Ver2.7 開発後記》1999.10.04

思い込みや勝手な解釈は、大抵の場合間違っているものです。ドラムパートは、その音程により音色を規定するため、ピッチ指定などできる訳がないと思い込んでいました。もしできるとしたら、スネアからシンバルに次第に音色が移っていく様な、音楽版モーフィングという高度技術になると勝手に想像していました。

ところが実際に試してみたら、ちゃんと同一の音色でピッチが変化するではないですか。それどころか、ドラム音色の表情をかなり変化させることができます。ドラム Roll に関しては、ロールスピードの調整といった変化さえ起ります。喜び勇んで早速実装することにしました。これでドラムを鳴らす楽しみが増えます。

改版のついでに、Muse が生成するモードレスなダイアログ群に、スクリーンに対する 4 隅フィット機構を付けたり、メンバ色一覧の楽器表示のちらつきを抑止したりしました。また、エクスポート MIDI で拍数メタイベントの出力を止めました。

さて、実は今回、もっともっと大きな思い込みによる過ちの修正を施しました。それは著作権侵害の件です。多くのホームページで MIDI データが掲載されていることから、MP3 の様な原音コピータイプでなければ著作権に抵触しないと勝手に解釈していたのです。これがとんでもない話であることを大海の中で知りました。船出した Muse が初めて出会う嵐と言えましょう。

と言うわけで、Readme.txt のすべての例示楽曲と、添付している Sample*.mus を見直し、JASRAC のホームページで著作権が消滅していることを確認しました。嵐は去り、晴れて公開可能な状態になりました。さあ、また航海を続けましょう。

《Ver2.8 開発後記》1999.10.30

もし、MIDI ファイルを Muse フォーマットに変換できたら、世界中に存在する膨大なデータを Muse で鳴らすことができます。それは初版の頃から思いついてはいたのですが、Muse は MIDI に比べて器が小さいので、各種イベントを取りこぼしてしまい、まともに変換できないだろうと思っていました。

しかし当時と比べて、Muse もかなり成長しました。そこで、mid2mus.exe の開発に着手することにしました。音長に対する表現機構が全く異なるので、大変苦労しましたが、どうにか

及第点に達する段階にこぎ着けました。このツール、始めは Muse パッケージの付録としてフリー配布しようと思っていたのですが、もっと面白いリリース形態に気づきました。

それは私に Muse データを送ってくれたら、ご希望の方へプレゼントとしてお贈りするという方法です。この方法の卓越したところは、一方向の提供形態であるフリーソフト Muse のコミュニケーションを双方向にすることができる点にあります。私が Muse を提供し、利用者がデータを返し、更に私が mid2mus を贈るのですから、最低でも 1 往復半のやり取りを起すことができます。

さて Muse 本体の話題に戻ります。今回、mid2mus の開発に際し、保有している MIDI ファイルでテストを繰り返したのですが、2 つ気付いたことがありました。1 つは、ドラムセットの切替えが、そのチャンネルのプログラムチェンジで簡単に行えるということです。そこで、Muse もドラムセットの選択を可能にすることにしました。文法上はガードを外すだけで済みましたが、いろいろなダイアログに影響を与え、結構手間がかかりました。

そして、またしても GS と XG の仕様差に悩まされました。エスニックや SFX の音が XG では簡単には出ませんでした。両音源の互換性を維持しつつ、鳴らす方法もあるのですが、あまり凝ると重たくなってしまうので妥協しました。そろそろ、Muse も GS と XG の分岐点まで到達したのかもしれない。

もう 1 つは、モジュレーション(揺らぎ)の遅延効果を使用しているデータが思いの外多いということです。確かに、市販のキーボードの左端にはピッチベンドと並んでモジュレーションのホイールが付いている様です。

という訳で、揺らぎ・残響・コーラスの 3 つのエフェクト指定に遅延効果を実装することにしました。これで結局、ほとんどのメンバー属性で遅延効果が使えるようになった訳です。始めから、何も考えず機能実装してしまえば良かったのですが、私は『不要な機能は無駄なだけでなく、邪魔なものである』という思想を持っているので、どんな些細な機能実装も充分な検討を施した後でないとは実行する気になれないのです。軽いフィーリングなのに何故か高機能というソフトを実現するには、機能実装の決断検討にかかる時間が、大きく影響すると思うのです。

《Ver2.9 開発後記》1999.11.23

毎回、新しいバージョンを出す度に「もうこれで完結編！」と思ってリリースするのですが、何故か連鎖的に次々とテーマが持ち上がってしまいます。せっかくドラムセットが各種選択できるようになったけど、ドラマーが 1 人では、異なるドラムセットを重ねて同時に演奏できません。この指摘はパワーユーザの 1 人、百元氏から頂きました。尤もな話です。そこで、チャレンジ！GS と XG の差に怯えながらも、どうにか利用者に差を意識させない仕様で実装できました。

実にゆっくりしたペースではありますが、Muse は MIDI 仕様の進化の軌跡を忠実にトレースしながら強化しているように思えます。そうすると、次に来るのは音色パラメータの制御 NRPN です。しかし音色加工の世界に入り込むと、大変な騒ぎになってしまう感があり、今回は開発着手を思い止まりました。

演奏時間 55 分、全 12 曲。CD アルバムを丸ごと詰込んだデータが送られて来ました。Muse を開発した本人も流石に腰

が抜けそうです。そのデータは、各曲を個々のファイルで入力し、最後に束ねて 1 ファイルにしたとのこと。その際、単純に連結すれば良いわけではなく、曲と曲の間で、メンバー属性やフィンガー属性の初期化を行わなければなりません。特にフィンガー属性の初期化は、だらだらと長たらしいものになっていました。

今後もこの様な束ねの大作が登場することを期待したい所ですが、そうなってくると、どうにか初期化の手間の緩和案を講じたくなります。しかし単純な初期化コマンドを作るだけでは芸がないと感じていました。その時、半年以上も前のクレームを思い出しました。それは、音名をすべて英米系で入力する場合、いちいちすべてのフィンガー属性に x1 を指定しなければならない、というものでした。

思い出した瞬間に、2 つを同時に解決する *FING コマンドの設計が完了しました。という訳で、今回は DRUM と FING という 2 つのコマンド追加となったのです。

《Ver3.0 開発後記》1999.12.30

バージョンの桁が上がる時は、劇的変化をするのが一般的ですが、それ程派手な機能強化ではありません。考えてみると、進化の過程で偶然人類が 10 本の指を得たため 10 進法を採用しただけのこと。巷ではミレニアムだなんだと騒ぎですが、2000 年も 16 進法だと 7D0 年、2 進法では 11111010000 年で、あまりキリが良い感じがしません。でも自然の摂理に節目がないからこそ、人工的な節目を何らかの形で設定することが、人間にとっては大切なのかも知れませぬ。

さて、今回の強化は 3 つ。1 つは、アルペジオの記法をサポートしたことです。Muse の構想を練っている段階から、自分自身で絶対欲しいと思っていたいながらも、連符・和音・コードが複雑な入れ子になった場合の解釈定義の煩雑さから、半ば断念していた機能です。小学校の頃、ギター 1 本でなんて素晴らしい伴奏ができるのかと驚いた経験があり、ずっと後にそれがコードであることを知りました。だからコード機能は私にとって絶対必須だったのです。でも、すべての音が一辺に鳴るとギターらしくありません。今回のアルペジオ実装で、当初準備した ToDo リストが総て消し込まれました。これは、私にとって記念すべきことです。

今 1 つはコマンドラインの引数で、起動時のオプションを指定できるようにしたことです。この強化のトリガは、これからミュージアムとしてデビューしようとしている、同僚の内藤義明氏によります。彼はなんと Muse ファイラーの開発に着手してくれました。複数の曲を管理できない Muse にとっては心強い限りです。皆さんから頂いた曲を BGM にしてパソコンに向っている私にとって、アルバム機能は大変嬉しいものです。

もう 1 つはスペースキー等、キーボードによる演奏制御を可能にし、更にエラーメッセージをクリップボードに出力するようにしたことです。これは、名古屋南山大学の教員の方がきっかけを与えてくれました。視覚障害を持つ学生に試行させてみたいというご要求です。マウス利用が困難であり、キーボードが重要な I/F であるという事実も、改めて認識させて頂きました。果して今回の強化、視覚障害を持つ方に認めてもらえるのでしょうか？そして使ってくれるのでしょうか？不安と、今までにない大きな期待の入り混じった年の瀬となりました。

3 つの強化とも地味なものではありますが、1 つ目は自己の

初心を貫徹した最後の機能であり、後の2つは Muse が他者との新しい関わり方を開始したという意味で、節目バージョンに相応しいと、私は思っています。

《Ver3.1 開発後記》2000.01.15

もし、ソフトウェアにも誕生日というものがあるなら、Muse の誕生日は 1/15 です。ちょうど 1 年前の今日、Muse の最初の 1 行目をコーディングし始めました。今回のバージョンアップ・テーマは“曲想表現の強化”とでも申しませうか。ヴェロシティ（強弱）の相対変化、微妙なタイミング制御（微分音長）、そして柔軟なテンポ指定の 3 つの強化を行いました。1 年が過ぎてなお、これ程の文法上の強化をする余地があるとは！「音」なんて、ただの空気の振動なのに、どうして「音楽」って奴はこんなにも奥が深いのでしょうか。

中でも“柔軟なテンポ指定”の強化は大手術でした。手術は何日にかけて行われ、一時は死ぬのではないかと（私ではなく Muse がです）とさえ感じました。なにせ、時間制御の根幹部分にまで手を入れたのですから。% 部分の強制タイミング合せに関する文法互換も崩してしまいました。お陰で、今までのデータはコンバートしなければならぬわ、mid2mus.exe も作り直しを余儀なくされるわで、大騒ぎです。もっとも悩まされたのは、すでに Muse を利用している多くのユーザの方にこの事態を把握してもらうことでした。でも、河合君がホームページに掲示板を作ってくれたお陰で、この困難もかなり緩和されました。心底感謝しております。

満 1 歳を迎えた Muse。手術は成功しました。新たな曲想の表現手段を装備して、また歩き始めます。これからご支援の程、よろしく願います。

《Ver3.2 開発後記》2000.03.02

Muse 文法は一種のプログラム言語です。だから、デバッグの容易性はとても重要なポイントです。その意味で自分自身が欲しいと思いつつ、他のプログラム言語にもないので、許容していた点がありました。それは、ブロックコメントの階層記述です。そもそも！マークという開閉関係も定かでない 1 文字の記号を割当ていたので、それを表現するのはとても困難でした。文法互換は崩したくないし、さりとて、もう特殊文字は残っていません。

しかし先日、突然に閃きました。スタックカートや調性記号のように！の列挙記述個数を利用すれば表現できるのです。自分でも驚いているのですが、入れ子だけでなく互違いの関係も記述できる、ちょっと他のプログラム言語では例を見ないブロックコメントを実現できました。

以前より気になっていた、タンギング時のぼやけた印象にも手を打ちました。ノンレガートやマルカートをスタックカートの一つとして実装し、ゲートタイム制御を統一的に扱える表記を与えました。文法互換も維持でき、納得の仕様です。

複数の楽章/楽曲を含む大作が増えてきました。そこで、各楽曲の頭出しを容易にする MARK コマンドを実装することにしました。この機能のヒントは百元君からもらいました。また、STOP や MARK があってもダイレクトなシークポイント指定を可能とするため、頭出しの操作仕様をスライド領域から左右の三角ボタンへと変更しました。データ内容に依らない、統一感のある操作仕様になったと思います。

ホームページの掲示板上にて頂いた 2 つのご要求にも対応し

ました。1 つは、Muse のウィンドウを常に手前に表示する機構で、データ作成時の使い勝手が向上すると予想されます。もう 1 つは、起動時のウィンドウ位置を初期化ファイルに記憶することです。前々から逡巡していた機能ですが、MuseFiler との連携という根拠を示して頂き、実装する気になりました。一条さんのお陰です。

メールで頂いたご要求にもお答えしました。それは強制タイミング合せのメンバー毎の指定です。この実装により、私自身今まで歯痒い思いをしていたスコア譜とパート譜の混在が可能になりました。そして、同じ記述発想で、調性記号の指定もメンバー単位に行えるように改良しました。これは移調楽器が混在した交響楽の記述に絶大な効力を発揮します。鈴木さんのお陰です。

実はもう 1 つあります。Windows2000 への対応です。ちょっとトリッキーなことをやっている部分があり、それが引っかかってしまいました。シークバーにおけるマウス右ボタンのリロード&ラン機構です。どうやら Windows2000 はスクロールバーに右ボタンによる標準的なポップアップが実装された模様です。そこで今回、正当なフック処理を行うように改めました。榊井さんのお陰です。

前回の大手術の後遺症がありました。無駄な MIDI イベントを多量に吐出するという障害です。見つけにくいバグだったのですが、岡本さんの詳細な現象レポートのお陰で完治することができました。このように Muse を支えるミューザーの力が、最近一段と大きくなってきました。感謝の念で心が満たされる思いです。

《Ver3.3 開発後記》2000.04.03

前バージョンから薄々感づいてはいたのですが、とうとう Readme.txt のファイル容量が Muse.exe より大きくなってしまいました。あまりにも長くそして日増しに膨らむ内容に、ミューザーの方々が閉口しているのではと心配です。でも一般書籍に比べれば薄っぺらなものですから、ご容赦下さい。分冊も考えたのですが、私はインストールするとファイルがごちゃごちゃ展開されるソフトが好みではありません。だから、Muse.exe も DLL なしの一発勝負で行くつもりです。

今回のバージョンで強弱 v にも遅延効果を記述できるようにしました。ピアノ曲のデータを作成する際、クレッシェンドなどに音量 v を使うことに妙な罪悪感がありました。これからは容易に、そして積極的に v が使えます。それに v はフィンガー単位なので、ドラムに表情を付けやすくなります。

もう 1 つ。DOS 窓からウィンドウを起動することなく、バッチ的に、Muse データを MIDI ファイルに変換するコマンドラインオプション *0 をサポートしました。あまり使用局面がないかもしれませんが MML コンパイラ風に利用できます。実はこの機能は、私自身の開発用です。新しい文法を実装した際、過去の皆さんのデータで上位互換が保証されるかを、MIDI 出力しバイナリ比較してチェックしています。その作業がバッチ的に行えるようになって効率が上がります。

初版から気になっていたのですが「まっ、いいか」というお気楽モードで放置していた不具合にも対処しました。再描画の時、音が出ていない鍵盤に色が付いてしまうことがよくあったのです。今回プロセスのシグナル制御のお勉強をして、やっと手を打てました。お気楽も良いけど、立ち向えばその分賢くなるものです。

最近になって、やっとコードネームの命名規則に自分なりの解釈を与えることができました。今までは、各コードが固有名詞の如くパターンを決めていると妥協するしかない程、その規則性を見出せなかったのです。今回見つけ出した規則性をもとに自信を持って新たに 12 コードを増強し、合計 41 コードとすると共にテンション記述も加えました。数学的組合せとしては、完全に網羅しているとは言えませんが、音楽的には譜面記載のほとんどのコードを書下せると思います。

また、長い開発後記になってしまいました。できるだけ Readme.txt の容量を増やさぬよう短くまとめようと思っていたのですが...。この辺で筆を置きます。ちなみに私の e-Mail アドレスが変りました。よろしく願います。

《Ver3.4 開発後記》2000.05.10

よく音楽は、作曲家と演奏家が必要と言われますが、私にはそれに加え第 3 の重要人物“聴衆”が必要だと考えています。音楽が完成する瞬間とは、演奏された楽曲がそれぞれの聴衆の頭の中で再構築される時なのです。それは聴き手の人生経験という各人各様のかかわりで組立て直される訳であり、同一の音楽でも聴き手の数だけ微妙に異なる音楽が存在しているとも言えます。

今回ベートーベンの交響曲をミュージングしてみても、この念を更に確信しました。彼の創り出した響きが、200 年近い時の流れを越えて Muse 演奏され、私の頭の中、いや五感で再構築されていく。そんな感じを強く受けたのです。

さて本題の開発の件ですが、流石に交響曲ともなると各楽章の演奏時間が長く、シークバー上でどこからどこまでが 1 つの楽章か知りたくなりました。そこで、テキスト表示領域に *MARK の位置を表示するゲージマークを ON/OFF できるようにし、*STOP も色違いのゲージマークで表示するようにしました。この機構はデバッグの際にも便利だと思えます。またよく CD ジャケットに記載されている各楽章の演奏時間も知りたくなりました。そこで、上記ゲージマークを表示している時に、マウスの左ボタンクリックで演奏時間がポップアップする工夫をしてみました。

文法上のちょっとした強化も行いました。曲想表現を豊かにするため、前々回のノンレガート表記を水平拡張し、スタッカートやスタッカティシモよりちょっと長めの指定も可能にしました。これで 8 分割のゲートタイムは完全サポートです。

例によって少々改造でバージョンアップをしてしまいますが、ぜひ利用してみてください。ソフトウェアにも“聴衆”が必要です。企画制作する人、広く世に知らしめる人、そしてそれを使用する人の 3 者がそろっていなければ存在意義を失います。皆さんに使ってもらうことで、初めて Muse は完結すると思うのです。

《Ver3.5 開発後記》2000.06.17

開発を始めて 1 年半も経ってしまいました。元来移り気な私が、これ程長期に渡り一つのソフトウェアに取り組むなんて、冷静に考えてみると気を失いそうです。

今回は“コード転回”を実装しました。当初はコードネームだけで平易に伴奏が付けられることを売りにしようと考えていたのに、気が付くと 7 つものパラメータを持つ重量級の機能コマンドになってしまいました。でもコード進行は現代音楽理論の主軸なので、しかるべき結果かも知れません。転回機構

は今年の 4 月頃、百元氏や大塚氏から要求されてから、実に 14 ヶ月に渡って暖め続けてきた機構です。特にベース音やテンションをどう扱うか迷い続けました。それ程大袈裟なものではないかも知れませんが、私としてはしっかり熟成させたいつもりです。また転回との絡みで、単独記述できた方が良いと考えたコードを 2 つ加えました。

展開マクロ側ではできるのに、定義マクロ側で繰返し回数が指定できないという非対称な文法も改善しました。無名マクロを実装した頃から気になってしょうがなかったのですが、マクロ解析の複雑さを言訳に後回しにし続けていました。

更にもう 1 つ。無音の音符で鍵盤アニメーションを実現するという驚くべき発想のデータも作られるようになり、その対応策を施しました。強弱 v0 でも鍵盤表示を行うようにすると共に、MIDI データを送出せず音源負荷を与えぬようにしました。これで、v0 を使用する限り、どんな過激な鍵盤の乱舞でもモタレが生じません。

皆さんから寄せられた Muse データも何百という数に上り、10 ヶ月前に不安げに公開した頃の心境を思うと、感激で気が遠くなりそうです。そして、「使い易い」「楽しい」「感動した」「凄い」「素晴らしい」「驚いた」「夢が叶った」という皆様からのお褒めの言葉が、気を失いかけた私の背中を押し続けるのです。

《Ver3.6 開発後記》2000.07.23

「音楽」という言葉を“音を楽しむ”あるいは“楽しい音”として創出した先人たちのセンスには敬服するものがあります。皆さんから「Muse は実に楽しい」との有難いお言葉を多く頂いておりますが、多分それは音楽の本来の楽しさを阻害する、派手なだけの機能を実装しないように心掛けた結果だと振返っています。

その意味で、今回はちょっと大がかりすぎたかなと感ずる機構を実装してしまいました。少々小難しい表現をさせてもらおうと、時空に描かれた絵画である「音楽」を平面に定着させる、と言った所でしょうか。この絵巻物の様な長い絵画を、高速にスクロールさせる処理に結構苦労しました。しかし何より苦労したのは実装ではなく設計です。五線譜のイメージをできるだけ残し、Muse のデジタルなデータを如何に表現するか。そのデザインを決める作業は、七転八倒の日々でした。結局、でき上ってみれば誰でも思い着くような結果かも知れませんが、でもそれでこそ自然で無理のない仕様の証なのだと考えています。

“ピアノロール”っていうのは結構ありますが、“五線ロール”はちょっと希有なんじゃないか、そして全メンバーが重なっているなんて見たことがないと 1 人悦に入っています。メンバー色一覧のスイッチに連動させるということに気づいた時がブレスクルーの瞬間だったと回顧しています。それにしても、想像以上に綺麗な“音楽の絵”が展開され、自分としては大満足です。

もう 1 つ、文法強化も行いました。名越さんに背中を押されて、音量 V も相対指定できるようにしました。この機構は、ずっと以前に友人の田中君からも受けていたのですが、事を複雑にするだけなのではとの気が払拭できず、今に至っています。でも、V で各メンバーのバランスを取った状態で一気に全体の音量を変化させたい、という主旨をお伺した瞬間、実装する気になりました。

さてさて今回の譜面モタ。作り始める時はやりすぎかなと

思っていたのですが、自分で実際に使ってみるとタイミングずれのデバッグに大変強力な機構であることがわかりました。まさに“一目瞭然、百聞は一見にしかず”とはこのことです。また、mid2mus と併用すれば一般 MIDI ファイルの疑似譜面による確認ルートも開拓できた訳であり、他の MML コンパイラとは一線を画し、さりとして市販のシーケンサーとも違う、実にユニークなツールに成長したと言っても許されるでしょう。とは言うものの“楽しい”のは「Muse」ではなく、それが対象にしている「音楽」であるということには違いありません。

《Ver3.7 開発後記》2000.08.10

半年以上前に Ver2.9 で思い止まった NRPN の実装を、まさかやることになるとは思っていませんでした。前回の“譜面モニタ”に匹敵する重量級のバージョンアップです。しかし「もしやるとしたら...」と空想の世界で設計を継続していたので、思いの他、実装は短期間で完了しました。綿密な設計がいかに実装を効率的に押し進めるかを、改めて実感しました。

3つの特性に含まれる8つのパラメータをどのような文法書式で表現するか。文法互換と Muse の基本理念を維持しつつ実現するにはどうすべきか。波形加工の確認作業をどのような機構により簡便に行わせるか。またそのウィンドウはどうあるべきか。楽器群とドラム群で統一感を持たせるにはどうしたらいいか.....。3つの特性が、3つのエフェクト(R, W, Q)に相関があることに気づいた時、美しい文法が閃きました。苦行の混迷状態から、突然天空に舞上がる至福の瞬間です。

前回のバージョンは限定リリースの形態を取り、しかも応募者の皆さんから有意義な意見を掲示板で頂きました。それらの感想をもとにいくつもの手直しを施しました。薄墨色の縦棒で現在の演奏箇所をリアルタイムに表示すること。演奏自動追尾の譜めくり機構を実装すること。情報系ダイアログ3種に関して、前回の位置と寸法を初期化ファイルに記憶し再現すること。特に自動追尾に関しては、皆さんからの強い要望がなければとてもやる気にならない程、厄介でした。

私自身のアイディアも盛り込みました。譜めくり[T]ボタンの等価キーをスペースからタブキーに変更し、鍵盤ウィンドウ側のスペースキーによる演奏制御との競合を回避しました。また、[A]ボタンを新設して自動譜めくりの楽しさと、ピンポイント・リロードの使いやすさを両立させました。

さて、話は波形加工に戻ります。これを開発する契機は、Fitさんが公開された MIDI データです。Muse で作った MIDI データに他のシーケンサーでリリースコントロールを付加し、実に伸びやかな残響を与えていました。GS と XG で共通してこの効果を得るためには、棚上げし続けていた NRPN に手を出す必要がありました。できしてみると、まるで Muse がシンセサイザーになったような感じです。しかし、ピッチベンドさえうまく使いこなせない私は、この波形加工の機構はとても操れそうにありません。ミュージアの皆様が縦横無尽に活用されることを祈るばかりであります。

《Ver3.8 開発後記》2000.09.29

2つの機能追加と3つの変更を行いました。機能追加の1つ目は、起動直後に主なサブウィンドウを表示させる指定を可能としたことです。私は元来ズボラな人間なので、終了時に次回起動のことに気を遣うなんて性に合いません。散らかしたウィンドウが次回も出てくるなんて耐えられないので、初期化ファ

イルではなく起動引数形式にしました。これで、データ作成用と音楽観賞用のショートカットを別にして管理することもできます。自動譜めくりのスイッチも付けました。

機能追加の2つ目は、前回実装した波形加工を果敢に利用しようとして疑問を感じた Fit さんからのメールがトリガとなりました。波形加工の試聴で調整した音と実際の演奏音が異なるように感じるという内容でした。よくよく調べてみると強弱 v で周波数特性の効果がかなり変わることがわかりました。そこで波形加工のウィンドウに強弱 v の調整レバーを付け、その効果をその場で確認できるようにしました。ひとえに Fit さんのお陰です。ありがとうございました。

変更の1つ目は、メンバ色一覧などの各種サブウィンドウにおいて、チェック済メニューを再度選んだ際、従来それらを閉じていましたが、前面にポップする様にしたこと。ノートパソコンなど解像度の低いモニタでは、ウィンドウ同士が重なり、隠し合ったりしますが、その様な場合に作業効率が高まると思えます。

2つ目は、譜面モニタにおけるマウス右クリックのリロードの際、演奏開始位置はモニタ表示上の最左端からでしたが、マウスクリック位置からとしたこと。表示半ばから再演させたい時、一々譜面モニタのスクロールをせずに済みます。

そして3つ目は、一見すると機能低下に思われるかもしれない変更です。「音源の選択」メニューにおける GS/XG の選択ラジオボタンがなくなり、ついでに不正確な情報が得られなかったいくつかの欄も撤廃しました。音源タイプ選択のボタンは、両音源のドラムマップ差を吸収するためだけに存在していました。しかし XG 系音源には TG300B というモードがあり、それを利用することで GS 互換となることを知ったのです。DTM をやっている方にとっては常識的なことかもしれませんが、恥ずかしながら私はこんな事も知らずに Muse を開発していました。

この発見の経緯は、草場さんが *DRUM (ドラム転向コマンド) がうまく効かないという不具合報告をしてくれたことに端を発しています。それに触発され、岡本さんが各音源に関する調査をしてくれました。そして、前回の NRPN 実装に伴って、GS/XG の初期化をはじめたことも相まって、TG300B モードに行き着いたという次第です。草場さんのご協力には大変感謝しております。また、岡本さんの精緻な調査レポートは、Muse をより高い完成度にしたという情熱のようなものが感じられました。Muse に成り代ってお礼を申し上げます。まるで2人と共同開発している感覚に襲われた日々でした。自前のドラムマップ変更処理を排除したことで、副次的に演奏モタレが軽減したし、XG 音源でも SFX ドラムが使用可能になりました。本当にありがとうございました。

「音源の選択」ダイアログが小さくなって、機能が減ったように見えますが、それは使用準備として指定しなければならない項目が減っただけのこと。利用者にとっては、音源差を気にすることなく気軽に使用できるアプリケーションになったと確信しております。私は常々“多機能”と“高機能”は本質的に違うと思っています。一般的にはバージョンアップに伴って必ずボタンや欄が増えていくのが普通ですが、それらが減るバージョンアップがあっても良いのではと思っています。否、むしろそれこそが“多機能化”ではなく“高機能化”への道なのではないでしょうか。目的が果せるのなら機能は少ないほど美しいと思うのです。

《Ver3.9 開発後記》2000.10.21

新しいタイプの音源には、更に拡張されたドラム音が存在する事を知りました。従来、ドラム音域にガードをかけてエラーメッセージを出していましたが、少なくとも Muse 鍵盤の範囲ですべて通すようにしました。これに伴い、ドラム試聴のダイアログのボタンフェースも [No Assignment] を減らして実名化しました。残念ながら、私の持っている音源では鳴らないのですが...

楽器側でも発音可能な音を増やしました。前回の TG300B モード採用を踏台にして、とうとうバンクセレクトまで食指を伸したのです。TG300B モードがバンク制御に関しても GS/XG を共通化させることに気づいたからです。今回のバンクセレクト実装で、ついに Muse は音源の持つほとんどの音を鳴らすことができるようになりました。しかし実装は大変でした。バンクは歯抜け状態で不規則に並んでおり、しかも音源モデルによってそれがまちまちなのです。

最も厄介だったのは、未定義のバンクを選ぶと通常の楽器選択さえ効果しなくなるという音源仕様でした。これでは、バンクを駆使した MIDI データは、再生音源を限定してしまいます。これを回避するため必要に応じてバンクゼロを送出し、未定義パリエーションが指定されても、標準音色が鳴るようにしました。不要なバンクセレクトのイベントを極限まで除去し、しかも標準音色が出るように必要最低限でバンクゼロを挿入するアルゴリズムはかなり神経を使うものでした。ただし残念ながら、この標準化機構も音源によっては限界があるようですが...

多分通常のシーケンサーでは、このような MIDI 音源特性に気を使いながら、データを打込んでいくのだと想像しています。しかし、Muse はその煩わしい作業をすべてプログラムに押し込めることに成功しました。メンバー属性 P にパリエーション番号を添えるだけで、後は Muse がよろしくやってくれます。利用者は“MIDI”ではなく“音楽”に専念できるわけです。これこそ Muse のポリシーなのです。

私自身が Muse で何を実現しようとしているのか、言葉が見つかり始めています。私は“MIDI シーケンサー”を作ろうとしているのではなく、“音楽シーケンサー”を作ろうとしていたのです。MIDI の機械語的な命令セットを隠蔽し、より抽象度の高い文法を編み出そうと常に心掛けてきました。Muse にとって MIDI は発音機構としての手段にすぎず、目的はあくまでも「音楽」なのです。先日、むらったさんから素晴らしいお言葉を頂きました。「私にとって Muse は最高の楽器である」と。

《Ver4.0 開発後記》2000.12.06

{とう}2 Muse もバージョン 4.0 まで到達してしまいました。{ちび}2 と機能強化をしては{しゃあ}2 と版数を上げていく自分が{少}2 恥ずかしい思いです。そしてついに初版リリースから 2 年も経過してしまいました。あつと言う間でしたが、Muse を通じ魅力的な多くの{方}2 と知合いになれたのが大きな収穫でした。これからも末永くお付き合い頂けると幸いです。

今回も性懲りもなく、細々とした機能強化です。音源の選択をメニュー化して簡便に切替えられるようにしたり、それに引きずられてメニューレイアウトを見直したり、フィンガー拍数のメンバー属性を選択できるようにしたり、それに伴って最終テンポも表示させたり、譜面モニタの加線をグレーにして視認性を高めたり、右ボタンリロード&ランの反応領域を拍子エリ

アまで拡張したり、ドラムの響きを向上させるために DRUM コマンドでアサインモードを指定できるようにしたり、連符内の連結に関する障害を取り除いたり、波形加工データでエクスポート MIDI が間延びしてしまう障害に対応したり、出力する MIDI ファイルのセットアップ小節を定格化したり、テンポの下限値を %1 まで拡張したり、楽器の試聴で 9 番パリエーションの試聴も追加したり、コントロールキーを併用したカーソルキーで MARK や STOP の位置決め機能を付けたりしました。こうして、のんびりと成行き任せに階段を上っていくのもフリーソフトならではの。しかも、ミュージアの皆さんと一緒にですから、なおさら愉快的な旅です。現に、上記項目の一部は、名越さん・ばるさん・FRANZ さんたちに、障害レポートを頂いたり、開発のきっかけやアイデアを与えてもらいました。本当にありがとうございます。

Muse とはゼウスの 9 人の娘で、芸術や学術を司る女神たちの総称であり、Music の語源でもあります。もちろん、この事を意識して命名したのですが、先日あるホームページで面白いことを知りました。実は、彼女たち自身が芸術を遂行するのではなく、人々に靈感(インスピレーション)を与える女神たちなのだそうです。それはまさに、私の Muse が果そうとしている事にピッタリではありませんか! 更に、単人称でないところが、ミュージアの皆さんとの活動さえ象徴しています。

いよいよ 3 年目に突入する Muse。果してこの女神たちに、今度は何が起るやら。実のところ私自身、皆目見当が付きません。新世紀も成行き任せの旅は続きます。

《Ver4.1 開発後記》2001.03.01

今世紀最初のバージョンアップは、門出にふさわしく“初期化”に関する改良となりました。Muse の初期化のためのイベントは最大でも 0.3 秒以内に納めるのですが、MIDI を譜面化するソフトに対応するためには、最低でも 1 小節を与えて、区切りの良い出だしにする必要がありました。しかし、そのために MIDI 再生時に 2 秒ほどの待ち時間が発生してしまうという代償があったのです。携帯着メロのデータ作成に活用されている方や、ゲームの BGM に活用されている方から、この件に関するメールを頂いており、無限ループ形式のこれらのデータでは、この 2 秒がとても気になるのご指摘を頂いていました。

今回、テンポメタイイベントを調整用に差挟むことで、小節の区切りを崩すことなく実時間を圧縮できることに気づきました。今まで 2 秒あったセットアップ小節のインターバルを約 10 分の 1 に圧縮でき、これで譜面化と出だしの空白圧縮という両者の要件をほぼ満たす状態に持込めたと思います。

更に、偶然にも全く違う方向から初期化改良のきっかけとなる一報が届きました。Nam さんの新しい音源で、ドラム転向が機能しないというご報告です。mid2mus による Nam さん自身の調査の結果、音色指定 P をしないと転向が起らないという音源固有の問題であることがわかりました。しかし、音源固有の問題であっても、再び同様の症状がどこかのミュージアで起るとも限りません。そこで、これを回避する初期化処理を施しました。こちらでは再現しない症状でしたが、Nam さんの確かつ速やかな調査報告のお陰で効率よく対処できました。心よりお礼申し上げます。またこれを期に、裏技的な GM と GS のマルチリセット方式を撤廃し、GS リセットのみとしました。これは前述のインターバル圧縮にも貢献しました。

実は、私自身は MIDI をエクスポートして活用することが

殆ど無く、ましてや未知の音源ともなればなおのこと、上記のような症状を発見することはありません。掲示板やメールでミュージアの皆さんとのやり取りがなければ、Muse もこれらの改良をすること無く一生を終えていたことでしょう。まさに Muse は、ミュージアの皆さんのご意見ご感想で「呼吸している」のだと感じる春の日であります。

《Ver4.2 開発後記》2001.04.16

プログラム開発は、各フェーズそれぞれがエキサイティングです。まだ影も形もない時期に頭の中で次第に形にしていくフェーズ。実際にコーディングを始めて、処理上の工夫や新しいアーキテクチャを学んでいくフェーズ。そして完成した後に、いろいろな発想で改版していくフェーズ。更に、一般には嫌われていますが、私にとって最も興奮するのがバグと戦うフェーズです。

バグは無いに越したことはないのですが、その戦いの中で学ぶものも多いのです。私は、障害を追いそれに対峙している時、自分が別の人物になっている様な錯覚に陥ります。ある時は難病を治そうとしている外科医。ある時は、限られた情報から真理を見出そうとする科学者。ある時は犯人を次第に追いつめていく刑事。そしてある時は、すべての状況を客観的論理的に判断して、判決を下す裁判官。所詮自分がまいた種ですから、あまり偉そうなことは言えないのですが...

今回の改版内容は、その大半がバグ対処です。一番厄介な障害は多重起動の抑止部分で起ったものでした。起動タイミングの関係で、まれにウィンドウが半死半生の状態になってしまうのです。*p オプションも併用すると、MIDI デバイスまでロックすることがありました。これは致命的な障害です。しかし、私の環境ではあまり起らないこともあり、原因にたどり着くまでにひどく時間がかかってしまいました。しかも原因がわかってからも大変でした。どのようにシーケンスを組み直しても、過激な連続起動をするとゾンビのようなウィンドウが出現してしまうのです。多重起動抑止の際、今までの Muse は、既存のウィンドウを強制終了させて、後から起動するタスクを優先する方法を取っていたため、どうしても無理があったのです。

結局、根本的に方式を変えることにしました。素直に後から起動した方が身を引く方法です。この方式にしたことで、かなり強靱になりました。数々の Muse 関連ソフトが *p を利用しているので、このオプションと *e だけはタスク間通信で受け渡すようにしました。ただし、*f, *m, *s, *a など、他のオプションはあえて継承しないようにしました。加えて、ゲージマーク表示、譜面モニタの拍子・小節調整などを維持するようにしました。これにより、Muse 関連エディタの連携において、操作性が向上すると考えています。

上記以外にもいくつかの障害報告を頂き、それなりの対処をしました。現時点でほぼ潰せていると考えています。しかし、音源固有の特性によるものや、当方で再現性が得られないものなど、一部の方の症状は完治させることができませんでした。今後とも継続的に調査しますのでご容赦下さい。考えてみると、私の生み出すものは、私自身を越えることはできません。そして、私自身が完璧ではありませんので、当然のことながら Muse も完璧ではない訳です。まあ、多少持病があった方が長生きすとも申します。今後とも末永くお付き合い下さい。

《Ver4.3 開発後記》2001.07.07

Muse にはバージョンアップのインターバルが長かったのですが、長い割には文法上の機能を 1 つ追加したに留まりました。掲示板でミュージアの方から多くのアイデアを頂いたのですが、ほとんど実現できない自分が少々不甲斐ないです。

この所公私ともに多忙になり、楽道家の私も流石に精神的余裕がありませんでした。バージョンアップに相応しく、もう少し手を加えてからと思っていたのですが、ボヤボヤしているとまた何ヶ月も経ってしまうので、公開に踏切りました。

唯一の強化機能は、コード記述時の逆アルペジオです。和音記述との関係、特に複合記述時の整合を取るのに少々苦労しましたが、どうにかうまく行ったようです。本件は、アルペジオ開発時に気になっていた機能ではありましたが、それ程強い要求も無いだろうと思って実装しませんでした。今回、掲示板やメールを通じ、実はとても欲しかったという方が思いの外多かった事を知りました。

まるみぎさんのホームページに居候している Muse ですが、去る 6/11 についてアクセス・カウントが 10 万に達しました。思えば遠くに来たものです。たかみちえさんから素敵な Muse のイラストも頂きました。ありがとうございました。

今日は七夕です。七夕は雨の特異日と聞いたことがあります。今年の梅雨は雨が少なく本日も晴天。忙しさに追われて、最近夜空を見上げることなどありませんでしたが、今日は綺麗な星が輝きそうです。久しぶりに天上の音楽に耳を傾けてみようと思います。

《Ver4.4 開発後記》2001.11.18

既に Muse 文法の表現力は、音源の差異を吸収できるギリギリの所まで至っているため、そのポリシーから言ってもはや本質的な機能アップはなく、せいぜい操作性やモニタ機構、あるいは“音”とは直接関係のない簡易文法の強化が残されるまでかと思いきや、メニューを追加する程のバージョンアップを施すことになりました。実に 5 ヶ月近く間をあけた今回のリリース。それは「演奏会場の設定」のご提供です。

一般に出回っている MIDI ファイルを覗くとシステムエフェクトと呼ばれる設定が日常的に活用されています。また、MIDI の解説を行っているホームページでは、この機能説明が最終ページを飾る場合が多々あります。Muse もシステムエフェクトをサポートできたら、一端のシーケンサーとして“卒業証書”を頂けるかなと思っていました。しかし、システムエフェクトは音源による癖が強く出ます。それが実装への意欲を殺ぎ続けてきました。

ところが、ひと月ほど前のある事に気づきました。文法としてのサポートをせずプレーヤーとしての Muse にこの調整機構を持たせればいいのか!? 丁度、高級コンボがいろいろな音質特性を自分好みに調整するツマミを持っているように、Muse にも「My 音源」に対するシステムエフェクトの調整レバーを付ければ良いのです! こうして 15 個もの調整パラメータを持つ「演奏会場の設定」ダイアログが完成しました。同一の Muse データでも、石造りの教会で演奏させたり、大ホールでのコンサートとしたりできます。しかも、既存の MIDI ファイルを mid2mus で変換すれば、それを「My 会場」で演奏させる事さえできるのです!

しかしいざ出来てみると、15 個の設定値から自分好みの組合せを見つけ出すのは大変な作業でした。そのことはすなわち、気の利いた組合せには高い付加価値があるという事です。そう

なると、見つけ出した組合せをデータ化して、ミュージアム間でやり取りがたくなります。少なくとも私には、そういったやり取りの風景がとても楽しいものに想像されました。「私の自慢の設定を味わってみて！」「こんな会場はどうでしょう？」といった案配です。この風景の魅力は音源差に対するポリシーを乗越えるほど大きなもので、とうとう ROOM コマンドを立ち上げてしまいました。そして、ROOM コマンドとダイアログ設定のどちらを優先するかを決める「Fix スイッチ」を設けることで、始めに考えたプレーイメーと、ROOM コマンドでのデータのやり取りを両立させました。

システムエフェクトの MIDI コードを、初期化インターバル 0.3 秒の中に押込める事や mid2mus の対応に大変苦しましましたが、どうにかやり遂げました。手前味噌かもしれませんが、MIDI プレーヤーとしてもユニークな機構を提供できたことに大きな喜びを感じています。さてさて、これで果して Muse は、シーケンサーとしての“卒業証書”を頂くことができるでしょうか？

《Ver4.5 開発後記》2002.02.02

初版をリリースしてから丸 3 年が経ちました。先日、Muse の機能強化項目を書き散らした“検討リスト”を整理し優先度を付けてみました。強化項目自体がもう数える程しかないだろうと高をくくっていたのですが、なんと 100 を越える項目が列挙されているではないですか！これは“卒業”なんて言ってもらえません。このまま行くと冗談抜きで“ライフワーク”になってしまうかもしれない（笑）

今回の強化は、視聴覚それぞれの表現力に関するものです。聴覚に関する方の強化は“グループ感”の指定です。もうかなり前から、スタッカートに対するいくつかの要望を頂いていました。音符 1 つ 1 つに指定するのではなく、v や o のようにそれ以降に一律効果する指定方式も欲しいとか、短くするだけでなく長くする指定も欲しいとか、それらの値を定量的に指定したいとかです。

しかし、単純にそれらの要求を個別に実装しては Muse 文法としての統一感が崩れてしまいます。そもそもこれらの要求の背後にはもっと奥深いものがあると感じていました。それらをまとめ上げる 1 つの鍵が“グループ感”にあったのです。音の出だしと音の切れ目における微妙なズレを調整することで、デジタルで正確無比な音楽を、躍動感あるものにすることができると。スタッカートという奏法は音の切り方におけるグループ感の、最も特徴的な表現の 1 つだったので。

こうして、p, q という一対のフィンガー属性を設定することになりました。音の切れ目だけでなく出だしに対しても対称的な扱いができるようになり、高い抽象概念のもとにまとめ上げることができたと思っています。ただし外部仕様は統一的でも、コンパイルアルゴリズムに要するデータ構造は p と q では大違いでした。今まで前方のみ処理していた解釈が p によって初めて過去に遡るようになったのです。おかげで新しいポイントを新設しなければなりません。未来と過去では大違い。開発中、時間に関する哲学めいた気分がささってしまいました。

更に、新しいフィンガー属性値である p, q を「フィンガー拍数」ダイアログのフッターに表示するようにしました。ついでに今まで表示しきれなかった波形加工の属性値や調性の状態も添えました。でも一番苦労したのは“グループ感”の概念説

明に当てる“言葉”でした。当初は「音頭」と「音尾」にしたかったのですが「音頭」では「盆踊り」みたいになってしまいます。日本語は難しいです（笑）。結局、「出音」「止音」という造語を作りだし、説明することにしました。

視覚に訴える機能強化はフォント指定です。T.Monma さんや MIZ さんをはじめとするミュージアの皆さんは、テキスト領域というたった一行のエリアを縦横無尽に活用し、テキスト・アニメーションという表現技法を創出してくれました。しかし、今まで Muse で使用できたのは、プロポーションな「MS P ゴシック」というフォントのみでした。この制約をどうにかしたいと常々感じていましたが、フォント再現性の問題から今一つ実装する気になれなかったのです。でも今回、皆さんの熱き期待に後押しされて、FONT コマンド実装に踏切りしました。

Windows はそのフォントデータのインストール状態によって、フォントの再現性が変化するアーキテクチャです。この事は Muse でデータを作成する際、再生するパソコンの状況を意識しなければならぬ事を意味しています。そこで、始めは単純に非固定ピッチと固定ピッチの切替サポートだけにしようと思ったのです。しかし、今後のことを考えるとそのレベルの表現力ではすぐに物足りなくなってしまうのではないかと考え直しました。思えば、著作権フリーのフォントデータも出回っています。もしかしたら Muse のためにフォント自体も作成し、それを添えて提供するパワーミュージアムも現れるかもしれないと考え、フォント名指定も可能にする仕様としました。もちろん、太字・斜体の修飾付きです。

できると、今度はフォント自体を選び出すサービス機構が欲しくなりました。メモ帳など他のアプリケーションでフォント確認をしてもらうという運用案も考えたのですが、やはり Muse 自身で自律的に確認できた方が格段に便利です。こうして「フォントの確認」ダイアログまで立ち上げてしまいました。

さてさて、今回の 2 つの表現力強化。これらを活用してミュージアの皆様はどんな素晴らしい“芸術”を創出してくださるでしょうか。今から瞳を輝かせて待っている 3 歳児の Muse なのです。

《Ver4.6 開発後記》2002.09.06

今からおよそ 2 年半前、Hashiita さんから頂いた初めてのメールを今でもはっきりと覚えています。「小学校の頃、音楽の成績は“2”以外とった事がなく（笑）、音楽的知識がほとんど無かった自分でも充分楽しめる『Muse』は素晴らしいです」彼は、その自嘲表現からは想像も付かない程の高品質なデータを次々と投稿してくれました。その当時、彼から受けた要望が「現在の 16 色より色数をもっと増やし、各メンバーの色を指定することはできないか？」というものでした。殿堂の登録曲がまだ数十件程度だった頃のお話です。登録曲が 1000 件を越えた今、やっと 2 年越しのご要望に答えることができました。COLR コマンドのご提供です。

始めはコマンドではなくメンバー属性とし、どのようなタイピングでも色の変化を可能にしようと思ったのですが、譜面モニタにおける処理があまりに煩雑になるため断念しました。まあ今回の仕様でも、Muse 演奏時の曲想イメージ表現が文字通り多彩になると考えています。ぜひ、お好みの色で演奏して下さい。

現在の Muse の文法仕様は、遅延音長は省略音長が使用でき

ないはずなのですが、コンパイル処理のガードが甘く、部分的省略でコンパイルが通過してしまうという不具合に気づきました。今回は、きっちり文法チェックを行うよう改善しガードを堅くしています。それに伴い、現在までに登録されている殿堂曲の全点チェック作業も行いました。1000曲を超えるデータなので気が重かったのですが、*o コマンドを利用してバッチで一括チェックすることで比較的容易に作業を終えました。結果として17曲でエラーを検出しました。まるみぎさんに差替を依頼済みです。ちなみに、その中の1曲はなんと私自身のものでした(苦笑)

YSTK さんのご要望で微分音長の範囲拡大も施しました。従来は i60 までのサポートでしたが、i960 まで拡張しました。いろいろな局面で音長比率による指定を施す場合があり、その際の把握のし易さというのが主旨です。この手の改良は比較的容易で、しかも効果もありそうなのでさっそく採用いたしました。YSTK さん、ありがとうございました。

Muse のコンパイラは、極力音源に負荷を与えないように無駄なデータをカットする処理を組み込んでいます。例えば、同じ音量値の属性 V が連続して何度も指定されていた場合に、一番初めの V だけ残しあとはすべてカットするなどです。これらの処理は今まで完全ではありませんでした。気にはなっていたのですが、複雑な手順を踏むため改修するのをためらっていました。今回、腹をくくって手を入れました。久々の改版だったため、自分のソースの読取りハビリから始める必要があり、実に辛かったです。でも、出力される MIDI は更にシェーブアップされ、コンパイル速度も多少ですが向上したはずです。

Hashiita さんのメールの言葉を思い出す度に、どんな人でも本当は何かしら素晴らしい才能を持っており、既成のものさしではその可能性を推し量ることができないのだという思いを強くします。今回のバージョンアップは前回からかなり間が空きましたが、Muse がその秘めたる才能を開花させる一助となりうる限り、開発を継続していこうと思っています。むろん頂いた諸々の要望の実現は、今からおよそ2年半後になるかもしれませんが(笑)。

《Ver4.7 開発後記》2003.02.12

ついに、最後の大文字 X を組込む時がやってきました。残されたこの一文字を何に使うべきか、ずいぶん長い時間考え続けてきました。そして今回、最も汎用的で最も相応しいと思える機能を割当てることが出来たと思っています。Muse 文法を考える際、MIDI の規格を勉強しないと使えないというツールにならないように心掛けてきましたが、今回の X 指定でそれが多少崩れた感も否めません。しかし、それを補って余りある効果を出すだろうと想像しています。

私は、昔から「一石二鳥」な行為が大好きで、今回の X 指定一発で三鳥ほど捕まえたと思っています。実に気分がいいです(笑)。一羽目は、「MIDI のボリュームコントロールでバランスを取りたい」「抑揚を付けながらフェードアウトしたい」というご要求 X7 をお使い下さい。二羽目は、Y 指定のような ON/OFF ではなく「ハーフペダルを使いたい」というご要求

X64 をお使い下さい。そして三羽目は、Muse の U 指定よりも細かい「1セントレベルの微分音程を指定したい」というご要求 X129 をお使い下さい。

どちらかというと MIDI 規格に傾倒した X 指定ですが、X128 と X129 は Muse らしい工夫を施したと思っています。また X

指定は、私もまだ知り尽くしていない音源機能の駆動を可能とするので、未来の鳥も打ち落せるかもしれませんね(笑)。

岡林さんから、*COLR コマンドの複数メンバー指定をする文法記述のアイデアを頂きました。Muse 文法の統一感を崩さない美しい案です。早速採用させて頂きました。YSTK さんからは q 指定に関する新しいアイデア、固定音長指定を頂きました。対称性を考えて、p 指定にも与えるか否かかなり悩んだのですが、両方同時に指定すると解釈不能になる場合があることに気づき、q 指定のみとしました。

草場さんから Windows2000 におけるハング報告を頂きました。その回避策も、今回施しています。また一つ Muse の品質が高まったと思っています。ありがとうございました。私自身は Windows98 ユーザーなので、かなり想像力を必要とする対処でした(ちなみに、あと1週間ほどで私も XP ユーザーになります)。

今まで、再生側パソコンに存在しないフォント名が FONT コマンドで指定されると、不愉快な文字化けを起していましたが、コンパイル時に再生パソコンのフォントを調査し、不在の場合はデフォルトフォントを採用するようにしました。コンパイル速度が低下するのを恐れましたが、十分な処理速度で対応できたようです。はじめはデフォルトフォントではなく「フォントの確認」ダイアログで指定されたフォントを採用するようにしようと思い、コーディングを開始したのですが、半角系のフォントを指定された場合、別の理由で起る文字化けが解消しないことに気づき、泣く泣く(本当は苦笑しながら)修正コードを廃棄しました。

最後にもう1つ、mid2mus の X 指定対応を行いました。久々の改版であったということもありますが、実はこのプログラミングが最も厄介でした。皆さん、「mid2mus 希望」とメールに書き添えるのを忘れなく(笑)。

《Ver4.8 開発後記》2003.05.25

2年以上前、私は「MIDI の機械語的な命令セットを隠蔽し、より抽象度の高い文法を編み出そうと常に心掛けてきました」と、この開発後記にて語りました。その言に嘘は無いのですが、今度のバージョンでは、そのポリシーを覆すようなコマンドを実装してしまいました。前回 X 指定を実装した段階で既にその兆候はあったのですが、今回は16進コードの列挙という実に機械語的なパラメータで、どうにも言い訳の仕様がありません。

しかし、皆さんから高価な MIDI 音源の話や、インサクションなどの凝ったエフェクトに対する要望を受けるたびに、Muse が音源や、ミューザーの皆さん自身の能力を最大限に引き出せていないことを思い、情けなさを感じ続けていました。特に、Muse というソフトを切っ掛けにして、高価な音源を購入して頂いた場合は尚更です(音源メーカーから販売促進料を頂きたくらいです(笑))。

今回実装した DATA コマンドは、当初はあらゆるイベントを出力させようと考えていました。しかし、リアルタイムメッセージは MIDI ファイルに埋め込むべきではないし、メタイベントで TPQN やセットアップテンポを記述されると全体の制御構造が崩れ大幅な改造を要するため、エクスクルーシブのみの対応にしました。

ここまでくると私も使い込みに自信がありません。自分であまり使わない機能は、品質上の目が行き届かないものです。

そこで、β版の評価モニタを募りました。Imahara さん・くさばさん・岡林さん・鳳凰蝶さんが手を上げてくれ次々とテストを実施してくれました。くさばさんは Readme.txt のわかりにくい部分を、岡林さんは文法に関するエラーチェックの漏れを、鳳凰蝶さんは OS の差異に関するレポートを、そして Imahara さんは寝不足と闘いながら途中再生における完全性の検査を何度もしてくれました。これらのレポートが無ければ、リリースの後に Muse のバグ対処バージョンが炸裂していたことでしょう。皆さん、本当にありがとうございます。

16 進を直に書く機構ではありますが、工夫した点もあります。POOL コマンドは全体の記述量をかなり軽減し、見通しを良くすると思います。また、今まで出力していた演奏開始時点の自動送出メッセージを抑止する DATA"" の機構がなければ、今回のエクスクルーシブ機構も恩恵が半減してしまうことでしょう。数値群を括弧でくると、チェックサムを自動計算し、その結果をデータに添える機構も実装しました。面倒な 16 進演算が不要になっています。

エクスクルーシブ対応の他にも、3 つほど改善を施しました。1 つは、マウスホイールによる譜面モニタの横スクロールを可能にしたことです。かなり快適な操作性を提供できたと感じています。もう一つは、WAVE コマンドを強化して、マルチメディア系のほとんどの音声ファイルを再生できるようにしました。これで Muse は、mp3 だって再生できるようになりました。本当はもっと早くやりたかったのですが、自分のパソコンを新しくしないと開発環境が整わなかったのです。

最後の 1 つは、MIDI エクスポートの際の「Generated by MUSE」の刻印を、著作権メタイベントから Instrument メタイベントへ移行したことです。著作権の項にツールの刻印をするのは出過ぎた真似だよなあ～、と前々から思っていました。Instrument は楽器名の項ですから、刻印として打って付けの記録場所ですよ。そうそう、あと 1 つ変更点がありました。添付の WAVE ファイル名を拍手.wav から applause.wav に変えました。将来、Muse が国際的になる日に備えて(笑)

さて、今回のバージョンで、Muse は音源の能力を確実に引き出せるシーケンサーとなりました。おまけに音声ファイル系の再生能力も高まり“音”を出すソフトとしては、とうとう盛り詰めたかなと思っています。ミュージアの皆さんの強いご支援がなければ、決してこの頂を踏めなかったでしょう。それを思うと、心が感謝の気持ちでいっぱい膨らみます。皆さんとの出会いに、乾杯です。

《Ver4.9 開発後記》2006.05.12

非常に長い間メジャーなバージョンアップを怠っており、改めて指折り数えてみたら丸 3 年も経ってました。自分でもビックリしていますが、バージョンアップをしなくても Muse の活動が粛々と維持されているという事実に、ミュージアの皆さんへの感謝の念を改めて強く感じました。本当にありがとうございます。

極たまに実施する Muse のイベントのために、特別版 Muse の開発はしていましたが、本格的バージョンアップとなると、3 年のブランクを前に緊張感が走りました。この開発後記を書くことさえ戸惑っているというのが正直なところです(苦笑)。少なりハビリが必要だと感じ、今回はコンパイルエンジンや演奏機構に直接触れず、ユーザインターフェース部分に対してコソッと改版することにしました。

2 点あります。1 つはテキスト表示部の背景色を、個々に設定している Windows プロパティのメニュー背景色と同色にしたことです。デザイン配色の観点から以前より手を入れたいと思っていたのですが、テキスト文字が固定色なため無条件でメニュー背景色にしてしまうと文字自体が目立たなくなってしまう。

この課題をクリアするために、メニュー背景色の明度を検出し、暗い色の場合はデフォルト色に移行するという制御を思い付きました。しかし技術的には、存外に苦労しました。何と Windows という OS は、従来の Windows スタイルと新しい XP スタイルでは、メニュー背景色のシステム変数が異なるのです。例によって開発者泣かせのマイクロソフト仕様です。微妙に上位互換を崩しているこの設計思想に激しい虚脱感を感じつつ、まあどうにか裏技で両スタイルに対応したつもりです。もし何か不具合があったらお知らせ下さい。

もう 1 つは自分が Musings していてぜひ欲しいなあ～と感じた機能。譜面モニタ・メンバー色一覧・フィンガー拍数の 3 つのウィンドウを、一発で ON/OFF する機能です。従来の Muse は最大化ボタンだけがグレーアウト状態だったので、ここに割り当てる事にしました。通常の Windows インターフェースから逸脱してしまいますが、大目に見て下さい。各種ウィンドウがスクリーン上にパッと展開される感じは「最大化」として拡大解釈出来ないこともないでしょう(苦笑)

《Ver5.0 開発後記》2007.05.18

Muse のバージョンも 5.0 となり、初版から 8 年もの歳月が経過しました。初版開発当時の自分が何を考え、何をしていたか顧みるとその長い年月を強く感じます。今回のバージョンアップ項目は結果的に盛り沢山となりました。当初はそれ程欲張るつもりは無かったのですが、徐々に Muse のプログラムコードを目の前にし、強化を着手してみると、その先に利用者であるミュージアの皆さんの顔が浮かんで来て、あれもこれもと手を付けてしまいました。(昨年 9 月に Muse オフ会があり、生まれて初めてオフ会になるものに参加しました。実に楽しかった・・・という訳で、ミュージアの皆さんの顔がリアルに浮かぶようになりました)笑

まずは桁上がりバージョンに相応しく新メニューを加えました。履歴機構です。既に私の手元には 5,800 曲を越える Muse データが存在します。一応体系的にデータ管理はしていますが、ロードまでの手数がどうしても掛かります。そこで巷のアプリケーションのように、履歴を覚えていくことにしました。実装を終え、しばらく試験運用していると、一過的な履歴だけでなくそれをパーマネントに記憶するアルバム機構も欲しくなりました。しかし必要以上に煩雑な機能構造にはしたくありません。通常は履歴機構だけが見え、アルバム機構が欲しかった人が何かをすると初めてそれを享受できる。そんな思想で組み込みたいと考えました。アルバムの編集に関して、はじめは手の込んだダイアログを開発しようとも思っていたのですが、Muse 利用者のミュージアたちはテキスト編集はお手のものはず。ここはミュージアらしく、ダイレクトに履歴ファイルを編集してもらおう方針で行こうと決めました。お陰で、実にすっきりとした機能仕様になったと思います。ミュージアの皆さんが開発された Muse 支援ツールのように高度なアルバムの編集や再生の機構は持っていませんが、簡易版としては十分使用に耐える出来だと思っています。ぜひ

試してみてください。

圧縮解凍ソフトやメールの添付から直接 Muse データを立ち上げる場合、Windows のテンポラリフォルダからデータがロードされますが、その際 Muse のメインウィンドウタイトルが長くなりすぎて演奏時間が表示されないという不満が、少なくとも私自身には強くありました。私にはそのシチュエーションがとても多いのです。そこでちょっとした改善ではありますが、テンポラリフォルダは @ マークに集約して表示することにしました。更に今回新設した履歴も更新しないようにしてあります。次回アクセス時に「ありません」とエラーになるだけなのでテンポラリフォルダのデータを覚えても仕方ありません。こういった配慮はソフト設計でとても大切だと思うのですが、Microsoft の製品でも記憶してしまうようです。

かねてより要望のあった「常に手前 (T)」と「繰返し演奏 (R)」を、起動引数にて ON にするという機構を実現しました。更に、これもまた以前より要望の高かった、「データ編集 (D)」におけるテキストエディタの任意指定も可能にしました。この指定は、直接初期化ファイル Muse.ini に登録してもらうことにしています。ついでに履歴機構に対するいくつかの制御も、この初期化ファイルで行えるようにしました。こうなってくると将来、この初期化ファイルにいろいろなユーザ指定を背負わせる可能性があります。そこで、今回初期化ファイルの書式を大幅に見直し、ユーザ指定領域 [USR] とシステム管理領域 [SYS] を明確に可読できるようにしておきました。フォーマットが変わったので、新バージョンを立ち上げた時、初回はウィンドウの位置や選択音源がデフォルトに戻りますが、次回起動時から通常どおり運行しますのでご安心下さい。また、Muse の初期化ファイルは読み込み時の解釈を広く取っていますので、編集したことで立ち上がらなくなるといった悲劇が起こりにくいようにしたつもりです。過去の話ではありますが悪名高い win.ini のようなことはありませんので、果敢にお試し下さい。

2 メンバーをメロディパートに転向させるエクスクルーシブの存在を、草場さんから教えてもらったので、これを *DRUM コマンドとして実装しました。加えて、ソフト音源 VSC は、このエクスクルーシブを単純送信するとブルースクリーンになって固まってしまうという不具合の存在と、それを回避する方法も教えてもらいました。Muse コンパイラには、もちろんそのノウハウを埋め込んでおきました。こういう最適化があっただけで、単なる“ファイル変換”ではなく“コンパイラ”であると胸を張って言えるのだと常々思っています。Muse を使っている限り、文字通りブルーにならず音楽に没頭できますので、安心してお使い下さい。

フィンガー拍数の左肩にあった [偶奇] ボタン。単に 4/4 拍子と 3/4 拍子を切り替えるだけの影の薄い存在でしたが、今回このボタンを撤廃し譜面モニタの拍子切り替えと連動するようにしました。これで表示できる拍子のバリエーションが一気に広がりました。なぜ今までこの連動デザインに気づかなかったのだろう！これで、譜面モニタ・メンバー色一覧・フィンガー拍数が有機的に結びつく状態となりました。とても美しい構成です。もう 1 つフィンガー拍数に関して改修を行いました。あまりフィンガー毎に肌理細かく指定しない x 指定をフッター部に移動し、リスト部を (ほんのちょっとですが) スリムにし可読性を高めました。本当は邪道な？ 指定こそフッターに降るせ！ という声が聞こえてきそうですが、私自身が多用して

いるため、残留をお許し下さい (苦笑)

Muse の特徴の 1 つに、鍵盤上部エリアへのテキスト表示があります。これを活用したテキストアニメーションなど、Muse による表現の一大ジャンルになっているぐらいです。この TEXT 系コマンドのデータは、MIDI ファイルへのエクスポートの際、Lyric イベントで出力されますが、巷の MIDI プレーヤーではそれらデータの価値を発揮できず、単なる容量増にしかありません。そこで今回、エクスポートにおける保存のコモンダイアログを拡張カスタマイズして、最下部に「テキスト系コマンドのデータを出力しない」というチェックボックスを加え、オプション選択ができるようにしました。結果として期待したほどファイル容量は小さくなりませんでした。まあ MIDI 音源の負担を少なくする効果はあると思います。

長い時間放置したプログラムをまたメンテナンスし始めると、自分の開発したソフトであっても客観的な視点で検証することになりますし、またテスト段階でも、当時想定していなかったオペレーションをしたりして、新たな不具合を見つけ出すといった効能もある様です。今回もそんな不具合に遭遇しました。譜面モニタをスクリーンいっぱい横に広げた際に、その状態が初期化ファイルに記憶されないという不具合を見つけました。「ゲッ！ いろいろ手を加えたので改悪してしまったか・・・」と一瞬思ったのですが、なんと過去バージョンでも再現する現象でした。ウィンドウをスクリーンの四隅にフィットさせる機構との絡みがあり、この修正は結構難解だったのですが、どうにか手直しできました。

もう 1 つ。これも不具合改善の範疇だと思います。今回の履歴機構の導入で次々と曲をチョイスしていけるようになったのですが、直前の曲の残響が激しい場合に、そのコントロール効果が次曲に影響し、とんでもない演奏になってしまう症状を見つけました。この現象は、曲の切り替えをスピーディに行えば従来のバージョンでも出る症状なのですが、それが出にくかったのです。試みに症状の出るデータ群をエクスポートし、それら MIDI ファイルを使って Windows Media Player に連続ドロップする実験をした所、全く同じ現象が起きました。つまり、MIDI 音源のコントロールの本質的な部分に関わる制御なわけです。オール・ノートオフやオール・サウンドオフ、リセット・オール・コントローラなど、かたっぱしから送信を試みましたが事態は改善しません。MIDI 仕様のマニュアルをよくよく読んでみると「NRPN で設定されたバリューはリセット・オール・コントローラを受信してもリセットされません」とか「プログラムチェンジ、バンクセレクト、ボリューム、リバース、コーラスなどはリセットしません」と書いてある始末。これじゃあ“オール”じゃないじゃん。リセット・チョット・コントローラとでも名前を変えて欲しいです (笑)。冗談はともかく、こうして大変な試行錯誤を繰り返している内にある方法を思い付き、結果として解決に至りました。この対策の効果は素晴らしく、Muse 上で演奏を停止すると残響も含めてスバツと止まります。まるで竹を割ったかのような心地良さ。少々大袈裟かもしれませんが、Muse のプレーヤー性能は、Windows Media Player を凌駕したのかもしれませんが、とにかく、これで自信を持って Ver5.0 を皆さんにリリースできるというものです。

ううっ、今回の開発後記は今までの最長記録かもしれない (苦笑)。最近、改版頻度は低下気味ではありますが、これからも魂を込めて開発を続けていく所存ですので、ミュージアの皆

様今後ともよろしくお願いたします。8年経っても未だ開発項目の残リストは山のようにあるんです。

《Ver5.1 開発後記》2007.06.30

今回は近年まれに見る短いスパンのバージョンアップになりました。前回新設のアルバム機構。自分でも早速活用し始めたのですが、皆さんから頂いた演奏データは珠玉の作品ばかりで、あれもこれもとお気に入り追加してしまい、気づいてみたら総計 600 曲にもなってしまいました。まあ母数が 6 千曲近くあるのですから高々 1 割のピックアップをしたとしても膨れ上がるのは無理ありません。

そうなってくると 1 階層だけのアルバムでは整理が付きません。そこでアルバムの多段階階層化に臨むことにしました。どうせなら階層段数に制約のない構造に挑みたくになります。当然の事ながら再帰呼出しプログラミングになる訳で、しかもタイトル行とログ行が混在する実に難解な制御です。まあマクロ解釈のシーケンス程ではありませんでしたが、それでもアルバム機構のデータ構造を大幅に見直すことになり、久々にエキサイティングな開発でした。開発期間中は頭が冴え渡っていたような気がします。ボケ防止のために一生プログラミングを続けよう、なんて思ってしまいました。そんな訳でとても苦労したのですが、出来上がってみれば前回のバージョンよりもプログラムが構造的で、そこに数学的な美しい規則性さえ見出せる結果となりました。こういう結果を得る時にこそ「完成した!」という実感を味わうものです。一般的にその製品やスペックの美しさに弱りのある状態とは、まだ過渡的で未完の状態なのかも知れません。

本格的な階層が組み立てられるアルバムとなったので、タイトル部と登録曲とが入り乱れた状態になることも考えられます。そこで、タイトル部と登録曲の視認性を高めるため、登録曲の先頭にグレーの三角形マークを添えることにしました。プログラム内部のデータ構造のみならず見栄えもなかなか美しいのでは、と自負しております。

既にリリースしてしまった履歴ファイル形式ですので、当然ですが上位互換を意識しなければなりません。多分、ユーザーの皆さんの中には苦労して履歴ファイルを編集された方もいることでしょう。その人たちに「もう 1 回作り直して下さい」と言い放つ事はとても出来ません。上位互換を保って無限多段の階層を表現するにはどうしたら良いのかいろいろ考えました。結果として、* の個数で階層レベルを表現するという極めて単純な書式に辿り着きました。後で気づいたのですが、Wiki のセッション記述文法と類似しています。誰が築いた文法か知りませんが、まだ見ぬその人と妙な親近感を感じてしまいました。

基本的に、アルバム機構の編集はエディタにて外側で行ってもらう方針でしたが、やはり登録・除去という基本機能ぐらいは Muse 起動上から実施したいと、利用者である私を感じ始めました。さりとて専用のダイアログが出てきたり、操作の手数がやたらと増えるような形式にたくありません。最低限のストロークで望みの機能が実効する洗練されたユーザーインターフェイスにしたいと思いました。初めは、履歴メニュー上でマウスを右クリックすると編集用のメニューが出てきて、そこで機能指定するといったタイプにしようと思いました。ところがメニュー上でポップアップさせる方法がどうしてもわかりません。メニューがキャプチャをグリッしている間、ウィンドウ

メッセージをどうしても取得できないのです。この技術的な限界に直面し、別の切り口の操作仕様に思考を向けたとたん、閃きが起こりました。右クリックしたメニュー上の場所がタイトル部なのか演奏曲部なのかで、利用者が登録と除去のどちらを望んでいるのかを一意に決められることに気づいたのです。まさに「必要は発明の母、障壁は閃きの父」とでも申しましょうか。結局、1 クリックだけで登録も除去もできる快適な機能となりました。

アルバム登録時、それが登録済みの曲の場合は二重登録しないように考慮しました。しかし、この二重登録ガードに引っかかる場合、マウス右ボタン押下に対してまったくの無反応となります。利用者の指示に対してコンピュータ側が何もエコーバックしてこないのは、挨拶しても無然としている愛想の悪い人に出会ったようで実に不愉快です。さりとて、いちいち「登録致しました」というメッセージを出すのでは、意味も無く礼儀にこだわる執事に追い回されている様な堅苦しい気分になります。そこで対象曲のグレー三角マークを黒く強調するエコーバックを思い付きました。控えめでありながら手応えのある反応です。他ではあまり見かけない、ユニークなマンマシンインターフェースを実現できたと思っています。この開発で苦労したのは、その黒く強調したマークを元に戻すタイミングでした。Windows が管理している制御メッセージを解析し、独自ステータス管理でこっそり戻すという綱渡りをしています。

今回のアルバム登録・除去の機能提供に伴い、前回 Muse.ini のパラメータで提供した LGS (アルバムからの選択時に履歴を更新する/しない) の機構を撤廃しました。曲の履歴が残ること、誤った除去オペレーションをしてもすぐに登録し直せることに気づいたからです。履歴が、一種の UNDO バッファとしての役割も発揮するという訳です。一方で、そもそもアルバム中心で履歴エリアは不要と考える方のために、履歴メニューにおける表示曲数 LGM にゼロを指定できるようにもしました。加えて、Muse.log のセーブタイミングを Muse 終了時のみに変更し、データロード時やアルバムの登録・除去時の負荷を低減することにしました。

アルバム機構の強化が中心の今回のバージョンアップでしたが、あと 2 つ小さな強化を加えています。前回、テンポラリフォルダに展開されたロード曲は、ウィンドウタイトルを @ で縮退表示させるようにしましたが、hiro さんから所詮 Muse データを格納管理しているフォルダパスが長いと意味を成さず、演奏時間が確認できないことを指摘されました。当然と言えば当然なのですが、今まで誰からも指摘がなかったので、まあなんとかが大丈夫なのだろうと高をくくっていたのです。やはり付け焼刃では駄目ですね。対応をいろいろ考えた末、初期化ファイルにパラメータ DWT を新設することにしました。このパラメータ設定で、ウィンドウタイトルにフォルダ自体を表示しないようにすることができます。

もう 1 つは、アクセントという指定を追加したことです。これは長年実装したいと考えていて延び々々になっていた文法強化です。アクセント値には負値も指定できるようにしたので、いわば逆アクセントといった面白い演奏表現も可能です。開発上苦労したのは、和音やコードそして再現表記などの組み合わせ問題です。表面上の機能は単に一過的に音を強くするだけなのですが、複数の音が鳴る場合、元の強弱値に復帰するタイミングを制御するのは結構大変でした。仕様としては、アクセント量をフィンガー単位に保持し、省略記法も提供すること

で、今までの Muse 記法コンセプトに従った統一感のあるものにできたと思っています。完成したアクセント指定。特にパーカッションのメリハリに効果があるのではないかと私は期待しています。この新文法が、皆さんの Musing のお役に立てれば幸いです。

《Ver5.2 開発後記》2007.10.10

ついに MIDI の発音域をすべてフォローする日がやってきました。4 年前、DATA コマンドをサポートした際「Muse は音源能力を確実に引き出せるシーケンサーになった」と開発後記で豪語したのですが、まだこの極めて本質的な課題を積み残していました（苦笑）。初版の開発時点で、Muse は自らが考える音楽表現ツールとして設計したこともあり、オーケストラの音域がすべて入るピアノの幅で充分と判断していました。しかし、特に MIDI のドラムパートに関しては、Muse サポート外の音域で鳴る多彩な音が存在することを多くの方から指摘され、また一端のシーケンサーとしての地位確立のためにも、いつかサポートしなければなるまいと心に決めていました。

とは言うものの、全音域サポートの実装がここまで長引いてしまったのは、Muse らしいやり方でこの課題に対処することへの拘りがあったからです。単純に全音域にすると Muse 鍵盤をかなり横長にする必要があります。しかし、現在の Muse はサブウィンドウも含めて 800 × 600 に納まるように設計されており、そのポリシーを崩したくなかったのです。鍵盤表示はしないまでも、発音だけでも可能にするという案もありましたが、コーディングした文字列を視覚的に確認できる機構がないと Muse らしくありません。そんなジレンマの中で、長い期間あれこれ考え続けていたのです。今回鍵盤はそのままにして、譜面モニタのみ対応することで開発に踏み切りました。譜面モニタならばウィンドウ寸法が可変ですから、ユーザ環境に合わせて大きさを調整できます。加えて譜面モニタ上に鍵盤の音域を示す点線を表示する工夫も施しました。

実はこの開発が佳境に入った時、鍵盤ウィンドウの方も一鍵々々の寸法を一回り小さくすれば、600 × 800 に納まることに気づきました。しばし手を止めて熟慮しましたが、結局鍵盤ウィンドウに関しては、今の音域でのサポートが適切であると結論付けました。ほとんどの楽曲はピアノ音域を持つ Muse 鍵盤に納まります。にもかかわらず必要以上に鍵盤を左右に広げると、演奏時に色付く鍵盤が中央に偏り、鍵盤の両端はほとんど発音しないといった実にも間の抜けた演奏状態になってしまうことに気付いたからです。それは美しくありません。

音域拡張のプログラミング作業は思いの外難航しました。着手時点では単に音域ガードの if 文を外すだけだろうと安易に考えていたのですが、今までの音域を前提にした処理が様々な箇所にて点在していたのです。結果として、上位互換性に不安を感じるほど多岐に渡る改修となったため、本格的な互換テストを実施することにしました。今までに皆さんから頂いた 6 千曲の Muse データに対し、強化前と強化後の Muse でそれぞれ MIDI 変換を行い、バイナリレベルで一致を確認したのです。結果として、本件での上位互換を確認しましたのでご安心下さい。

テキストアニメ作成では右に出る者はいない、と言われている大御所 MIZ さんからの提言でフォント処理にも改善を加えました。指定フォントが PC 上に存在しない場合、今まではデフォルトの MS P ゴシックで表示していましたが、直前で有

効となったフォントを維持するという仕様に変更しました。本件は、残念ながら上位互換性を崩しますが、まあテキストフォントの表示体裁に留まりますので、お許し下さい。

実は、初めて FONT コマンドをサポートした当時、フォントが存在しない時のこの挙動をどちらの仕様にすべきか悩み、結局、選定理由の決定打が無いままにプログラミングが容易な方を選んでしまったという記憶があります。今回の MIZ さんの提案には「Musing の際、表示フォントの優先順を組み込むことを可能にする」という明確な根拠が存在します。思想に裏付けされた仕様は、美しいものです。また 1 つ、気骨のある意志が Muse に注入されました。MIZ さんに感謝いたします。

その昔、taka さんから左利き用のマウスカーソルも備えて欲しいと言われたことがありました。過去のメールを紐解いたら 2001 年、つまり 6 年も前のことでした。ずいぶんとお待たせしてしまいましたが、やっとご提供することができました。初めは左右対称のマウスカーソルをリソースに埋め込み、なんらかの指定で切り替えられるようにしようと思ったのですが、いっそのこと初期化ファイルでマウスカーソルのファイルを記述してもらい、如何様なカーソルでも指定できるようにしようと思立ちました。「左利き対応」が汎化され「カーソル指定」機構に昇華しました。アニメーションカーソルも指定可能ですので、お試し下さい。

もう 1 つ積年の課題をクリアしました。スタッカートや q 指定の音符に連結 & を効果させる件です。これは Muse コンパイルエンジンにメスを入れる改良なので、大変神経を使いました。しかも、今までの制約を逆手に取ったデータに関しては、微妙に互換性が崩れます。しかし、それを補って余りある Musing 効率を生み出すと考え、今回実装を決断しました。更に、これもまた互換性を多少崩してしまうのですが、和音内の & 記述に関する仕様も少々変えました。今までは、音尾部分の音長を調整することで和音内の連結を 1 音毎に制御することが出来たのですが、今回の改訂でそれが出来なくなったためです。何にしても今回の強化項目群は、古き棚に積まれていた荷物を整理したような気持ちにさせるものばかりでした。徐々に Readme.txt から制約事項の行を削除する快感を得ることが出来ました。

最後にご報告です。先月 (9/6) にかの有名なダウンロードサイト Vector に、Muse の専用カテゴリが新設されました。これもひとえに、ミューザーの皆さんの支えがあってこそです。本当にありがとうございます。これからも皆さんに「豊かな時」を与えることができるソフトを目指し、Muse の強化を続けていきます。

《Ver5.3 開発後記》2008.02.06

初版から継続してきたダイアログ名称「メンバー一覧」と「フィンガー拍数」を、それぞれ「メンバー情報」と「フィンガー情報」に変更する事にしました。初版の頃は単一機能であったため「名が体を表して」いたのですが、その後多くの機能が付加され、違和感を感じるようになったためです。考えてみるともっと早く実施すべきだったかもしれません。

その「メンバー一覧」改め「メンバー情報」に対して 2 つの強化を行いました。1 つ目は、楽器名にバリエーション番号を添えて表示するようにしたことです。これはバリエーションをサポートした頃から実現したかったテーマでしたが、パンク切替と楽器選択とが微妙に絡んだ処理ですし、シークバーを動か

した時もきちんと追従させなければなりません。その複雑さに今まで二の足を踏んでいたのです。しかし今回無事対応することが出来、リアルタイムにバリエーションも表示されるようになりました。また、バリエーション番号の表示域を確保しつつ、従来のウィンドウサイズを維持するため、フォントサイズを1段落としました。副次的に以前より引き締まった印象となり悦に入っています。

「メンバー情報」の2つ目の強化は、キーボードでメンバー毎のミュートを可能にしたことです。昨年は視覚障害を持つ優れたミュージアの方々とお近づきになる事ができました。特に横田さんはオフ会にも参加してくれました。ハンディキャップをものともせず、明るく前向きで活動的なその姿にとても感激しました。キーボードからの発音 ON/OFF 制御は、以前から実現を計画していたのですが、ダイアログからのキーイベントが取得できないという根本的な部分で挫折し続けていました。今回横田さんにお会いし、この構想のお話しをすると「それは便利そうです。ぜひ欲しいですねえ～」と微笑みました。その微笑が再度挑戦する気持ちに火を付けたのです。今度の挑戦で、フックを仕掛ける方式で成功することがわかり、とうとう実装に漕ぎ着けました。これでマウスという入力デバイスを扱えない方々も、メンバー毎の音の確認が簡便に出来るようになるはずで

す。初めこのキーによる ON/OFF 制御の仕様は、マウスクリックと同様に「トグル」で切り替えることを考えていました。しかし / の表示が確認できないユーザーにとってこの「トグル」方式は、その時点の ON/OFF 状況が掴みにくく、迷子になりやすい仕様であることに気づきました。そこで大文字なら ON、小文字なら OFF と「一意」な操作仕様にすることにしました。その他にも、Space や矢印キーによる演奏やリロードの制御を「メンバー情報」ダイアログから可能にしたり、不用意にウィンドウフォーカスを移動させないようにするなど、数々の配慮を施しました。皆さんの Musing 効率が高まれば幸いです。

「メンバー情報」以外にも大きな強化を実施しました。今まで何度もトライしては挫折し続けていたマクロの繰返し数ゼロ指定です。これが完成すると、*STOP"" を記述してそれ以降に定義マクロを連ねるといった構成上の制約から解放され、定義マクロをどんな場所でも自由に記述できるようになります。しかも Muse 文法としての審美感が高まる点も大きな魅力です。その事は何年も前から気づいてはいたのですが、定義マクロであってもデフォルトでまずは展開するという Muse の根本的なコンパイル・アーキテクチャが横たわっており、改造に着手しては失敗を繰り返していました。“0”と“1”では大違い。そこには論理処理としての大きな溝が存在しているのです。人類が自然数から脱皮し、ゼロを数学上の論理記号として活用するのに長い時間を要したのがわかる気がします。

今回、1度展開した中間解析リストを部分的に離脱するというアイデアが閃き、処理速度やメモリ効率の低下を起こすことなく、実装することに成功しました。その品質テストは、マクロの階層記述や、展開マクロ・定義マクロが入り組んだ複雑なパターンがあるので眩暈がするほど大変な作業でしたが、思いつく限りの検証をして完成させたいつもりです。後は皆さんに駆使して頂けたらと願わずには居られません。

MIZ さん、お待たせいたしました。波形加工遅延機構のご提供です。オフ会の折、「皆さん、どんな機能強化をお望みで

すか？」という私からの問い掛けに、いの一番にこの機構を挙げた MIZ さんの、控えめな声が忘れられません。そこには、実際の Musing 局面でこの機構を何度と無く必要とした経験が強く感じられました。しかし、波形加工は単独出力であっても NRPN と DataEntry を組み合わせ大変神経を使う処理です。そのため今までは、遅延による連続変化の実装になかなか踏み込めなっていました。Muse が複数の特性パラメータを一気に指定する文法なため、ますます難度の高いものになっていたのです。出来上がってみると、ある波形特性から別の波形特性に向かって、複数のパラメータを並列して変化させるという他のシーケンサーには見られない連続変化を容易に記述できる使用となりました。

現在譜面モニタには、[G][T][A]の3つのボタンが存在します。その内、[G][T]は既にショートカットキーが割り当てられているのに対し、[A]だけはありません。常日頃から気になっていたのですが、それ程優先度も高くないため放置してきました。今回はキーボード処理をいろいろと探索したので、ついでに本件も対応し[@]キーを割り当てる事にしました。[G]エンター、[T]タブ、[A]アットと、比較的直感的なアサインとなり、なかなか気に入っています(エンターは、Go!というイメージです。突っ込まないで下さい(苦笑))。

今回の強化テーマ群は、謀らずも永年挫折や放置を繰り返した内容で占められることになりました。普段の私なら、ちょっと試してみてもたすく諦めてしまったかもしれません。しかし今回は違いました。その気概を与えてくれたのは、他ならぬミュージアの皆さんの声や笑顔である事は疑う余地がありません。私一人が閉じた世界で開発を続けていたらある所で妥協し、決してこれだけの粘りは出せなかったことでしょう。皆さん、これからもよろしくお願いします。

《Ver5.4 開発後記》2008.07.11

今から2年前、2006年の夏にクールジョーさんが主催してくれた Muse コンテスト。先日ネットサーフィンをしていたら、そのサイトに辿り着きました。とても懐かしい思いで眺めていると、金賞を受賞した ratilt さんの受賞者アンケートに目が止まりました。クールジョーさんが用意した「Muse に期待する機能は？」の問いに対し、「譜面モニタ上で音符をクリックすると、テキストデータ上からその部分を探し出してくれる機能」と回答が認めてありました。

2年前もすべてに目を通したはずだったのですが、その時点では“ミュージアの自由選択に任された各々のテキストエディタと連携するなんて不可能に近い”と感じていたのです。しかし今読み返してみると、直接連携をしなくとも、行番号の表示だけで十分に Musing 効率が向上するだろうと気づいたので。これはとても便利そうです。自分自身が「心から欲しい」と思う気持ち。それは「魂を込めた開発」の必要条件です。その条件を手に入れた私は、早速開発に着手しました。

譜面モニタは極めて簡素な様相をしています。市販の音楽シーケンサーのようにボタンやスイッチ群が盛り沢山に陳列されていません。このあっさりとした表情の中に、秘めやかに気の利いた機能が実装されている姿が、私はとても気に入っています。今回もそのフィールを失わないように慎重に設計しました。算出結果はあくまでさり気なく、ウィンドウタイトル部に表示されます。検索モードへの移行オペレーションも、安易なボタンの追加をせず、SHIFT キー併用としました。本機能の

ために新しく立ち上げたマウスカーソルも、シンプルなデザインでありながら、クリック時のエコーバック感をしっかりと味わえるように工夫しました。

実装は結構重労働でした。SHIFT キーとの併用は従来機能との競合が起こらないように細心の注意が必要でしたし、フィンガー宣言や行番号を表示するために、内部の音符メモリ構造を強化する必要に迫られました。各音のデータ量を増やすことになり少々抵抗感はありませんでしたが、しかし今回の機能は、MML 型のシーケンサーにとって、それを補って余りある有効性があると確信しています。ちょっと気になった演奏ニュアンスを修正したいと感じた時、譜面モニタ上からテキストデータ側の記述位置を簡便に特定できることは、新しいタイプの情報流です。今までの Muse は、エラーメッセージを除き、基本的にテキストデータから Muse に向かう流ればかりでした。

先日、るうさんから試聴系ダイアログのタブキーや方向キー押下時の動きに一貫性がないという指摘を受けました。恥ずかしながらこの点に関しては、少々無頓着に開発しておりました。Muse 開発に臨む姿勢は、こういった肌理の細かい配慮を大切にしたいと日々考えていましたので、今回のるうさんのご指摘には大変感謝しております。そしてこの指摘は、単純な美感や統一感だけの話ではなく、視覚障害を持つ方にとって大変重要な操作仕様であることにも気づきました。今まで配慮が欠けていたこと、ご容赦下さい。今回、楽器およびドラムの試聴にて、統一感のある仕様を組み立て直しました。タブキーにてオクターブや楽器グループ毎に移動し、方向キーでそのグループ内の移動をするという仕様です。諸熊さんに査閲して頂き及第点をもらえたので、自信を持ってリリースできます。このように、すぐにレビューに参画してもらえる仲間がいるということはとても恵まれた開発環境だと思います。諸熊さんに心からの謝辞をお送りいたします。

ドラム試聴の改修が掲示板で話題になったため、空かさず MIZ さんから、予めからのご要望「ドラム試聴にてノート番号を表示して欲しい」の督促を再度受けました(笑)。むろん検討リストに記載しており、いつか実施しようと考えていたのですが、音名切替に連動するという仕様イメージから脱却できず、綺麗な設計が出来ずに今に至っていたのです。しかし今回、音名切替とは独立に表示させれば良いのだと気づいた途端、あっという間に設計が完了しました。何故にこれ程まで単純な事を今まで思い付かなかったのか、赤面の至りです(苦笑)。

毎度のことではありますが、新バージョンをリリースすると、直ぐに様々な不具合をミュージアの皆様に発見してもらい、ご報告を受けます。複数の目で検証されることの大切さ。そして、それにきちんと対応していく事で、Muse の品質が次第に高まっていく事を実感する今日この頃です。前バージョンも多くのご報告を頂きました。hiro さん、sana さん、木下さん、MIZ さん、BnKayak さん、楠本さん、Pascal さん、five さん、浅川さん、H.N.WPKIDS さん、ひまじん 925 さん……。すべての方をここにご紹介できないのがもどかしい思いです。Readme.txt の誤植指摘も含め、Muse が美しくあって欲しいという皆さんの期待を強く感じます。今後ともよろしく願いいたします。

草場さんが立ち上げてくれた MuseWiki もかなり充実してきました。草場さん、ありがとうございます。お陰様で、不具合対応のバージョン履歴管理がしっかり出来るようになりました。そこで今回のバージョンから、小数点第 2 位のマイナー

バージョン番号も Muse メニュー部に明示することにしました。管理がより厳密になるので、少々緊張している私がいま(苦笑)。

《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、1 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なりました。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、1 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、開発は苦勞の連続でしたが、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。中身の無いマクロが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、これからも Muse に愛を込めて育てて参りますので、皆さんよろしく願いします。

《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 10 年も掛かってしまった(笑)。そういう目で改めて眺めると、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ビ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、無駄なものも邪魔なもの。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 1 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います。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 2 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。起動直後は取って替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!

《Ver5.7 開発後記》2010.04.09

譜面モニタに関する機能強化を 2 つと内部処理の改善を 2 つ、そして文法解釈の変更を 1 つ行いました。譜面モニタ強化の 1 つ目は前回予告したテンポ表示の拡張です。数ある属性の中から、譜面モニタ表示で Musing が効率的になるもの、すなわち曲中指定の可能性が高いものを厳選し、それをポップアップで選択出来るようにしてあります。前回の開発を通して表示スタイルやフィンガー情報ダイアログとの連動等、その完成イメージの骨格は固まっていたのですが、開発は難航を極めました。全体属性であるテンポ、メンバー属性である音量、フィンガー属性である強弱指定など、それぞれが値の範囲や士記号に対する記法定義に微妙な差異を持ちます。それらに対応し統一感のある機能としてご提供するには、内部データの持ち方やフラグの意味をも改修していく必要がありました。改造は MIDI エクスポートまで波及しプログラムを大幅に見直す事になりました。

そんな苦難の改良でしたが様々な工夫を盛り込みました。前回は単純に 2 行間を交互に表示するロジックでしたが、テンポ以外の属性表示をしてみるとその仕様では重なって判読できなくなる状態が頻発。しかし機械的なシフトで闇雲に行数を増やすのは私の美観が許しません。そこで各文字列長を算出し余裕のある場合は 1 行目を優先しつつ、混み合う際には複数の行に分散していくといったインテリジェンスな処理を導入しました。その際、行数限界の妥当性検証に皆さんの優れた Muse データが大いに役に立ちました。充実した Muse データが蓄積されているということは、そこから得られる音楽の楽しさだけでなく、開発自体に直接貢献しています。皆さん本当にありがとうございます。

前回のテンポ表示と同様、同じ時刻の属性を 1 つにまとめて表示するようにしましたが、この集約処理は遅延系属性に対してのみ効果的で遅延の無い属性はむしろ可読性を損なう事に気づきました。そこでグループ属性やペダル属性、そしてアクセントは取って替えて同時刻でも複数行に展開するようにしています。

遅延指定がある場合、音部記号エリアの数値は基本的にその最終値を表していますが、強弱属性は各ノートが担う情報であることから例外的な扱いにしています。また強弱は v だけでなくアクセント w も同格に扱う工夫も施しています。一方グループ属性 p, q は表示すべき値が音長となるため、レイアウトの都合上、音部エリアの表示は断念しました。ご容赦下さい。

操作機構上の工夫も施しました。表示対象となるメンバーやフィンガーは、フィンガー情報ダイアログの選択行と連動させるだけでなく、譜面モニタでメンバー ID のキーを入力すると切替えられるようにしてあります。サクサクと切替えられるので重宝すると思います。

さて、2 つ目の譜面モニタに対する強化は、WAVE コマンドも表示するようにした事です。WAVE コマンド自体はあまり使用されませんし、レイテンシー（もたつき）問題も発生するため、効果は今ひとつかもしれませんが、もし使用する際はタイミング調整が格段にし易くなったと思います。そして内部処理の改善は、実はこの WAVE コマンド表示の開発過程から派生しました。この開発の最中、WAVE 発音のレイテンシーが、数バージョン前の Muse と比べて悪化していることに気づいたのです。よくよく調べてみると、なんと前々回の改版で実施した同時刻発音ノートの順序最適化処理に絡んでいました。実験の結果、WAVE コマンドは同時刻内の最後尾に配置するとレイテンシーが緩和されることが判明。多分その次の音源コマンドとのインターバル確保がしやすくなるためだと分析しています。早速、最適な順序処理をするように改善しました。ちなみに今回から添付のオーディオファイル（拍手）を mp3 化してみました。

もう 1 つの内部処理の改善は、曲尾の判定を占有音長だけでなく実際の発音音長も加味し、より長い方を採用するようにしたことです。実はこの改善も積極的なものではなく、後述する文法解釈の変更に伴い必要に迫られて実施しました。以前、掲示板の書き込みで himajin925 さんから、曲の最後に和音を記述した際の曲尾処理に関し疑問を投げかけられたことがあったのですが、今回の改善によって、違和感の無い自然な仕様になったと思います。

さて肝心の文法解釈変更の内容ですが『スタッカートなどで音長が短くなっている音符に連結 $\&$ を添えると、たとえ次に来る音符の音程が同一でない場合であっても、規定音長まで伸ばす』というものです。同一音程で接続される場合がタイとすれば、今回の仕様によってスラーも追加されたと言えます。実は 2 年半前の V5.2 において連結 $\&$ の文法解釈を変更した際に一気にやってしまったかったのですが、プログラムがあまりに複雑に絡み合い、どうしてもまともな動作するコンパイルエンジンに仕立て上げる事ができませんでした。今回譜面モニタ強化に伴い大域的に構造を見直す機会があったので再度チャレンジし、無事完成に漕ぎ着けることが出来ました。無論、データの互換性が少々崩れますが、今回の解釈変更に関してはむしろ意図的にスラーとして記述していた過去データの方が多いだろうと楽観しています。また、それを補って余りある効果があると判断しました。例えば、スタッカートで演奏している一連の旋律の中で、一部だけスラーにしたい場合、今までは音長を明に記述して音を規定音長として指定し、更にその次の音符で再びスタッカート指定をしなければならませんでした。これからはスラーにしたい音符に $\&$ を付けるだけで済みます。あたかもアクセントコマンドのように、一時的指定として機能します。

コーディングの可読性も格段に増すと思われます。

今回の開発後記、少々長くなってしまいました。しかし実際いっぱい開発したのですから仕方が無いですよ（苦笑）。苦勞して開発したこれらの機能、ぜひ有効に活用して頂きたいと切に願っています。

《Ver5.8 開発後記》2010.04.24

その昔譜面モニタが無い頃、昨日まで打ち込み終わった箇所から Musing を再開する際に、それまでの各メンバーの音量値や各フィンガーの強弱、オクターブ値などを確認するために、フィンガー情報を活用していました。譜面モニタが出来てもその Musing スタイルは変わらず、曲尾の値がハッキリしないと不安な私です。そんな私にとっては、フィンガー情報は必須のダイアログなのですが、皆さんからは、今一つ利用頻度が低いという声が挙がっていました。しかし今回のバージョンアップは、そのフィンガー情報の利用頻度を確実に高めるであろう強化です。

従来より、メンバー単位の演奏 ON/OFF はメンバー情報にて可能でしたが、今回、フィンガー情報による特定フィンガーのフォーカス機構を追加しました。すべてのフィンガーを個別に ON/OFF するという機構ではなく、あるフィンガーに焦点を当てる（フォーカスする）というイメージの機構です。フォーカスされたフィンガーが属するメンバー内で、他のフィンガーが全て OFF になります。この機構は同一メンバー内にフィンガーが複数あるドラムパートなどで、特に威力を発揮すると予想しています。今まで、逐一他のフィンガーをコメント化して確認していましたが、その手間が省けます。

このフォーカス機構はメンバー情報の ON/OFF よりも優先して機能し、かつメンバー情報の ON/OFF 機構と併用可能ですので、全てのメンバーを OFF にしておき、上下のカーソルキーでフィンガー情報のリストを高速移動させると、譜面モニタに次々と当該フィンガーのみが表示されていきます。前回サポートした譜面モニタ属性表示機構と組み合わせるととても便利だと思います。余談ですが、各フィンガーでコマを作っておけばバラバラ漫画に使えるかもしれません。無論各コマ単色になりますが、最大 160 コマ作れます（笑）

地味ではありますがもう 1 つ、譜面モニタの自動譜めくり強化を行いました。以前から譜めくりの際にマージンが全くないのが気になっていました。指揮棒カーソルがウィンドウの右端に完全に到達した段階で譜めくりするため、ある程度先を確認しながら聴く場合、譜めくり直前はフラストレーションが溜まります。しかしどの程度のマージンが最適かは、曲のテンポや音長の細かさ、音価の密度などで変化しますので一概に決められません。そうすると、マージン量を ini ファイルファイルで設定してもらおうか、専用の GUI を立ち上げて都度調整してもらおうといった仕様になりますが、私は指定しなければならないパラメータがゴチャゴチャと存在するソフトが大嫌いなのです。パラメータが多いほど、ユーザが好みに合わせて調整できるので優れたソフトであるという考え方もありますが、専用のメニューや機能が無くとも、極自然に手に馴染むような調整が好みです。

そんな訳でなかなか踏み切れなかった譜めくりマージンの改良なのですが、今回譜面モニタの小節線に着眼すれば、私好みの仕様が実現する事に気づきました。Muse 文法には小節の概念はありませんが、譜面モニタはガイド線としての役割で小節

線が存在しています。現在見えている最後の小節線に到達した時、譜めくりするようにすれば、譜面モニタのウィンドウサイズ調整でマージンを設定できます。譜めくりが小節線上で起こるため、必然的に譜めくり後の状態は描画先頭位置に小節線がきっちりと合った状態になります。実に心地良い仕様です。

譜面モニタの拍子はかなり柔軟な値を設定できるようになっているため、拍子の取り方次第では非常に長い小節となり、譜面モニタのウィンドウに1小節が納まらない状況にすることもできます。こうなると譜めくりタイミングの小節線が存在しない局面が出現し上記処理は破綻します。そこで、このような状況下では従来どおりの動作をするという例外処理も組み込みました。ちなみに再び余談ですが、この譜めくりの正規化により、もしかしたらこれまた譜面モニタのパラパラ漫画が作り易くなったかもしれません。先程のフォーカス機構と異なりこちらのコマは、横方向(時間方向)に並べていくので、16色表示が可能です(笑)。という訳で、今回は凶らずもパラパラ漫画に活用できる2つの強化となりました。

そうそう、もう1つありました。譜面モニタの左下に並んでいる[G][T][A]のボタン。その内の[G]と[T]はGive & Takeの頭文字から採ったのですが、思えば実にわかり難い命名です。良い機会なので新しい名称に変更しました。再生開始の意であるPlayと、手動譜めくりの意であるManualです。それぞれの頭文字を採り[P][M]にしてあります。少しは把握し易くなったでしょうか。

前回の改版から半月足らずでメジャーアップをするという、近年稀に見る早さのバージョンアップとなりました。実に気まぐれですが、フリーソフトであるが故の所業とお許し下さい。まあ、直ぐに不具合が見つかり、例によって即マイナーアップをするかもしれないので、今に始まった事ではないんですけど(苦笑)。

《Ver5.9 開発後記》2011.04.23

1年ぶりのバージョンアップとなりました。今回は6つの開発テーマをピックアップしてみました。何年も頭の中で煮詰めていたものから、開発直前に思い付いたもの、ミュージアの皆さんから頂いた要望への対応など、取り上げたテーマは様々です。久々の開発でしたが、着手してみると次々と当初想定していなかった部分にも手を付けたくなり、ボリュームとしても結構盛りだくさんとなりました。

1つ目は、新たなフィンガー属性コマンドの追加です。その名も“移調楽器 y”。大編成のオーケストラやプラスバンドの音楽を Musing する際、異なる移調楽器を1メンバー内の各フィンガーに割り付ける記述法に、従来からもどかしさを感じていました。調性 \ と移調 T がフィンガー属性でないため、次のフィンガーに移る度に指定し直さなければならないからです。そこでこれらのフィンガー属性を新設する検討を以前から行っていました。しかし単に夫々のフィンガー属性を立ち上げる対応は、対処療法的で気が乗りません。納得感が得られないと実装に踏み切れないのが私の性向なのです(苦笑)。そしてやっと先日、お風呂の中で閃きました。課題は移調楽器の指定なのであるから、調性と移調を一括してセットできる機能こそ、目的にストレートに向かって美しいと。こうして完成したのが“移調楽器 y”です。もっと早く欲しかったです。自分自身が(笑)。

2つ目は、小西さんからご要望を頂いた件です。V5.6の改

版でフィンガー拍数のカウント表示を撤廃したのですが、微分音長を活用したデータの場合は譜面モニタでもタイミングのずれを確認できないため、せめて微分音長だけでも定量的な表示を復活させてくれないか、というリクエストでした。小西さんはどの局面でこのような事態に陥るかを具体的な例で示し、しかも極めて謙虚な態度でご依頼してくれました。利用局面の明快なイメージが沸いたため採用させて頂きました。有意義なご提案、ありがとうございます。この要請の真意を汲み取ると、その表示は一覧形式が相応しいと考え、音名指定 x をフッターの方に移して対応しました。ついでにフッターの行数を増やし、今回新開発した“移調楽器 y”の状態も表示させました。フッターの下膨れ感が否めませんが(笑)、このエリアの右クリックによるリロードがしやすくなったことでもあるし、まあ許容範囲でしょう。

3つ目は、多くの方から要望を頂いていたテーマです。ボリューム X7 の遅延指定。X コマンドは実に8年前に実装したのですが、当時から断続的に遅延対応のご要望を頂いており、積年の課題となっておりました。本当は、この機会に X7 に限らず全ての X コマンドの遅延に対応しようと思ったのですが、いくつか複合的な課題があったのです。ご存知のように、Muse の遅延指定は到達値だけを指定するポリシーなため、データの冒頭でいきなり遅延を記述した場合、開始値として音源の初期値を採用する必要があります。ところがいくらサイト等で調べても、代表的なコントロールの初期値しか判らないのです(苦笑)。また仮に判ったとしても、128個の X タイプのすべての値を記憶するのみならず、キープレッシャーに於いては128メンバー×128音程の計16384個の値を管理しなければならない。更にはそもそも遅延という機構には相応しくないスイッチ系のコントロール制御も多く、また V や R など Muse が独自に予約語化したメンバー属性と併用されてしまった場合の開始値の扱いをどうすべきか等々、課題山積でなかなか美しい指針を得られませんでした。結局、実質的に最も使用頻度が高く、実効力のあるコントロールのみをサポートしようと考えました。そして、上記の課題を全てクリアするコントロールは、X7 を置いて他に無いと判断した訳です。この実装は思いの外プログラミング量が多く、Muse.exe の容量が一気に増えてしまいました。と言っても、まだこの Readme.txt より小さいのですが(笑)。何はともあれ、これで V で抑揚を付けながら X7 で全体をクレッシェンドするといった演奏表現が可能となりました。

4つ目は、Ayakong さんの掲示板の書き込みに端を発した改善です。V5.8でのマイナーバージョンアップにて、メンバー情報でキーボード操作をした際、フォーカスをメインウィンドウに強制移動させる処理を施しました。メインウィンドウはすべてのメニュー操作ができるハブ的なウィンドウなので、基本的にフォーカスはそこにある方が視覚にハンディを持たれた方の Musing が楽になるだろうと考えたからです。しかしこれは実に浅はかな推測でした。読み上げソフトはアプリケーションによるフォーカス移動を知らせてくれないため、利用者が迷子になるという事実を諸熊さんから教えてもらいました。「小さな親切大きなお世話」の典型ですね(苦笑)。私はちょっと独善的な面があるので、気を付けなければいけません。今回、強制的にフォーカスを移動する処理を全廃することにしました。Ayakong さん、諸熊さん、ありがとうございます。

5つ目は、アクセント w の文法解釈の微調整です。和音の

カッコの直前に w を添えると和音内のすべての音がアクセント対象となるのですが、連符はカッコ直前に w を添えてもアクセントが効くのは次に出現する 1 音だけ、というのが今までの仕様でした。連符は和音と異なり 1 音づつ奏でられていくので、この仕様の方が自然だと当初は考えていたのです。しかし先日、修飾音を連符で記述しその修飾音全体を一時的に強くしたいという局面に出くわしました。考えてみると、カッコの外側から指定するか、内側の音符に添えるかで効果スコープを制御できる方が明らかに優れた文法です。という訳で連符も、和音と同じ仕様になりました。和音内連符とか連符内和音、連符内コードなどがあるため、プログラム制御が結構複雑で苦労しましたが、綺麗に仕上がったと思います。少々データの互換性が崩れますが、Muse の美しい進化のためにどうかご容赦下さい。

6 つ目は、何年も前から多くの方からご依頼を受けていたテーマです。Muse は、文法エラーを検出すると、各種ウィンドウ状態を初期化します。そして、その文法エラーを直した後リロードしても、直前状態には復帰しません。演奏位置は曲頭に巻き戻されたままですし、演奏をオフにしていたメンバーもオンになったままです。ですから、Musing 中に文法エラーが出ると、エラーを直すよりも直前状態に指定し直す事の方が面倒で、私自身もエラーが出る度に舌打ちしていました(苦笑)。実は Muse は、従来からリロード時に直前状態を極力維持するようにしてありました。無論の事、リロード後のデータが直前状態を維持できない内容に変更されていても、破綻が起きないようにリカバリする処理も組み込んでいます。例えば先程までいたメンバーがいなくなったり、直前状態の演奏位置よりも演奏時間が短くなったりした場合です。これらの対応を万全にしていたので、論理的にエラー時からの復帰もその派生にすぎず、結構簡単に対応できるはずだと頭では考えていました。しかし何と申しましょうか、何やら嫌な予感がして手を付けられずにいたのです。そして、今回改良に着手しその予感が当たっていることに気づきました。結局今回の 6 つの開発テーマの内、このテーマが一番苦労しました。厄介だったのは、エラーが起きてからリロードをするまでの間はデータが何も読み込まれていない状態にしなければならないということです。この状態にしておかないと内部状態と外部状況に乖離が起り、論理的に収拾が付かなくなります。にもかかわらず、リロードでまた直前状態に復帰させなければなりません。この様に、状態クリアと状況維持という背反律を満たさなければなりません。とても神経を使う改良でしたし、動作テストもパリエーションが多くて大変でした。何か気になる挙動があったら、どうぞお知らせ下さい。

他にもいくつかの改善を試みました。初期起動時の音源応答遅延により鍵盤表示と発音がズレる傾向を抑止するために起動直後に MIDI 音源をダミーで Open/Close させたり、フォントのロードによる文字列描画のモタレを軽減するために同時刻イベントでの文字列描画タイミングを後方に移動させたり、バッチ処理モードでは必要ないウィンドウ制御などを排除して高速化を図ったり、譜面モニタがアクティブになっていなくても [SHIFT] キー押下で虫眼鏡カーソルに変化させる、といった諸々の改善を施しました(非アクティブウィンドウのキー押下イベントは検知できないため、マウスを少し動かさないと虫眼鏡になってくれませんが)。という訳で、久々の本格的な開発であったにもかかわらず、どうやらプログラミングの勘は鈍っていない模様で、以上の改修すべてを 2 週間程で終える

ことができました。でも、現段階で偉そうな事を言うてはいけませんね。皆さんからバグ報告がピンバシ飛んで来る様でしたら、やっぱり勘は鈍ってます。ていうか、それは今に始まったことではありませんね(苦笑)。

《Ver6.0 開発後記》2011.07.06

整数部が桁上げされる節目のバージョンアップですので、もう少し派手な更新内容をと当初は考えていたのですが、残された検討テーマや課題は難しいものばかりで、結局ギブアップしてしまいました。ごめんなさい(苦笑)。

1 つは前回リリースした X コマンドの遅延機構を拡大したことです。X 遅延は多くの方が待ち望んでいたようでリリース後反響が大きかったのですが、X7 に限定した仕様に対して再考の要請が起りました。るうさん、諸熊さん、H.N.WPKIDS さん、MIZ さん、PT2K さん、浅川さん達の掲示板によるご意見を参考に、今回の改訂で X0 ~ X127 の範囲で遅延を可能にしました。Muse 予約コマンドと X コマンドとのエイリアス化というアイディアも頂きましたが今回は見送りました。また、そもそも遅延処理が意味の無いコントロールもありますが、本機構を活用するのはスーパーユーザー諸氏であると楽観し、その利用節度を委ねようと思います。

今 1 つは、FING コマンドでメンバー単位にセット対象フィンガーを指定出来るようにしたことです。このアイディアは H.N.WPKIDS さんが掲示板で展開してくれました。それは表面上の仕様設計に留まらない、互換性や整合性にまで言及した完璧な提言でした。ほぼそのまま採用しております。ご提案ありがとうございます。なお本開発を通じて、譜面モニタへの表示属性データで不要なものが残存している箇所を見つけ、対処することが出来ました。重ねて感謝致します。

以上、結果として今回の改訂は機能強化レベルの 2 つとなりました。残された山は登頂困難なものが多いですが、体力、知力、そして気力が続く限り、Muse 開発を継続する所存ですので、今後ともお付き合いの程お願い致します。

Muse取扱説明書 PDF 版

著 者 加藤一郎

編 集 者 PT2K

2010年11月29日 Version 5.8.0

2011年4月27日 Version 5.9.0

2011年11月21日 Version 6.0.0