MOA2.0 モーションアクセラレータについて



図１. モーションアクセラレータ実行画面

ショートムービー　MOA2.0フリーズビッグモード　<https://youtu.be/GFEmF8dnV_M>

概要

MOAはMotionAcceleratorの略です。モーションアクセラレータと読みます。

簡単に言うと、モーションの自動遷移機能のことです。

（　アクセラレータという名前ですが、これはMS Visual Studioのキーイベントをリソース化する機能の名前を真似たものでモーションが速くなるというわけではありません。　）

MOAを使う利点は、大きく３つあります。

1. モーション遷移設定をビジュアルで行える。設定が２段までのツリービューで分かり易い。

②イベント番号（変更のトリガーとなるもの）を指定して関数を呼ぶだけで、モーションの遷移が行える。

　（モーションの途中から、途中への遷移も可能です。）

1. モーションの遷移時に、モーションとモーションの間の補間が自動的に行われる。

(おまけ)モーション変化時にFreezeBigMode(フリーズビッグモード)エフェクトを掛けることが出来る。

イベント番号について

1. でイベント番号という言葉を使いましたが、これについて少し説明します。

ゲーム中でモーションを切り替えるタイミングというのは、プレイヤーがキーボードを押したり自分の打った弾が敵に当たったときなどいろいろあると思います。これらのモーション変更のトリガー（きっかけ）となるものをイベントと呼ぶことにします。イベント番号とは複数のイベントをそれぞれ区別するために使う番号のことです。イベント番号は0から255までの値を使います。イベント番号０はモーションをアイドリングに戻すイベントとして使います。

次ページへ続く

具体的な手順は、

1. あらかじめ走るモーションはイベント１、転ぶモーションはイベント２、ジャンプはイベント４のように、

　ダイアログでそれぞれのモーションにイベント番号を設定します。

1. キーボード(将来的にはゲームパッドも)入力とイベント番号の関連付けをダイアログで行います。
2. キーボードを押してイベントを発行してモーション変化を楽しみます。
3. 設定が済んだらFileメニューからプロジェクトの保存をします。プロジェクト保存時にmoaファイルは保存され、プロジェクト読み込み時にmoaファイルは自動的に読み込まれます。

ダイアログでの設定方法

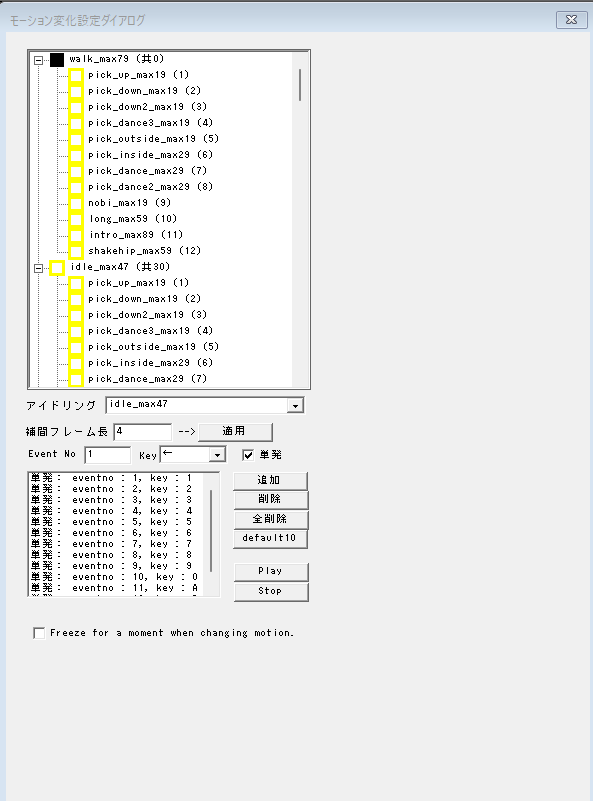


図２.　MOA設定ダイアログ

MOAプレートメニューを押すと右ペインに図２のようなMOA設定ダイアログが表示されます。

ダイアログ上部にツリービューの形式でモーションの変更設定が表示されます。

親子階層は２段までです。

親項目に対して任意で子供項目を設定します。ツリービューは親と子供の２段限定です。子供の子供については、子供を親とする項目を設定します。読み込み済の全てのモーション名がデフォルトで親項目としてあります。初期状態では子供はありません。

親項目は読み込んだモーションを表し、子供項目は親項目から分岐する（イベントトリガーで変化する）モーションを表します。

まずするべきことは、アイドリングモーションを１つだけ選ぶことです。

アイドリングモーションとはイベントが無いとき（イベント番号に０を指定したとき）や

再生中のモーションが終了したときに再生されるモーションのことです。

MOA設定ダイアログのアイドリングコンボボックスでアイドリングモーションにするモーションを１つ選びます。

MOAの設定はツリービューの項目をマウスで右クリックすることで行います。

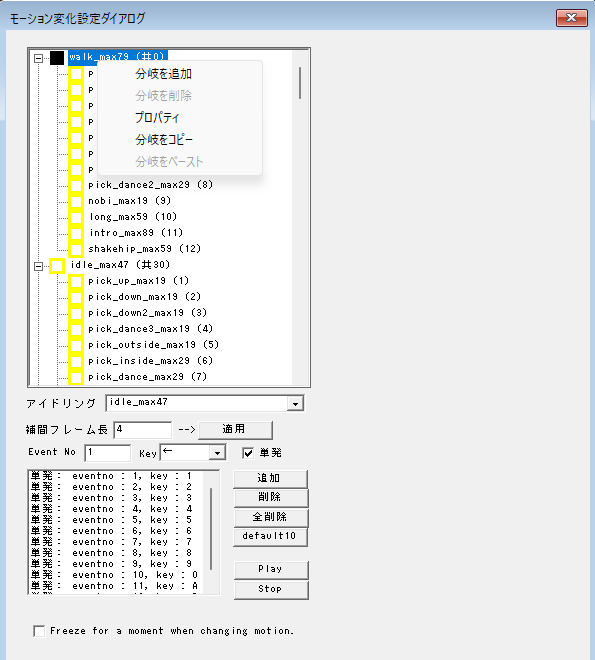


図３.　親項目を右クリックしたときのコンテクストメニュー

図３は親項目を右クリックしたときのメニューです。

「分岐を追加」と「プロパティ」の２種類の操作をすることが出来ます。

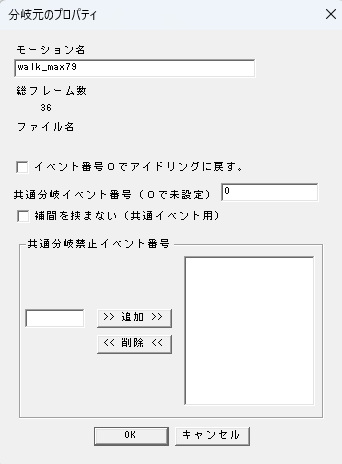


図４. 親項目のプロパティ設定ダイアログ

図４は親項目を右クリックし「プロパティ」を選択したときの図です。

図４の設定項目のほとんどはオプションで必須ではありません。

次ページへ続く

いよいよモーションの分岐の設定をします。

図５は親項目を右クリックし「分岐の追加」を選択したときの画面です。

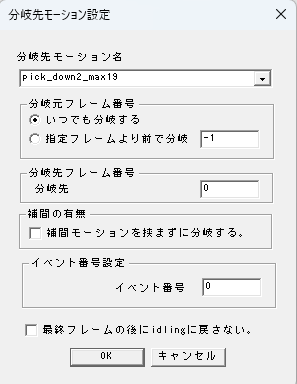


図５. 分岐の追加ダイアログ

まず「分岐先モーション名」のコンボボックスの矢印をクリックし、

親モーションからどのモーションに分岐するかを指定します。

「イベント番号設定」のところに０より大きい整数を入力してどのイベント番号でモーションの分岐が起こるかを指定します。「分岐元フレーム番号」は、「いつでも分岐する」と「指定フレームより前で分岐する」の２種類から選択します。

「いつでも分岐する」を選んだ場合は、

親モーション再生中に「イベント番号設定」で設定したイベント番号が生じたときに

「分岐先モーション名」で指定したモーションへ直ちに（補間モーションを挟みますが）分岐します。

「指定フレームより前で分岐する」を選んだ場合はエディットボックスにフレーム番号を指定します。

「イベント番号設定」で指定したイベントが指定したフレームより小さいフレーム番号で生じた場合にモーションが分岐します。

「分岐先フレーム番号」には「分岐先モーション名」で指定したモーションの

どのフレーム番号にジャンプするかを指定します。

通常はモーションの先頭に分岐するので０を指定します。

（先頭といっても実際には補間モーションが入るので補間モーションの長さ分あとから再生されます。）

親項目を右クリックし「分岐の追加」を選択し、

モーションとイベント番号を指定する作業を繰り返し、

どのようにモーションを分岐させるかを設定します。

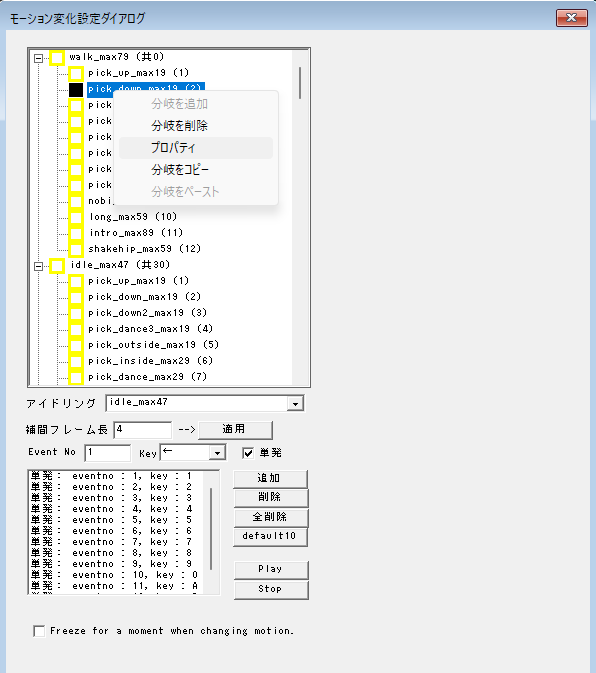


図６. 子供項目右クリック時のコンテクストメニュー

図６.は子供項目を右クリックした場合のメニューの画面です。

「分岐を削除」、「プロパティ」の２種類実行できます。

「分岐を削除」を選ぶと右クリックした子供項目を削除できます。

「プロパティ」を選択すると「分岐の追加」時と同じダイアログが出て、

設定を修正することが出来ます。

子供項目の表示順と優先順位

「分岐を追加」を複数回実行すると、子供項目の表示順番は実行した順番と異なる場合があります。

これは分岐時の優先順位が高いほうから表示する仕組みになっているからです。

分岐時には子供項目のうち上に表示されているものから順番に条件に合うかどうかを判定し、

条件に合うものが見つかった時点で判定を終了しモーションの分岐を行います。

イベント番号の小さいものが大きいものより優先順位が高いです。

イベント番号が同じもの同士の優先順位は、

「分岐元フレーム番号」によります。

分岐元フレーム番号に小さい値が設定されているものほど優先順位が高くなります。

「いつでも分岐」を指定している場合は、一番優先順位が低くなります。

共通分岐イベント番号

親項目のプロパティ（分岐元のプロパティ）ダイアログには、共通分岐イベント番号と共通分岐禁止イベント番号の設定部分があります。

共通分岐イベント番号は、全ての親項目に分岐を設定したいときに使用します。

例えば、「アイドル右」というモーションに共通分岐イベント番号１を設定したとします。

そうすると、全ての親項目に「アイドル右」の分岐を作り、「アイドル右」にイベント番号１を設定したのと同じ効果があります。

つまり、共通分岐イベント番号を使うと、何度も同じ分岐を設定しなくても済むということなのです。

共通分岐イベント番号を設定したモーションは、他の分岐モーションよりも優先順位が高く設定されます。

共通分岐禁止イベント番号は、共通分岐イベントの中で分岐させたくないモーションがあるときに設定します。

例えば、モーション１に共通分岐イベント１を、モーション２に共通分岐イベント２を設定していたとします。

モーション３の親項目では、モーション１の分岐を有効にしたいが、モーション２の分岐は無効にしたいとします。

そういうときに、モーション３の共通分岐禁止イベント番号に２を設定します。

イベント番号０でアイドリングに戻す

親項目のプロパティ（分岐元のプロパティ）ダイアログには、「イベント番号０でアイドリングに戻す」という設定項目があります。

これはループモーションをアイドリングに戻す際などに有効です。

例えば、「歩く」などのループモーションをボタンを押している間だけ再生したい場合、

歩くモーションに「イベント番号０でアイドリングに戻す」を設定しておきます。

キーを離しイベント番号が０になるとアイドリングに戻るようになるので便利です。

補間モーションを挟まずに分岐する

分岐先のプロパティダイアログには、「補間モーションを挟まずに分岐する」という設定項目があります。

イベントが生じたときにただちにポーズを変えたいときなどに使用します。

ファイルの保存

MAの設定が終わったらFileメニューからプロジェクトの保存をします。プロジェクト保存時にmoaファイルは保存され、プロジェクト読み込み時にmoaファイルは自動的に読み込まれます。

MOA2.0での追加オプション

MOAはもともとは１５年以上前にRokDeBone2に付けた機能です。

今回AdditiveIKにMOAを搭載する際に分岐の仕方と補間の仕方を書きなおしました。

更に、Freeze for a moment on changing motionというオプションを追加しました。

このオプションは不具合から生まれたオプションなのですが、モーションの変化にタメを作りメリハリを生みます。

図１のMOA設定ダイアログの下部にこのオプション用のチェックボックスがあります。

３、ファイルフォーマット　（version 3.0）

以下に、moaファイルの一例を、貼り付けます。

//////////////////// 次の行からから、moaのはじまり ////////////////////////////

Motion Accelerator File ver6.0 type01

#TRUNK {

"アイドル正面", "アイドル正面.qua", "Idling0", 0, 0, 0, 0

"アイドル左", "アイドル左.qua", "Normal0", 0, 0, 0, 0

(中略)

"へぼ踊り右", "へぼ踊り右.qua", "Normal0", 0, 0, 0, 0

"長い踊り", "長い踊り.qua", "Normal0", 0, 0, 0, 0

}

#BRANCH "アイドル正面" {

"アイドル右", 1, -1, 0, 0, 0

"アイドル左", 2, -1, 0, 0, 0

"へぼ踊り正面", 4, -1, 0, 0, 0

"小躍り正面", 8, -1, 0, 0, 0

"大踊り正面", 16, -1, 0, 0, 0

"koke2", 32, -1, 0, 0, 0

"zukkoke", 64, -1, 0, 0, 0

}

#BRANCH "アイドル左" {

"アイドル右", 1, -1, 0, 0, 0

"へぼ踊り左", 2, -1, 0, 0, 0

"小躍り左", 4, -1, 0, 0, 0

"大踊り左", 8, -1, 0, 0, 0

}

#BRANCH "アイドル右" {

"アイドル左", 1, -1, 0, 0, 0

"へぼ踊り右", 2, -1, 0, 0, 0

"小躍り右", 4, -1, 0, 0, 0

"大踊り右", 8, -1, 0, 0, 0

}

#BRANCH "小躍り正面" {

"小躍り終", 1, 10, 0, 0, 0

"へぼ踊り左", 2, -1, 0, 0, 0

"小躍り左", 4, -1, 0, 0, 0

"大踊り右", 8, -1, 0, 0, 0

}

（中略）

#BRANCH "長い踊り" {

"へぼ踊り右", 1, -1, 0, 0, 0

"小躍り正面", 2, -1, 0, 0, 0

}

}

///////////////////////////// moaファイルの終わり　///////////////////////////

moaファイルはビジュアルで編集し出力することを想定していますが

テキストフォーマットなので手で編集することも可能です。

moaファイルは大きく分けて３つのパートから成ります。

１、ヘッダー

２、TRUNKチャンク

３、BRANCHチャンク

の３つです。

一行目の

Motion Accelerator File ver6.0 type01

はヘッダーです。

これは一字一句間違わずに半角で記述し最後にエンターキー（\r\n）を記述する必要があります。

TRUNKチャンクは、

#TRUNK {

}

で囲まれた部分です。

「 ｛ 」、「 } 」の直後には、改行（\r\n）が無ければなりません。

TRUNKチャンクはRokDeBone2でのMA設定の親項目を記述する部分です。

内容は

"モーション名１", "モーション名１.qua", "Idling0", ev0idle, commonid, forbidnum, fotbid1, fotbid2, forbid3, notfu

"モーション名２", "モーション名２.qua", "Normal0", ev0idle, commonid, forbidnum, forbid1, notfu

というような形式になります。

モーション名、quaファイル名、モーションの種類　の３つを、半角の「 , 」で区切った形式です。

最後のモーションの種類には、

アイドリングモーションのときは、"Idling0"、それ以外のモーションのときは、"Normal0"と記述します。

アイドリングモーションは、１つだけ指定することを想定しています。

ev0idleの項目には「イベント番号０でアイドリングに戻す」を設定しているときに１、それ以外は０が記述されます。

commonid,共通分岐イベント番号が記述されます

forbidnumには共通分岐禁止イベント番号の個数が記述され、それぞれの共通分岐イベント番号がforbid1, forbid2...のように個数分記述されます。

notfuには補間を挟まずにこのモーションに変更するかどうかを指定します。０のときは補間を挟み１のときは補間を挟みません。

TRUNKのnotfuは共通イベント用です。（BRANCHにもnotfuがあります。）

BRANCHチャンクは、

#BRANCH "小躍り正面" {

}

で囲まれた部分です。

BRANCHチャンクは親項目の数分存在します。

（分岐の無い親項目については記述されません）

まず、#BRANCHの直後に書いてある名前が親項目のモーション名です。

記述してある親項目の分岐情報が、{ } の中に記述されます。

内容は、

"分岐先モーション名", イベント番号, 分岐元フレーム番号, 分岐先フレーム番号, 補間無しフラグ, アイドリングに戻さないフラグ

のようになります。

分岐元フレーム番号は、「いつでも分岐する」を指定した場合は、-1が記述されます。

補間無しの項目には、「補間モーションを挟まないで分岐する」を設定したときに１が、それ以外は０が記述されます。