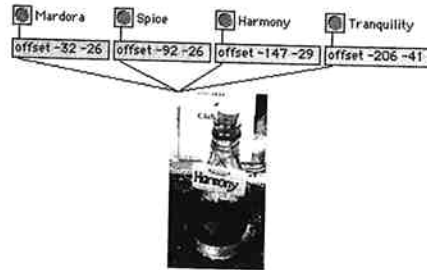


■6-1-11 offsetメッセージによる画像部分の指定



もちろん、これ以外にもmatrixメッセージやrectメッセージを使って、画像の拡大縮小、回転、歪みなどを行うことができる。

しかし、これらの操作の処理負荷は高いので、リアルタイム処理は難しいかもしれない。そのような場合は、必要な画像をあらかじめ作っておくことが考えられる。例えば、画像を10度ずつ回転させたい場合は、10度ずつ回転させた画像をムービーとして作成しておき、回転度数に応じたフレームを表示する方が現実的である。

04/12/3

6-2 ムービーの再生

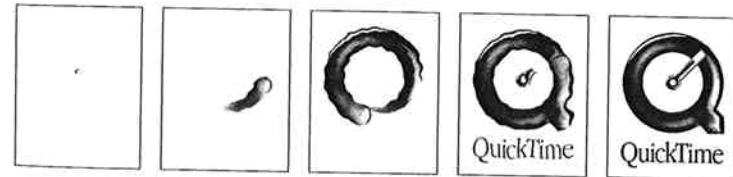
代表的な動画技術の1つにQuickTimeがあり、MaxでもQuickTimeのムービーを再生し、コントロールすることができる。QuickTimeムービーは単にビデオ映像を再生するだけにとどまらず、さまざまな応用が考えられる。ここでは、ムービーを再生するための主要な機能について説明するが、より発展的な活用法を考えてほしい。

● imovieオブジェクトによるムービーの表示



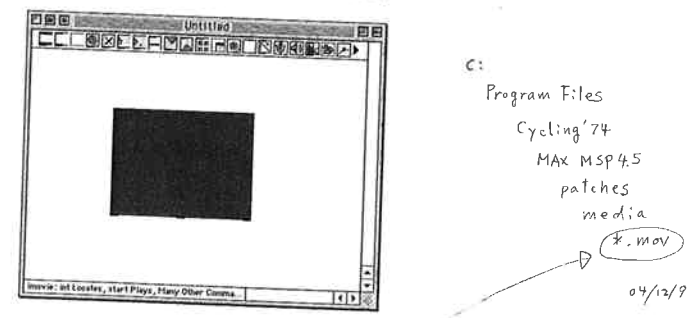
最初に、ここでの説明のために用いるムービーを説明しておこう。これはQuickTimeフォルダにあるSample Movieファイルだ。このムービーは、水滴が回転しながら、その軌跡がQuickTimeマークになるという5秒間のアニメーションで、それに合わせた音楽も鳴るようになっている。画像サイズは、横190ピクセル、縦240ピクセルだ。これはQuickTime 5に付属するサンプル・ムービーだが、もちろんそれ以外のムービーでも構わない。

■6-2-1 Sample Movieの映像



ムービーを再生する手軽な方法は、imovie《QuickTime Movie》オブジェクトとplaybar《QuickTime Play Controller》オブジェクトを用いることだ。imovieオブジェクトはオブジェクト・パレットの26番目にあり、ムービー・ファイルのアイコンで表される。アシスタンス・エリアにはQuickTime Movieと表示される。パッチ・ウィンドウにimovieを作成すると、四角い枠によってムービーの表示領域が示される。

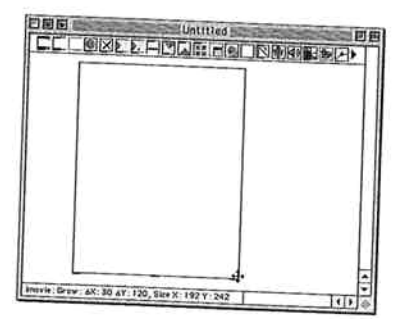
■6-2-2 パッチ・ウィンドウに作成したimovieオブジェクト



次に、imovieオブジェクトを選択した状態で、ObjectメニューのGet Info...を選ぶ。imovieオブジェクトの場合は、インスペクターではなく、ファイル選択ダイアログが開くので、再生したいムービー・ファイルを選べばよい。これでimovieオブジェクトにムービーが表示される。ただし、Sample Movieの最初のフレームは全面白色なので、判別しにくいかもしれない。

imovieオブジェクトの表示領域は自動的に調整されないで、ここでimovieオブジェクトの右下隅をドラッグして大きさを調整しておこう。ドラッグするにつれて、アシスタンス・エリアに大きさが表示されるので、それを参考にすればよい。Sample Movieの場合は、Size X:192 Y:242とする。imovieオブジェクトの大きさは、上下左右の枠線が加わるので、ムービーの大きさに縦横とも2ピクセル加えた大きさとなる。

■6-2-3 imovieオブジェクトの大きさを設定



playbarオブジェクトによるムービーの再生

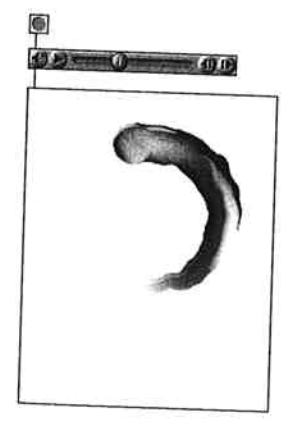
imovieオブジェクトだけではムービーの最初のフレームを表示することしかできない。そこで、playbarオブジェクトを使ってムービー・コントローラーを表示し、再生やコマ送りなどができるようにしよう。playbarはオブジェクト・パレットの38番目にあり、ムービー・コントローラーの再生ボタンのアイコンになっている。アシスタンス・エリアにはQuickTime Play Controllerと表示される。

パッチ・ウィンドウにplaybarオブジェクトを作成すると、ムービー・コントローラーが現れる。これはSimpleTextなど一般的なアプリケーションでムービーを再生するために用いられるユーザー・インターフェース・オブジェクトだ。playbarオブジェクトを機能させるためには、playbarオブジェクトのアウトレットからimovieオブジェクトのインレットへパッチ・コードをつなぎ、playbarオブジェクトにbangメッセージを送る。これでplaybarオブジェクトによってimovieオブジェクトをコントロールが可能になる。loadbangオブジェクトを用いて、パッチを開くときに自動的にplaybarオブジェクトにbangメッセージを送るようにすれば便利だろう。

cf. p.107
cf. p.187

参照 p.83

■6-2-4 playbarオブジェクトによるimovieオブジェクトのコントロール



このようにしてplaybarオブジェクトとimovieオブジェクトを結びつければ、ムービー・コントローラーを使ってムービーの再生やコマ送りなどを行うことができる。playbarオブジェクトの大きさをムービーの横幅に合わせて、ムービーの下に配置し、パッチ・コードを隠せば、より一般的なユーザー・インターフェースになる。

■6-2-5 playbarの大きさと位置を整えたパッチ



ところで、playbarオブジェクトにbangメッセージを送ると、playbarオブジェクトからimovieオブジェクトに特別なメッセージが送られ、それ以降、playbarオブジェクトは何のメッセージも出力しないにもかかわらず、imovieをコントロールすることができる。これは、オブジェクトはすべてメッセージによって動作するというMaxの基本的な動作原理からは外れる例外的な処理となっている。

⑧ 04/1/15

● movieオブジェクトによるムービーの表示

imovieはユーザー・インターフェース・オブジェクトだが、オブジェクト・ボックスによるmovieオブジェクトも利用できる。

movieオブジェクトは、アーギュメントとしてムービー・ファイルの名前を指定すれば、そのムービーを新しいウィンドウに表示する。アーギュメントを指定しない場合は、後述

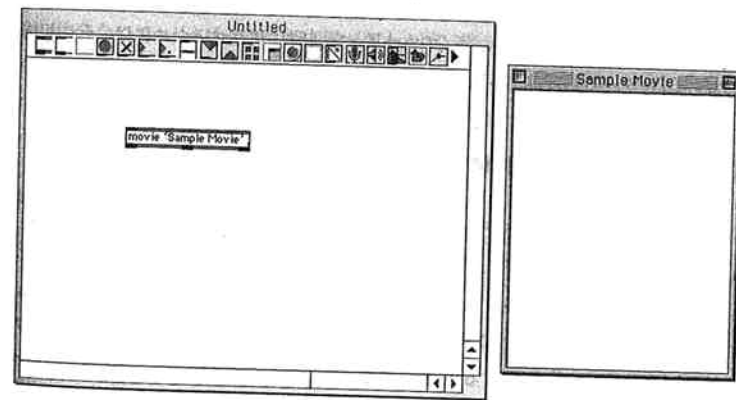
するreadメッセージなどを使って、ムービー・ファイルを表示することになる。

アーギュメントとして指定するファイル名はシンボルでなければならないが、Sample Movieのように途中にスペース(空白文字)が入るファイル名はそのままでは正しく扱えない。このような場合は、Finderでシンボルとして扱えるようなファイル名に変更することが考えられる。例えば、SampleMovieやSample-Movieのようにスペースが入らないファイル名に変更すればよいわけだ。

もう1つの方法として、スペースを含むファイル名を「と」で囲んでアーギュメントに指定することもできる。「と」で囲んだ文字列は、1つのシンボルとして扱われる。「と」は英語フォントでの特殊文字だが、日本語キーボードではoptionキーを押しながら]キーを押すと「を入力することができる。」はoptionキーとshiftキーを押しながら]キーを押せばよい。

いずれにしても、正しくムービーのファイル名を指定すれば、movieオブジェクトを作成すると同時に、そのムービーがムービー・ウィンドウに表示される。ウィンドウの大きさはムービーの大きさに一致している。

■6-2-6 movieオブジェクトとムービー・ウィンドウ(右)



imovieオブジェクトとmovieオブジェクトは、ムービーを表示するウィンドウが異なるだけで、機能としてはほとんど同じものと考えてよい。したがって、次のようにmovieオブジェクトをplaybarオブジェクトでコントロールすることができる。

04/1=3

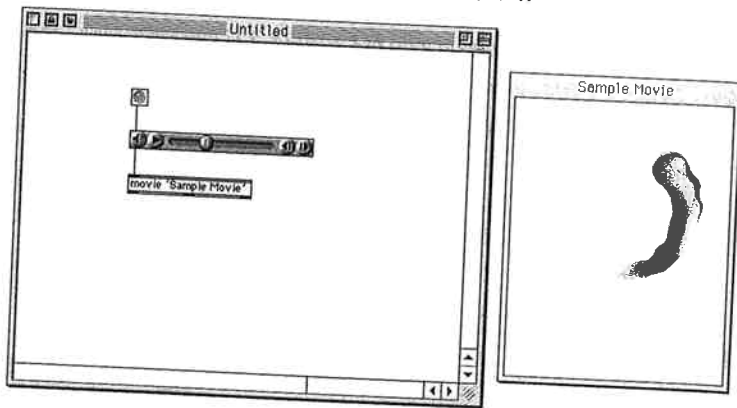
略

ムービーの再生と時間位置

先に述べたように、imovieとmovieはムービーを表示するウィンドウが異なるだけで、ほぼ同じ機能を持ったオブジェクトである。いずれも1つのインレットと3つのアウトレットを持ち、imovieオブジェクトへもmovieオブジェクトへもさまざまなメッセージを送ることができる。無論、出力されるメッセージも同じだ。したがって、ここではmovieオブジェクトを用いてムービーに関するメッセージを説明するが、imovieオブジェクトについても同じだと考えてよい。ただし、ムービー・ウィンドウに関するメッセージはimovieオブジェクトに対しては無効になるなど、多少の違いはある。

ではムービーをコントロールするメッセージのうち、最初に再生関連のメッセージを見ていこう。まず、ムービーを再生するには、startまたはbangメッセージを送ればよい。

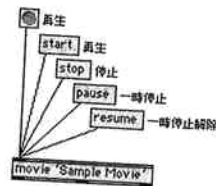
■6-2-7 playbarオブジェクトによるmovieオブジェクトのコントロール



なお、movieでもimovieでも複数のオブジェクトを用いれば、それぞれ複数のムービーを独立して表示し、個別に操作することが可能だ。movieとimovieオブジェクトを混在させても構わない。もっとも、多くのムービーを同時に再生するためには、それ相応のコンピューターの処理能力が必要になる。

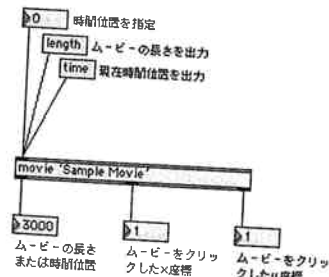
これらのメッセージでは、必ずムービーの先頭から再生されることになる。ムービーの再生を停止するのはstopメッセージだ。これに対して、ムービーの再生を一時停止するにはpauseメッセージを、一時停止を解除して再生を再開するにはresumeメッセージを用いる。

■6-2-8 ムービーの再生



整数のメッセージを送れば、ムービーの任意のフレームを表示することができる。これはムービーの時間位置を指定していることになる。ムービーは時間軸に沿ったデータの集合なので、ムービーの時間位置が変われば、表示されるフレームも変わるわけだ。また、ムービーの時間的な長さは、lengthメッセージを送れば、第1アウトレットから出力される。そして、ムービーの現在の時間位置は、timeメッセージによって第1アウトレットから出力される。

■6-2-9 ムービーの時間位置指定



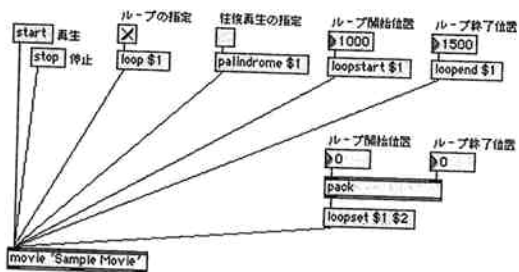
に見えるわけだ。一方、ループを指定した場合には、終端に達すると時間位置を先頭に戻すことになる。このため、ループ再生を行うと何度も再生が繰り返されることになるのである。ループの指定はloopメッセージを用いる。このアーギュメントが0以外の整数であればループが有効になり、0であればループの指定を解除する。

また特殊なループ方法としてpalindrome再生がある。これはムービーの再生が終端に達すると、再生方向が逆になって後戻りするよう再生を始め、さらにムービーの先頭まで戻ると再生方向が元に戻って通常の再生が行われるというものである。つまり、palindrome再生ではムービーの先頭と終端を往復するような再生になるわけだ。このpalindrome再生はpalindromeメッセージによって指定する。アーギュメントが0以外であれば往復再生となり、0であれば通常の一方方向の再生になる。ただし、palindrome再生を有効にできるのは、ループが指定されているときだけに限られる。

ループするムービーの範囲は、何も指定しなければムービーの先頭から終端までとなる。一方、特定の範囲をループ再生するように指定することも可能だ。このためには、ループの開始位置をloopstartメッセージで、ループの終了位置をloopendメッセージで設定する。いずれも、アーギュメントにタイム・ユニットによる時間位置を与える。また、loopsetメッセージは2つのアーギュメントを持ち、ループの開始位置と終了位置を同時に指定することができる。

では、次のパッチで、ループの設定によってムービーがどのように再生されるかを確認しよう。ただし、ループ範囲の設定は慎重に行う必要がある。例えば、ループの開始位置と終了位置が同じであるか、逆転した値であれば、コンピューターがユーザーの反応を受け付けられないフリーズ状態になることもあるので、注意を要する。

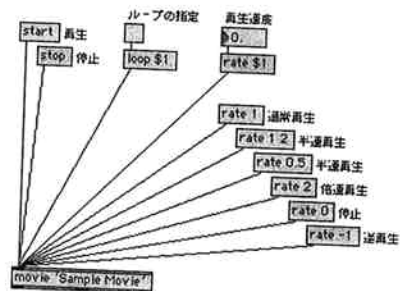
■6-2-12 ムービーのループ設定



ムービーの再生速度はrateメッセージで設定することができる。アーギュメントには整数または実数を与え、通常の再生速度を1として、その倍数で再生速度を指定する。従って、0.5なら半分の速度、2なら倍の速度となる。負の数値なら逆方向の再生となる。アーギュメントに2つの数値を与えた場合は、最初の数値を分子、2番目の数値を分母とする分数として再生速度を指定したことになる。rate 1 2なら半速で、rate 2 1なら倍速ということだ。

6-2-13のパッチで、ムービーを再生しながら再生速度を変えてみよう。速度が遅くなれば、画像や音がゆっくりと変化し、音は音程が低くなるはずである。逆に速度を速くすれば、画像と音が素早く変化し、音程が高くなる。いろいろな再生速度を試すとともに、異なるムービーでも試してみるのがよいだろう。

■6-2-13 ムービーの再生速度の設定



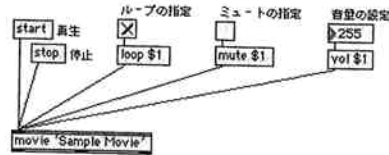
図, 05/3/29

● ムービーのサウンド再生

サウンド・トラック(音声トラック)を持つムービーに対しては、muteメッセージでサウンド再生をミュートすることができる。アーギュメントが0以外ならミュート(サウンドを鳴らさない)、0からミュート解除となる。また、volメッセージでは音量をコントロールすることができる。volメッセージは0から255までの整数のアーギュメントを持ち、数字が大きくなれば音量も上がる。0なら無音、255なら最大の音量となる。

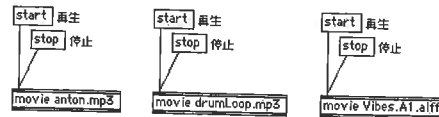
な

■6-2-14 ムービーのサウンド設定



ところで、ムービーと言えば、ビデオ映像やアニメーションを連想しがちだが、サウンドだけのムービーも作成することができる。この場合、ムービー・ウィンドウは表示されないが、ムービーの再生を行えば、そのサウンドが聞こえることになる。また、QuickTimeはAIFFやMP3などのサウンド・ファイルも扱えるので、これらを再生するためにmovieオブジェクトを用いることができる。複数のオブジェクトを用いて、複数のサウンドを同時に鳴らすことも可能だ。

■6-2-15 サウンド・ファイルの再生



movieオブジェクトは単純なサウンドの再生しかできないが、MSPよりも手軽に音を扱うことができる。したがって、簡単な効果音を鳴らすことや、アナウンスを再生するような用途には、MSPではなくmovieオブジェクトを用いてもよいだろう。さらに、QuickTimeはJPEGやPICTなどの静止画ファイルも表示可能だ。そして、MIDIファイルを読み込んで、QuickTimeミュージック・シンセサイザと呼ばれる内蔵ソフトウェア音源を使った演奏もできる。これらの用途にもmovieオブジェクトを使うことができる。ただしmovieオブジェクトはQuickTime VRムービーを完全には扱えないようだ。

05/2/9

QTVRのこと。

● ムービーの表示設定

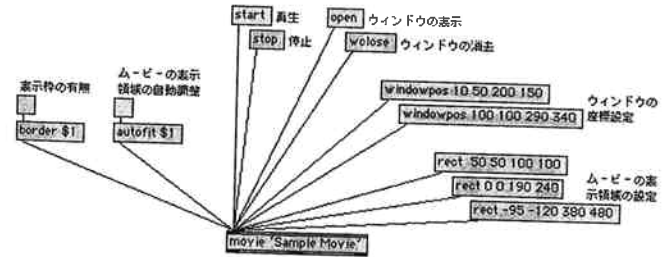
ムービー・ウィンドウを消去する際にはwcloseメッセージを用いる。この場合、ウィンドウが表示されていないだけなので、ムービーを再生すれば画像は表示されないものの、音は聞こえる。ウィンドウを再表示するにはopenメッセージを送ればよい。

borderメッセージは、ムービー・ウィンドウの形状を設定する。アーギュメントが0以外であれば、タイトル・バーを持つ通常のウィンドウになるが、アーギュメントを0とするとタイトル・バーのない枠線だけのウィンドウとなる。

一方、imovieオブジェクトの場合は、アーギュメントが0以外のときに枠線が描かれ、アーギュメントが0であれば枠線が描かれない。

ムービー・ウィンドウの表示位置や大きさを変えるには、windowposメッセージを用いる。アーギュメントには4つの整数を指定し、スクリーンの左上隅を原点としてウィンドウの左端座標、上端座標、右端座標、下端座標を指定する。なお、imovieオブジェクトにはwindowposメッセージは使えない。

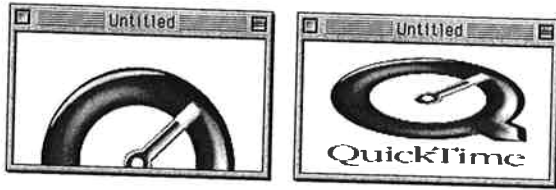
■6-2-16 ムービーの表示設定



ウィンドウの大きさがムービーの大きさと異なる場合は、ムービーの一部が表示されるか、ウィンドウに余白が生じることになる。ウィンドウの大きさに合わせてムービーの表示を調整したい場合は、autofitメッセージを用いるとよい。アーギュメントが0以外なら自動調整が行われ、ウィンドウの大きさに合うようにムービーを拡大縮小して表示され

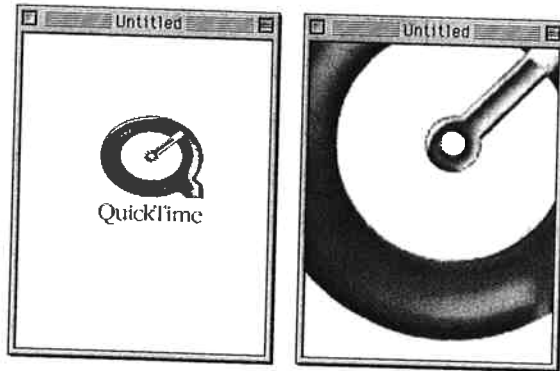
る。imovieオブジェクトの場合は、オブジェクトの表示領域に対して、ムービーの表示が自動調整される。アーギュメントが0なら、通常の表示となる。

■6-2-17 ムービーの表示の自動調整なし(左)とあり(右)



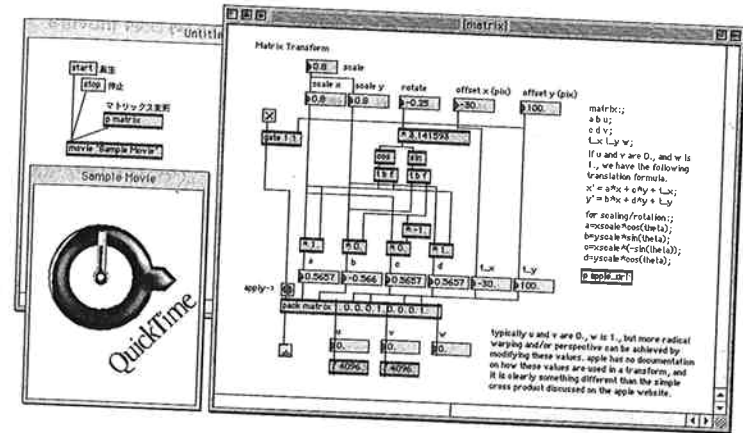
ウィンドウ内のどの部分にムービーを表示するかはrectメッセージで指定する。rectメッセージは4つの数値のアーギュメントを持ち、ウィンドウの左上隅を原点として、ムービーを表示する左端座標、上端座標、横幅、縦幅を指定する。横幅と縦幅の設定によって、ムービーを縮小または拡大表示することになる。rectメッセージはimovieオブジェクトに対して有効だ。

■6-2-18 ムービーの表示領域の設定による縮小表示(左)と拡大表示(右)



ムービーの画像に対しては、マトリクスによって変形操作を行うことができる。これはfpicオブジェクトで説明したマトリクス処理と同じだが、movieオブジェクトの場合はムービーを変形した状態で再生ができる。マトリクス変形の設定は難しいので、movieオブジェクトのヘルプ・パッチにあるmatrixサブ・パッチを利用するのがよいだろう。

■6-2-19 ムービーのマトリクス変形



● ムービー・ファイルの読み込み

movieオブジェクトが再生するムービー・ファイルは、アーギュメントだけでなく、readメッセージによって指定することができる。movieオブジェクトにアーギュメントなしにreadメッセージを送ると、ファイル選択ダイアログが開くので、再生したいムービー・ファイルを選べばよい。readメッセージのアーギュメントにファイル名をシンボルとして与えれば、指定したムービーが開かれる。ムービー・ファイルはMaxのサーチ・パス内に存在しなければならない。

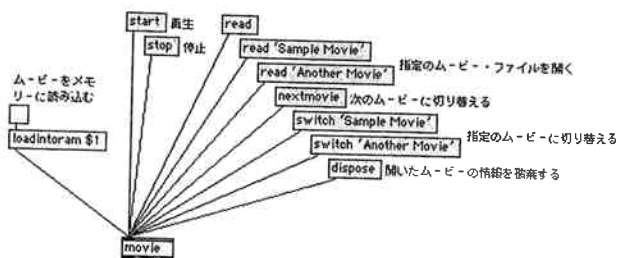
複数のムービーを開いた場合でも、すべてのムービーの情報が保持されている。つまり、readメッセージによって新しいムービーを開いたときに、以前のムービーは消滅す

るわけではなく、表示されなくなるだけだ。そのため、nextmovieメッセージを送ると、それまでに開いたムービーに順次入れ替わっていく。switchメッセージにムービーのファイル名を続けて送ると、そのムービーに切り替わる。同じように、startメッセージのアーギュメントにムービーのファイル名を指定すると、そのムービーの再生を開始する。

それまでに開いたムービーの情報を破棄するにはdisposeメッセージを送ればよい。このとき、同時にムービー・ウィンドウも閉じられる。他のオブジェクトでは、新しいファイルを読み込むと、それまでの内容は自動的に破棄される。これに対して、movieオブジェクトでは明示的にdisposeメッセージを送らなければならない点異なる。他のオブジェクトと同じように、次々と異なるムービーを開いていけば、やがてはメモリーが足らなくなってしまう。新しいムービーを開く度に、あるいは適当な回数ごとに、disposeメッセージを送ることが必要だ。

通常は、ムービーはファイルから一部の情報が読み込まれ、実際の再生はファイル(ハード・ディスク)からデータを読み出しながら行われる。一方、メモリーに余裕があれば、ムービー・ファイル全体をメモリーに読み込んで、メモリーからデータを読み出して再生することができる。ハード・ディスクよりもメモリーの方がデータの読み出し速度が速いので、再生時の負荷が少なくて済む。ムービーをメモリーに読み込むには、あらかじめloadintoramメッセージを0以外の整数のアーギュメントを指定して、movieオブジェクトに送っておく。そしてreadメッセージでムービー・ファイルを開けば、そのムービーはメモリーに読み込まれる。アーギュメントを0としてloadintoramメッセージを送れば、それ以降、ムービーはメモリーに読み込まれない。

■6-2-20 ムービー・ファイルの読み込み



パフォーマンスの改善

ところで、通常の再生は問題なく行えても、特殊な再生はスムーズにできない可能性がある。例えば、特定の時間位置に移動する場合や逆再生を行う場合に、一瞬間が空く場合や、映像や音が途切れ途切れになるコマ落ち状態になるかもしれない。これはコンピューターの処理能力を超える負荷がかかるためだ。

ムービー再生のパフォーマンスを改善するには、当然のことながら、より処理能力の高いコンピューターを用いることだ。すなわち、CPUのクロックが高く、キャッシュ・メモリーが大きく、バス速度が速い方がよい結果が得られる。また、QuickTimeはマルチ・プロセッサーやAltiVec (Velocity Engine)に対応しているため、そのようなCPU構成が望ましいだろう。

すでに説明したが、通常、ムービーはハード・ディスクからデータを読み込みながら再生される。したがって、ハード・ディスクも性能が高い方が、ムービー再生に有利になる。また、ムービーがメモリーに収まるなら、ムービーをメモリーに読み込む方が、再生がスムーズになる。

また、圧縮の設定もムービー再生の重要な要因だ。ムービーのデータはコーデックと呼ばれるメカニズムによって圧縮されており、ムービーの再生時には圧縮されたデータを元に戻しながら再生が行われる。QuickTimeはさまざまなコーデックを提供しているが、多くのコーデックはムービーを通常で連続的に再生するために最適化されている。そのため、特殊な再生にはコーデックが行う処理が増え、コンピューターの負荷が高くなる。

そこで、特殊な再生を行う場合には、その用途に応じて圧縮方法を変えれば、再生時の動作が改善される可能性がある。どのような圧縮設定を行えばよいかは、再生方法や再生環境によって異なるので、一概には言えない。

一般的なアドバイスとしては、コーデックの種類と特徴を理解した上で最適なものを選び、色深度や品質(圧縮率)、フレーム数、画像の大きさなどを適切に設定することだ。場合によっては、1秒あたりのフレーム数を落すことや、画像を小さくすることも考えられる。また、呼び出すフレームが一定間隔なら、それに合わせて基準フレーム単位(キー・フレーム)を設定するとよいだろう。