



## アニメーションを組み合わせて1本のムービーを組み立てる

前号では、ディレクターを使って簡単なアニメーションを作ってみた。今回は、それらのアニメーションを組み合わせて、もう少し体裁が整ったものを作ってみよう。この場合、1つのムービーのなかに複数のアニメーションが入ることになるから、ユーザーの指示に従ってアニメーションを切り替えるリンゴスクリプトが必要だ。

(本誌今月号付録CD-ROMにディレクター体験版、ショックウェーブ、アフターバーナー、サンプルムービーを収録)

藪 暁彦 *yabu akihiko*



# Shockwave for Director

## 第2回 動くホームページの作り方

<http://www.macromedia.com/>

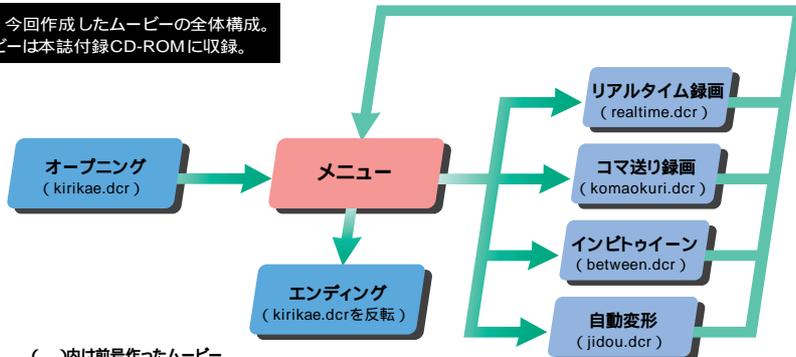
# ディレクタームービーの構成を考える

## ディレクターの機能を紹介する Shockwave タコ劇場

前号では、ディレクターを使ったアニメーションの作り方をいくつか紹介した。今回はそれらを1つにまとめ、ディレクターの機能を紹介するようなムービーとしてまとめてみようと思う。題して「Shockwave タコ劇場」。オープニングには画面切り替えの例として作ったアニメーションを使い、これに少々手を加えたものをエンディングとして使う。

全体の構成は、図1のような感じだ。ムービーを再生するとまずオープニングのアニメーションが流れ、続いてメニュー画面に移る。メニュー画面には、前回作った4種類のアニメーション「リアルタイム録画」、「コマ送り録画」、「インビトゥイーン」と「自動変形機

図1：今回作成したムービーの全体構成。  
ムービーは本誌付録CD-ROMに収録。



( )内は前号作ったムービー

能」をメニューとして表示。各メニューにカーソルを合わせると、メニューの色が変わり、画面右下に簡単な説明を文章で表示するようにする。そしてメニューをクリックすると、画面右上で選んだアニメーションのサンプルが再生されるのだ。

今回、一から作らなければならないのは、

このメニュー部分だ。カーソルを合わせるとメニューの色が変わったり、メニューをクリックすると該当するアニメーションを再生するところには、すべてリングスクリプトが必要だ。アニメーションが終わったらメニューに戻る設定も必要だが、そこにもリングスクリプトがいる。

## 前回、犯した3つの失敗

### 失敗その1

#### 表示モードの設定ミス

32000色やフルカラー表示ができるビデオボードを使っているユーザーも多いだろうが、そうでないユーザーもいる。ディレクタームービーにしる、ショックウェーブムービーにしる、どんな環境で再生されるかわからない。そこで不特定の相手が見るムービーを作るときは、表示カラーを256色(8ビット)にするのが普通だ。

それなのに、前回は32000色モードでムービーを作ってしまったのだ。そのため256色モードで再生すると、キャストに使ったタコがとんでもない色になってしまう(図2参照)。32000色で描いた絵には、256色のシステムパレットでは表現できない色が使われているためだ。

前号でいくつかディレクタームービーを作り、アフターバーナーでそれらをショックウェーブムービーに変換した。どれも特に面白いものではないが、とりあえずなんとかが恰好だけはついたのではないかと。作った当初は、そう思っていた。ところが、あとになってとんだ過ちを犯していることに気がついた。それも3つもだ。筆者はふだんマックを使っている。ディスプレイは、32000色(16ビット)表示モードだ。ディレクタームービーはその環境で作成し、同じ環境でショックウェーブムービーにコンバートして、動作確認をした。それが間違いのもとだった……。

Shockwave for Director(以下、ショックウェーブ)をホームページに埋め込む<EMBED>タグは、アトリビュート(オプションパラメーターのこと)を使えば、ムービー再生時に専用カラーパレットを使う設定に変えることができる。そうすれば256色表示モードでもタコの色が汚くならずすむが、それもなかった……言い訳をするわけではないが、専用パレットはショックウェーブムービーだけでなくディスプレイ全体の表示まで変えてしまう。画面が見づらくなるので、この方法はあまり好きではない。

256色と32000色とでは、画像ファイルのメモリ容量も違う。特にインターネット経由でデータを送るショックウェーブムービーは、ロード時間を考えるとファイルの容量は小さければ小さいほどいい。256色モードでムービーを作れば、どんな環境でも同じ画像

を再現でき、しかもロード時間が短縮される。例えば前回作ったリアルタイム録画のサンプル"realtime.dcr"のサイズは約10Kバイトと小さいが、タコの絵を256色に変換してればもっと小さくできたはずだ。

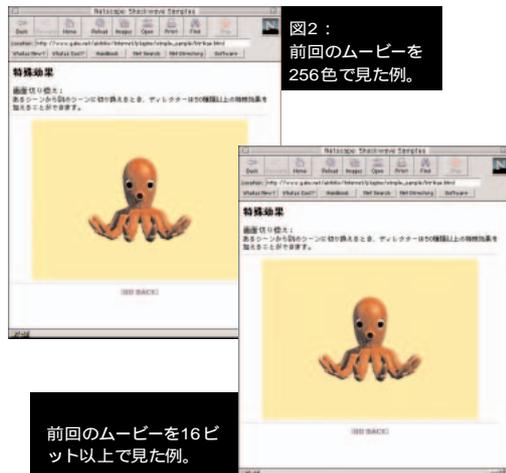


図2：前回のムービーを256色で見た例。

前回のムービーを16ビット以上で見た例。



図3：自動変形で作成するキャストメンバーが多すぎた前回のムービー。



図4：自動変形で作成するキャストメンバーを8点に絞った今回のムービー。

### 失敗その2

#### 3Dグラフィックを使ってしまったミス

そもそもディレクタームービーのキャストに、3Dグラフィックのタコを使ったのが間違いだった。少しでも容量を小さくしたいショックウェーブムービーに、容量が大きくなりがちな3Dグラフィックを使うべきではなかった。立体感を表現するために微妙な中間色を使っている3Dグラフィックを、システムパレットの256色できれいに表現できるはずがない。どうしてもという場合は、256色に減色してから、ディレクターに読み込むべきだろう。

キャストに使うグラフィックは、色数が少ないほど好ましい。今回は、タコの3Dグラフィックを16ビット(32000色)から8ビット(256色)に落としてから、ディレクターに読み込んだ。その際、カラーパレットにはシス

テムパレットを使用し、色ができるだけ元のグラフィックに近づくようにディザ処理を施した。ディザ処理をするとファイルサイズが大きくなってしまふので、できれば使いたくないが仕方ない。

サイズの大きな3Dグラフィックを使っただけで、もう1つ失敗をした。自動変形機能を使った例として、タコがべっちゃんこになるアニメーション(jidou.dcr)を作ったが、ここで不用意にムービーサイズを大きくするようなことをやってしまったのだ。

自動変形機能を使うには、まず変形するキャストをペイントウィンドウに呼び出して形状全体を囲み、効果メニューから「自由回転」、「傾斜」、「パース」、「ねじり」のいずれかを選び、最終的な形に変形させる。続いて効果メニューの「自動変形...」を選択して、新しく作成するキャスト数を指定すると、中間の形状を指定した数だけ作ってくれる。前

回作ったアニメーションでは、動きをなるべくスムーズにしようと、作成するキャスト数を36にした。それが間違いだった。

作成される中間の形状はすべて新しいキャストとして登録されるから、その数だけムービーサイズが増えることになる。そんなわけで、リアルタイム録画やコマ送り録画のショックウェーブムービーが10Kバイト以下なのに対して、自動変形は400Kバイト近くまで膨れ上がってしまったのだ(図3)。

べっちゃんこにするだけなら、何も自動変形機能を使う必要はなかった。ステージ上に配置したスプライトを変形させて、その間をインビトゥーン機能で埋めれば、キャストの数が増えずにすむので、ムービーサイズをかなり小さく抑えられる。だが、自動変形機能を使ったアニメーションはぜひ使いたい。そこで、自動変形ならではのサンプルを新しく作ることにした(図4)。

### 失敗その3

#### ウィンドウズを考慮しなかったミス

最後の失敗は、画面切り替えのサンプルアニメーション"kirikae.dcr"をウィンドウズマシで見ればひと目でわかる(図5)。タコが押す看板には「Shockwave タコ劇場」と書いてあるが、ウィンドウズ版では文字化けを起こしてしまうのだ。ディレクターは、ウィンドウズとマックの両方でムービーを再生したときに、ほぼ同じような文字が表示されるように、ムービー内にフォントマップテーブル(フォントの対応表)が作られる。ショックウェーブも、アルファベットフォントはこのテーブルを持っているのできちんと表示されるが、日本語フォントではダメだ。

これを避けるには、テキストキャストをビットマップキャストに変形すればいい。キャストウィンドウから変形するテキストキャストを選び、「キャスト」メニューから「キャストに変換」を選ぶのだ。いったんビットマップキ

ャストに変換すると元に戻せず、フォントやサイズ、スタイルの変更や文字の編集ができなくなるので注意が必要だ(図6)。

これでマックだけでなく、ウィンドウズでも同じ文字が表示されるようになる。ところがビットマップキャストに変えると、データサイズが大きくなってしまふ。そこで、次に変形したビットマップキャストを1ビットに変換。そうすれば、サイズを元のテキストキャストに近いサイズまで戻すことができる(図7)。

1ビットキャストでも256色モードなら、そのなかの任意の2色を表示できる。キャスト自体はモノクロだが、ステージ上に配置したスプライトにドローツールで色を付けるのだ。文字色はフォアグラウンドカラーで決まり、バックグラウンドカラーで背景色が決まる(図8)。この方法は、テキスト以外のビットマップキャストにももちろん利用できる。サイズを小さく抑えられ、なおかつカラー表示できる1ビットキャストの利用価値は、特にショックウェーブムービーにとって大きい。



図5：マックで見たムービー。ウィンドウズでは文字化けが起きた。

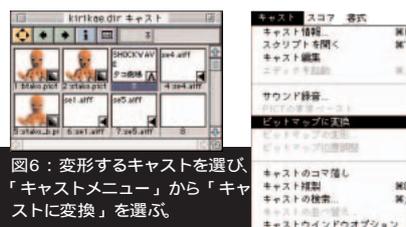


図6：変形するキャストを選び、「キャストメニュー」から「キャストに変換」を選ぶ。



ビットマップ変換後。

# メニュー画面の要は リンゴスクリプト

カーソル位置のSpriteの色を変えて説明文を表示する

「Shockwave タコ劇場」を構成する各部のアニメーションの問題を取り除いたところで、次はそれらを組み合わせて1本のムービーに仕上げる作業だ。ただし、新しく作らなければならない部分もある。前にもふれたメニュー部分だ。

メニューには、アニメーションの例として、「リアルタイム録画」、「コマ送り録画」、「インビトゥイーン」と「自動変換機能」の4つの項目を設ける。このメニューがステージ左側に並び、右上のエリアで選択したアニメーションを再生する(図9)。

メニュー項目にカーソルが移動すると、その部分だけ色が変わり、該当するアニメーションの説明が右下に表示され、同時にアニメーション再生エリアにタコの絵が現れる。マ

ウスボタンをクリックすると、クリック音のような効果音が鳴って、メニュー表示が反転し、マウスボタンを離すと、選択したアニメーションが始まる仕掛けだ。

これら一連の動きは、すべてリンゴスクリプトでコントロールされている。特定のメニュー上にカーソルが移動すると、その部分の色を変えるところで使っているのは、「rollOver」という関数だ。これはカーソルが指定されたSprite上にあるかどうかを判定する関数で、rollOver(n)を指定するとSpriteチャンネルn上のSpriteをチェックする。通常、rollOver関数は条件文(if..... then)と組み合わせて使い、条件が合うとthen以降で指定された処理を実行する。今回のメニューでは、カーソルがメニュー項目に移動すると、その項目名の表示色を変更するフレームへジャンプするように設定した。

前号でも触れたが、リンゴスクリプトには

いくつか種類がある。ムービー全体に設定する「ムービースクリプト」、特定のキャストに対して設定する「キャストスクリプト」、特定のまたは一定範囲のSpriteに対して設定する「Spriteスクリプト」と、特定のフレームに対して設定する「フレームスクリプト」の4種類だ。Spriteスクリプトとフレームスクリプトは、スコア内に表示されることからまとめて「スコアスクリプト」とも呼ばれる。

ムービーの再生が始まると、ビデオレコーダーやテープレコーダーのように、ディレクターの「再生ヘッド」はスコアやスクリプトの指示に従ってフレームを移動する。rollOver関数は、再生ヘッドがあるフレームのカーソル位置を判定する関数だから、メニュー項目の表示色を変更するスクリプトは、フレームスクリプトとして設定した。

図9：今回作成したムービー。これはメニュー選択前の画面。

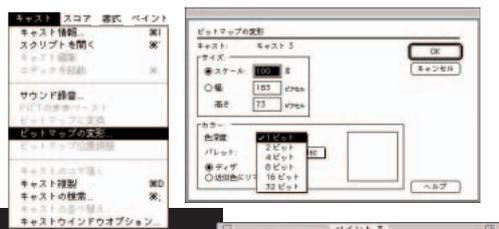
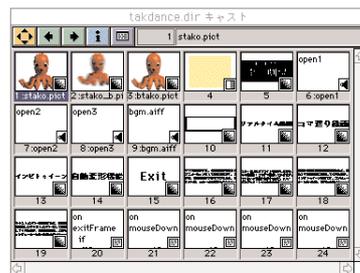
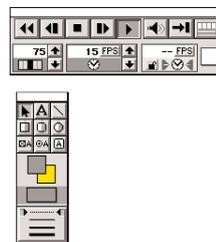
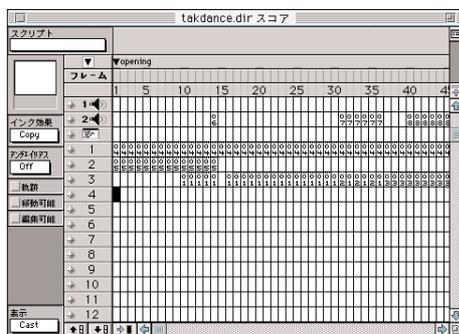


図7：「キャスト」から「ビットマップの変形」を選び、ピットマップキャストを1ピットに変換する。



図8：1ピットのビットマップキャストに2色の色を設定。



## イベント、ハンドラとスクリプトの設定場所

問題はこのフレームスクリプトをどのフレームに設定するのだが、設定しなければならない場所は全部で5か所ある。今回作るメニューには、4つのメニュー項目がある。そこで、各メニュー項目にカーソルがある場合と、メニュー項目以外の場所にカーソルがある場合の5つのフレームに、それぞれスクリプトを設定するのだ。判定条件はどれも同じだから、スクリプトは1種類だけでいい。

フレームスクリプトを設定するには、スコアウィンドウのスクリプトチャンネルでスクリプトが必要なフレームにカーソルを合わせて、マウスボタンをクリック。スコアウィンドウ上部のスクリプト表示ボタンをクリックすればスクリプトウィンドウが開く。このウィンドウ

## マウスクリックで、選択したアニメーションを再生

前に説明したスクリプトは、マウスカーソルの位置に応じてメニュー項目の色を変えるだけだ。肝心なのはその先で、メニューからアニメーションを選択し、マウスボタンをクリックすると選んだアニメーションを再生するスクリプトが必要だ。

メニュー項目のクリックを合図に別のフレームに分岐するわけだから、ここではアニメーション名を囲んでいる枠のスプライトにスクリプトを設定した。このようにフレームスクリプトとスプライトスクリプトは、組み合わせて使われることが多い。

マウスクリックにはボタンを押すイベントと離すイベントの2種類あり、押すほうは"on mouseDown"ハンドラ、離すほうは"on mouseUp"ハンドラにスクリプトを記述する。どちらか一方のイベントに対してスクリプトを設定すればいいが、今回はボタンを押すと効果音が鳴り、離すと選択したアニメーションに分岐するスクリプトを設定してみた。

に、次ページ表1のようなスクリプトを入力する。

スクリプトは、マウスボタンのクリック、マウス移動、キー入力、ムービー開始/停止といった「イベント」が起こったときに実行される。「フレームに入る」、「フレームから出る」のもイベントの一種だ。メニュー項目の表示色を変えるスクリプトは、特定のフレームに対して設定しているが、ここでは指定したフレームを出るときにスクリプトを実行するようにした。1行目の"on exitFrame"が、それを表している。

この"on exitFrame"を「ハンドラ」といい、フレームから出るときにこの行と最終行の"end"との間に記述されたスクリプトを実行する。ディレクターには、このほかマウスボタンを離れたときにスクリプトを実行させる"on mouseUp"などのイベントハンドラが用意さ

スコアウィンドウの左上に、「スクリプト」と書かれた部分がある。この下がプルダウンメニューになっているので、そこから「新規...」を選択。スクリプトウィンドウが開くので、次ページ表2のスクリプトを書き込む。なお、例にあげたのはリアルタイム録画のサンプルアニメーションを再生するためのスクリプトだ。

1~3行目が、マウスボタンを押したときに効果音"se2.aiff"を再生するスクリプトだ。通常、サウンドはスコアウィンドウのサウンドチャンネルで設定するが、ここではそれをせずにスクリプトで制御している。"puppetSound"は、スクリプトでサウンドを鳴らすときに使うコマンドだ。イベントに同期してサウンドを再生したいときなど、スコアのサウンドチャンネルではなく、puppetSoundが使われる。ただし、puppetSoundはサウンドチャンネル1より優先されるので、その間サウンドチャンネル1に設定したサウンドキャストは再生できない。

続いてマウスボタンを離すと、リアルタイム録画したアニメーションの再生が始まる。4~6行目がその部分のスクリプトで、"takoa"

れているが、自分で新しいハンドラを設定することも可能だ。

スクリプト2~3行目は、カーソルがスプライトチャンネル1上のスプライトにあるとき、"realtime"と名付けられたフレームにジャンプすることを表している。フレームには、「マーカ」と呼ばれる目印とマーカ名を付けることができる。"realtime"は、「リアルタイム録画」のメニュー項目の色を変えるフレームだ。ジャンプ命令"go to"は、フレーム番号で分岐先を指定できるが、このようにフレーム名でジャンプ先を指定することもできる。

同様に、4~5行目は「コマ送り録画」、6~7行目は「インビトゥイーン」、8~9行目は「自動変形機能」のメニュー項目の色を変えるフレームにジャンプする命令で、どのメニュー項目にもカーソルがないときは、10~11行目で指定した"select"へ分岐する。

はリアルタイム録画したアニメーションが始まるフレームに付けたマーカの名称だ。これと同様のスクリプトを、メニュー項目のアニメーション名を囲んだすべてのスプライトに設定すれば、条件分岐の設定が完了する。

アニメーションを見終わったら、メニュー選択に戻るための設定も必要だ。そこで設定したのが、アニメーションの下に表示される"STOP"の文字だ。ここにスクリプトが設定されていて、クリックするとスクリプト(表2下)を実行し、再びメニュー選択ができるようになる。このスクリプトは、アニメーションを設定したフレーム内のすべてのスプライト("STOP"の文字がある部分)に設定してもいいが、楕円形キャストに設定したほうが作業も簡単だし手間もかからない。

ほんの数行のスクリプトを書くだけで、こうしたインタラクティブな操作のできるムービーが作れるのだ。話を聞き、文章を読むだけだと難解そうに感じるスクリプトだが、実際に書いてみると意外に単純で、そのわりには大きな効果が得られる。



表1

項目名の表示色を変えるフレームスクリプト

```

on exitFrame (1)
  if rollOver(1) then (2)
    go to "realtime" (3)
  else if rollOver(3) then (4)
    go to "komaakuri" (5)
  else if rollOver(5) then (6)
    go to "between" (7)
  else if rollOver(7) then (8)
    go to "auto" (9)
  else (10)
    go to "select" (11)
  end if (12)
end (13)
  
```

- (1) 再生ヘッドがフレームを離れるときに、以下のスクリプトを実行する。
- (2)、(3) カーソルがスクリプトチャンネル1上のスクリプトにあれば、"realtime"と名付けられたマーカーがあるフレームにジャンプする。
- (4)、(5) カーソルがスクリプトチャンネル3上のスクリプトにあれば、"komaakuri"と名付けられたマーカーがあるフレームにジャンプする。
- (6)、(7) カーソルがスクリプトチャンネル5上のスクリプトにあれば、"between"と名付けられたマーカーがあるフレームにジャンプする。
- (8)、(9) カーソルがスクリプトチャンネル7上のスクリプトにあれば、"auto"と名付けられたマーカーがあるフレームにジャンプする。
- (10)、(11) 上のどのチャンネルにもカーソルがないときは、"select"と名付けられたマーカーがあるフレームにジャンプする。
- (12) 条件文の終了。
- (13) ハンドラの終了。

表2

リアルタイム録画のアニメーションにジャンプするスクリプト

```

on mouseDown (1)
  puppetSound "se2.aiff" (2)
end (3)
on mouseUp
  go to "takoa" (5)
end (6)
  
```

- (1) マウスボタンが押されると、次の行以降に記述されたスクリプトを実行。
- (2) サウンドキャスト"se2.aiff"を再生。
- (3) ハンドラ終了
- (4) マウスボタンが離されると、次の行以降に記述されたスクリプトを実行。
- (5) マーカー名"takoa"に分岐。
- (6) ハンドラ終了。

メニュー選択に戻るためのスクリプト

```

on mouseUp (1)
  puppetSound "se2.aiff" (2)
  go to "select" (3)
end (4)
  
```

- (1) マウスボタンを反すと、次の行以降に記述されたスクリプトを実行。
- (2) サウンドキャスト"se2.aiff"を、puppetSoundで再生。
- (3) メニュー選択画面に戻る。
- (4) ハンドラ終了。

各部を1つにまとめてムービーを完成させる

これまで作ったアニメーションは、それぞれ別のファイルになっている。最後は、これらのファイルを1つにまとめる作業だ。通常のアプケーション同様、新しいファイルにコピー&ペーストで各部を貼り込めたいのだが、ワープロのようなわけにはいかない。ディレクターは一度に1つのファイルしか開けないからだ。そこで各部のアニメーションファイルを開いてデータをコピーし、次に新しいファイルを開いてコピーしたデータをペースト。これを何度か繰り返すのだ。

コピーするのは、スコアウィンドウだけでいい。スコアウィンドウのデータをコピーすると、キャストウィンドウのデータもコピーできる。ただし、マーカーとマーカー名はコピーできないので、新しいファイルでもう一度設定し直さなければならない。

こうして出来上がったディレクタームービーのサイズは216Kバイトあり、アフターバーナーで変換したショックウェーブムービーは

86Kバイトまで圧縮できた。それほど巨大ではないが、気楽にロードしてみようという気になるほどコンパクトでもない。前回とは違い、マックで見てもウィンドウで見ても、ほぼ同じように表示されるムービーを作ることができた。だが、ファイルサイズが問題だ。一部のショックウェーブマニアの間では、小さなサイズのムービーを作るのが流行っている。それほど、サイズを小さくするのは難しい。

構成が複雑になればなるほど、サイズは大きくなる。それよりも1つ1つの構成はできるだけ単純にして、別のシーンに移るときは別のファイルを呼び出すような設定をしたほうがいいのだろう。そのためにショックウェーブのリンゴには、"gotoNetMovie"などネットワーク拡張コマンドが用意されているのだ。

他人が作ったショックウェーブムービーを見るのも楽しいが、自分で作るともっと楽しい。機会があったら、ぜひ作ってみてほしいと思う。

Shockwave for Freehandが付録CD-ROMに収録されています。ディレクトリは、  
Mac: Mac / Plugin / Shockwave for FreeHand  
Win: WIN\NSPLUGIN\SWFH

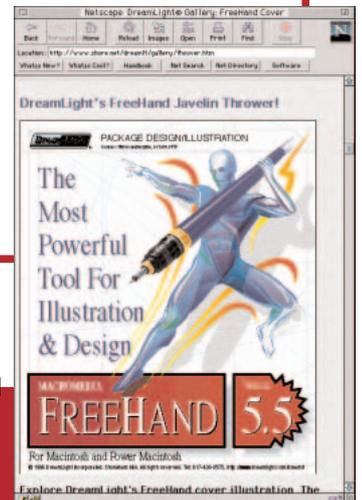
今回作成したムービーサンプルは本誌CD-ROMに収録されています。ディレクトリは、MAGNAVI / IP9607 / SWSAMPLE / movie。



ショックウェーブのムービーサンプル。

ショックウェーブ用ツールが続々と登場!

4月、マクロメディアからペイントツール「フリーハンド」対応の Shockwave for Freehand、Shockwave for Authorwareもリリースされ、ショックウェーブはますます広がりを見せつつある。





## [インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

**株式会社インプレスR&D**

All-in-One INTERNET magazine 編集部

[im-info@impress.co.jp](mailto:im-info@impress.co.jp)