

# Flash の 教 育 内 容

小 松 香 爾

## はじめに

「Internet Explorer」などの Web ブラウザ自体が、優れたユーザインターフェースを持つクライアントではない。この事実は、Web 上のアプリケーション開発者の間では共通認識になっている。より正確にいうと、HTML のみで、優れたユーザインターフェースを構築することはできないということである。具体的には、「サーバにアクセスしないと画面遷移ができない」、「複雑な画面を作ると動作が重くなる」、「ファンクションキーに自由に機能を割り当てられない」などの不具合があげられる。このような Web ブラウザ（および JavaScript）の限界を受けて、近年、視覚的なアドバンテージを持ち、かつ使いやすいユーザインターフェースへの要求が高まってきた。その流れの中で、Flash はリッチインターネットアプリケーション開発ツールへと変貌を遂げてきた。当初の Flash は、アニメーション制作専用のソフトウェアであり、インタラクティブなアプリケーション制作はターゲットに入っていなかった。しかし、Flash はバージョンを重ねるごとに多様化、高度化していき、バージョン MX（実質的なバージョン 6）は、RIA（リッチインターネットアプリケーション）開発ツールとして売り出されることになった。「Flash5」で「ActionScript」が導入されてから、Web アプリケーション（「Macromedia」社は、リッチインターネットアプリケーションと呼んだ）の制作に使用されるようになったからである。実際、2001年に業務を開始したソニー銀行は、Web の預金管理システムのユーザインターフェースとして Flash を採用した。「ActionScript」は、Flash のスクリプト言語であり、「JavaScript」と同じ「ECMAScript」に基づいて設計されている。従来、Web サイトのユーザインターフェースは HTML の form 要素と、「JavaScript」を組み合わせて構築されてきた。それが、Flash のみで作成されるようになったのである。現在の Flash のユーザ層は、アニメーションを制作する層とリッチインターネットアプリケーションを構築する層に分類される。また Flash で、制作されているコンテンツは、「Web アニメーション」、「アート」、「プレゼンテーション」、「ゲーム」、「教材」、「ユーザインターフェース」、「Web アプリケーション」など多岐にわたる。しかし、守備範囲の広さだけが Flash の重要性を証明するものではない。Flash の重要性は、Flash ムービーの再生環境、すなわち「Flash-Player」の普及率の高さにもある。「Adobe」社のサイトによると、日本にあるパソコンの<sup>(1)</sup>98.3%に、「FlashPlayer6」以降がインストールされている。この98.3%という数字は、Win-

dowsの標準ブラウザである「Internet Explorer」の普及率を上回ることはもちろん、「JavaScript」を有効にしているブラウザの率をはるかに上回っている。現在では、「Flash Player」をインストールしないことには、「Web上のコンテンツを正常にブラウズすることができない」という状態にあるといっても過言ではない。現在、大手企業のサイトの大半には、Flashで制作された動的なコンテンツが組み込まれている。企業向けのFlashを専門に制作するWebデザイン会社も存在する。日本では「defide」<sup>(2)</sup>が有名である。

Webから離れたFlashの動向としては、TV、映画放映があげられる。米国では、2001年頃から、Flash制作したアニメーションがTV放映されていた。既にFlashによるアニメーションがビジネスとして成立していたのである。日本では、FlashアニメーションのTV放映は米国に遥かに遅れた。初めてTV放映された作品は、2006年4月からテレビ朝日の深夜枠で流された「The Frogman Show」<sup>(3)</sup>である。ただし、Web上での有料視聴番組は、既に2001年から始まっていた。有名なものとしては、「株式会社コンテンツジャパン」<sup>(4)</sup>の「ブラックジャック」<sup>(5)</sup>（2001年8月放映開始）があげられる。「ブラックジャック」は、手塚治虫の人気原作漫画をWeb配信向けにFlashで描き直し、声優に「宇多田ヒカル」などを起用した。視聴料は全12話で2850円であった。「ブラックジャック」は、半年間で8万人以上の有料視聴者を獲得した。これは、Web上の映像の有料配信としては、成功した部類である。成功の原因は、その映像の美しさにあった。特に、大画面で見ても映像が美しかった。動画のストリーミング配信では、「Windows Media Player」<sup>(6)</sup>、「Real Player」<sup>(7)</sup>、「QuickTime Player」<sup>(8)</sup>などが代表的であるが、これらは全てビットマップ画像（連続的な表示で映像になる）を配信するためのアプリケーションである。これらの動画Playerでは、原理的に回避不可能な欠点がある。画面を拡大表示すると、画像が荒れるというビットマップ画像特有の問題である。「Flash Player」はベクトル画像を配信する。したがって、これらの動画Playerが持つ欠点はない。なお、「Flash Player」はパソコンだけでなく、PDAやゲーム機などにもインストールできる。携帯電話には、あらかじめ「Flash Lite」<sup>(9)</sup>という「Flash Player」が搭載されている。

上述のFlashの多目的性、普及率を考慮すると、Flashが、Webデザイナーを目指す学生が学ぶべきソフトウェアであることは、疑問の余地はない。しかし、Flashを教育機関で教える際には、注意が必要である。その多目的性から、4年かけても全てを教えることは困難であり、「何を教えて何を教えないか」という取捨選択を強いられるからである。カリキュラムは、大きくわけて「概要」、「グラフィックス」、「アニメーション」、「ActionScript」となるであろう。Flashをアニメーション制作ツールとして教える場合は、「グラフィックス」、「アニメーション」は必須であり、ユーザインターフェース制作ツールとして教える場合は、加えて「ActionScript」も必要となる。ただし、Webアプリケーション制作に関しては、現在はFlashから派生した「Flex」<sup>(10)</sup>というソフトが発売されている。「Flex」では「MXML」というXMLベースのマークアップ言語を使って、Flashのコンテンツを作成する。従来のプログラマーには、「Flex」の方が馴染みやすいことは間違いなく、本格的なアプリケーション制作

の為に、Flash を使用する機会は、今後減っていくことも予想される。

本論文では、第1節でFlash がメジャーソフトになった経緯、第2節で全体的な特徴を述べる。第3節では、Flash のグラフィックスの能力、第4節では、アニメーションの能力、第5節で、ActionScript の特徴を述べる。

## 1 Flash の歴史

Flash は当初、アニメーション制作ソフトであった。ところが、アニメーションを制御するスクリプト言語「アクション」が、オブジェクト指向言語「ActionScript」となり、リッチインターネットアプリケーション（RIA）開発ツールとして使用されていくことになる。

教育内容として、Flash の歴史自体を細かく教える必要はない。しかし、Web デザインの世界における基本知識として、以下の3項目は重要である。

1. もともとは、「Illustrator」と同じドローソフトであった
2. 「Director」からインタラクティブコンテンツ制作ソフトの主役の座を奪った
3. 「ActionScript」というスクリプト言語を使用できるようになり、従来のプログラマーが、Web アプリケーション開発ツールとしてFlashを使用するようになった

上記を踏まえた上で、下記の2-1節～2-4節の内容を肉付けして教育すればよい。

### 1-1 Director から Flash へ

インターネットが商用使用されるようになる前は、「Macromedia」社の「Director」がインタラクティブコンテンツ作成ソフトとして使用されていた。当時は、「マルチメディア＝CD-ROM」という時代であり、「Director」で作られたコンテンツも CD-ROM に収められていた。正確に述べると、当時は「コンテンツ」という単語自体がなく、「Director」で制作したコンテンツは、「マルチメディアソフト」と呼ばれて発売されていた。その後、インターネットが一般に開放され、「Director」で作った作品をネットで流すための「Shockwave」というプラグインがリリースされた。しかし、「Director」は、ビットマップ画像を扱うソフトウェアであり、生成されたファイルのサイズが大きかった。当時のインターネットの帯域幅では、ファイルサイズの大きな作品を流すのは無理があった。高速インターネットの時代に入っても、Web 上のインタラクティブコンテンツ制作の需要が高まっていた。インターネットという新しいメディアの成長とともに、「Director」にとって代わったのが、同じ「Macromedia」社のFlashである。ただし、Flashはもともと「Macromedia」社が開発したソフトではない。「Future Wave」社の「Smart Sketch」がFlashの原型である。「Smart Sketch」は、アニメーション制作ソフトではなく、ベクトルデータを扱うドローソフトでありながら、ペイントソフトに似た操作感を持つグラフィックスソフトであった。その後、「Future Wave」社は、「Smart Sketch」にアニメーション作成機能を加えて、「Future Splash Animator」というソフトを開発した。「Macromedia」社は、1997年1月に「Future Wave」社を買収し、

「Future Splash Animator」の中身を変えずに、1997年2月に「Flash」（Flash1）という名前で発売した。「ベクトル画像を用いたアニメーション」という、Flashの最大の特徴は、「Future Splash Animator」で既に実現されていた。しかし、「Future Splash Animator」では、サウンドが全く扱えず、再生速度が遅いという、アニメーション作成ツールとしては致命的な欠点があった。また、インタラクティブな機能はボタンしかなかった。そのため、使用用途は、ホビーユースに限られていた。Flashは、その後、「Macromedia」社によって、改良を加えられていくことになった。なお、「Flash2」からは日本語に対応した、日本語版のソフトが開発されるようになり、日本語版は「Flash2J」などのように「J」をつけられた。

### 1-2 アニメーション制作ツールとしての発展期

Flashがアニメーション制作ツールとして不動の座を築いたのは、「Flash2J」（1997年2月発売）と「Flash3J」（1998年5月発売）においてである。「Flash2J」では、サウンドとビットマップ画像を扱えるようになった。「Flash3J」では、「タイムライン」を独自に持つ「ムービークリップ」と、透明度を制御する「アルファ」、レイヤーの一部を隠す「マスク」が扱えるようになった。これらの機能の導入により、Flashの表現能力は格段に向上した。その結果、多数のWebデザイナーが、Flashユーザとなり、WebページでFlashの動画が頻繁に見受けられるようになった。1998年にはSWF形式の仕様が公開された。以後、サードパーティからFlash以外のSWF形式を生成するアニメーション制作ソフトが発売されていくことになった。

CGIとの連携が可能になり、Webコンテンツ制作ソフトとしての色合いが強くなったのは、1999年7月に発売された「Flash4J」においてである。具体的には、「アクション」の機能が強化され、外部のプログラムとの間で変数の受け渡しが可能になった。ただし、「アクション」は、スラッシュを使用する独自の構文を使用する上に、言語としての能力は貧弱であった。「アクション」には、少数の組み込み関数と、プロパティしかなく、プログラミング言語と呼べるものではなかった。なお、「Flash4J」では、携帯音楽Playerで主流のMP3形式のサウンドファイルを読み込めるようになった。

### 1-3 RIA 開発ツールとしての発展期

本格的なプログラミングが可能になったのは、2000年10月に発売された「Flash5J」以降である。「Flash5J」では、アクション機能が「ActionScript」としてまとめられた。「ActionScript」は「JavaScript」と同様に、ECMA-262改訂3版にほぼ準拠したスクリプト言語である。「ActionScript」は、オブジェクト、プロパティ、メソッドの概念を持つオブジェクト指向言語でもあり、従来のプログラマーもFlashを使用するようになった。「Flash5」は、Flashがアニメーション制作ソフトから、Webアプリケーション制作ソフトになった最初のバージョンであるといえる。また、「Flash5J」では「XMLSocket」により、XMLデータの通信が可能になった。しかし、通信可能であるのは、XMLとテキストデータのみであり、

画像、サウンド、動画などは取り扱えなかった。グラフィックスの機能としては、「ペンツール」が加わり、「Illustrator」で使われるベジェ曲線が描けるようになった。「Flash5J」のアニメーション制作ツールとしての完成度は高く、アニメーションを作るだけなら、現在でも「Flash5J」で十分であるといわれている。日本で、Flashの人气が爆発したのが、この「Flash5J」が発売後であったこともあり、Web上でアニメーションを公開する一般ユーザの間では、いまだに「Flash5J」が使われている場合も多い。

単なるアニメーション制作ツールとしての路線を捨てたのが、2002年3月に発売された「FlashMX」であった。「FlashMX」は「Macromedia」社の「Miracle eXperience」ソフトウェア群の一部としてリリースされた。そのため、リッチインターネットクライアント(RIA)開発ツール「StudioMX」の中にも組み込まれ、サーバサイドプログラミングが可能な「ColdFusionMX」との連携も重視された。これは、Flashで「データ駆動型アプリケーション」の開発が可能になることを意味していた。また「Macromedia」社純正のサーバである「Flash Communication Server」(現在のFlash Media Server)にも対応した。このFCSにより、ライブチャットシステム、オンラインTV会議システム、オンライン対戦ゲーム、Web上でのマルチメディアによるリアルタイムコミュニケーションを行うアプリケーションの開発が可能になった。また、「UIコンポーネント」というユーザインターフェース制作のためのテンプレート集も用意された。

「FlashMX」は、インターネットアプリケーションを制作するプロには受け入れられた。しかし、アニメーションを作るだけの一般ユーザのFlash離れを引き起こした。(定価58000円と高価格であったことも一因にある) サードパーティから、SWF形式のファイルを作成するソフトが発売されるようになったのもこのころからであり、アニメーション制作を目的とする一般ユーザ向けに、「Flash Maker」<sup>(11)</sup>、「SWiSH」<sup>(12)</sup>などの廉価版ソフトウェアも発売された。なお、現在では「ParaFla」<sup>(13)</sup>というフリーソフトもある。

#### 1-4 RIA 開発ツールとしての熟成期

「FlashMX2004」では、オブジェクト指向言語の標準を意識したスクリプト言語「ActionScript2.0」が搭載された。「FlashMX」の「ActionScript1.0」では、ビルトインクラスの拡張、カスタムクラスの定義といったクラスの扱いで、「Java」や「C++」など一般的なオブジェクト指向言語とは異なる方法を強要された。また、データ型のチェックがなく、コンパイル時にエラーが発見できなかった。「FlashMX2004」では、ECMA-262改訂4版に準拠した「ActionScript2.0」を採用したことにより、これらの問題点を解決した。また、通常版と同時に、「Flash MX Professional 2004」という、開発者向けのバージョンもリリースされた。プロフェッショナル版では、プロジェクトにおける制作環境と、携帯電話向けのSWFファイルを生成する機能が加わっていた。なお、「FlashMX2004」では、Web用アニメーションでよく使われる効果が「タイムラインエフェクト」として、Webのインターフェースとして使わ

れる効果が「ビヘイビア」として提供されるようになった。「ビヘイビア」は、「ActionScript」を半自動的に設定するものであり、プログラミング経験のないクリエイターもユーザーインターフェースを構築できるように設計されていた。

2005年5月には、「Macromedia」社が「Adobe」社に買収された。それにより、「Flash8」は、「Adobe」社から発売されることになった。「Flash8」では、ビデオ関連機能に重点が置かれ、「On2Technologies」社の「VP6」という高性能ビデオ配信コーデックが採用された。それにより、「WindowsMediaPlayer」などの動画プレイヤーの市場に食い込む可能性も出てきた。またグラフィックス関連では、ペイント系ソフトで使用される「フィルタ」や「ブレンディング」の機能が加わった。

なお、「Adobe」社によると、「Flash9」は、2007年の春に発売予定である。さらに、同社の「Creative Suite」にもバンドルされる予定である。これは同社の「Photoshop」、「Illustrator」、「AfterEffects」との連携が強まり、Flashのシェアが一層広がることを意味する。ただし、「ActionScript」は過去のスクリプトとの互換性がない「ActionScript3.0」を採用する予定であることから、より本格的なRIA開発ツールとして発売されることが予想される。

## 2 Flash および FlashPlayer の特徴

Flashの最大の普及要因は、「ベクトル画像によるアニメーション」が作成できたことであるが、「インターネットの一般利用の広がり」や「パソコンのCPUパワーの増大」とも深い関係がある。

教育内容としては、CGおよびインターネット映像配信の基本知識として、以下の3項目が重要である。

1. ベクトル画像の特徴
2. インターネットの時代に「FlashPlayer」が普及した要因
3. SWF形式による映像再生

特に1.においては2DCGの基本事項である「ビットマップ画像」と「ベクトル画像」の違いを含めるべきである。後述の3-1節でも述べているが、「インタラクティブなコンテンツの作成」に関しては、ビットマップ画像は使用しにくいといえる。

### 2-1 ベクトル画像の利点

Flashが普及した最大の要因は、「ベクトル画像によるWeb用アニメーション」が作成できたことである。ベクトル画像の利点は以下の4点である。

1. ビットマップ画像に比べて、一般的にデータ量が少ない
2. 拡大縮小、回転しても画像が乱れない
3. ビットマップ画像に比べて、編集、再利用が容易

#### 4. 3次元データを扱える

特に上記の1.の利点は、インターネットを通じた動画再生においてを非常に有利であった。ただし、ベクトル画像は「一般的に」データ量が少ないということに注意すべきである。ベクトル画像のファイルは圧縮できないので、複雑なグラフィックスのファイルはそれなりに大きくなる。上記の2.の利点も、特にインタラクティブなコンテンツでは、ビットマップ画像を使うより有利である。地図の表示などでは、画像の拡大縮小、回転などのユーザの操作を受け付けなければならないからである。上記の3.は、制作者サイドの利点である。画像は数値情報で保存されるため、制作後の編集や、画像の再利用が容易である。上記の4.は、3DCGソフトとの連携という点で有利である。「Swift3D<sup>(14)</sup>」というソフトは、Flashに読み込める形式で、3次元データを書き出すことができる。Web上で見られる、3DCGが混在したFlashは、この「Swift3D」あるいは3Dレンダリングエンジンを搭載した最近の「Director」を使用して作成されたものである。

#### 2-2 ベクトル画像の欠点

3-1では、ベクトル画像の利点をあげたが、ベクトル画像にも欠点がないわけではない。ベクトル画像は、モニタ上に再表示するたびにCPUで画像を計算するため、CPUのパワーを要求する。アニメーション再生において誤解されやすいのであるが、全ての画像を読み込んだ後では、GIFアニメなど「ビットマップ画像によるアニメーション」の方が速く再生できるのである。しかし、この欠点は、近年のCPUのスピードアップにより、次第に問題にされなくなった。もう1つの大きな欠点として、ベクトル画像には「フィルタ」を掛けられず、動画制作において徳に重要な「ブレ」を表現できないというものがある。「ブレ」は、物体、あるいはカメラを素早く動かしたときに起きるもので、数多くの映像作品で効果的に用いられている。しかし、この欠点は、ペイントソフトを併用すれば解決できる。ペイントソフトで「ぼかしフィルタ」をかければ「ブレ」のある画像が生成できるからである。原理的にビットマップ画像でなければ表現が難しいのが、デジタルカメラなどで撮影した写真である。Flashにもビットマップ画像を読み込む機能があり、ベクトル画像に自動的に変換（トレースと呼ばれる）機能もあるが、トレースされたビットマップ画像は、画質が悪くなりファイルサイズも大きくなる。また、「FlashPlayer」での再生も時間がかかる。ただし、「FlashMX2004」以降では、FLV形式でビットマップの映像を書き出すことが可能になった。それに伴い、「FlashPlayer7」以降でも、SWF形式の他にFLV形式がサポートされ、ビットマップの映像をベクトルに変換せずに再生することが可能になった。このFLV形式のサポートにより、最近の「Flash-Playter」は、ベクトル画像によるアニメーションの再生Playerとはいえなくなったのである。ただし、DVなどで撮影した長時間の映像（実際は画像が連続したもので、ビットマップ画像が1秒間につき30枚存在する）の再生に関しては、ストリーミング機能をもつ「Windows-MediaPlayer」の方が、「FlashPlayer」より適しているのは間違いない。「FlashPlayer」は

擬似ストリーミング機能しか持たないため、短い映像の再生用にしか用いられていない。FLV形式の動画ファイルを「FlashPlayer」で再生しているサイトは、「YouTube」<sup>(15)</sup>、「Google Video」<sup>(16)</sup>があげられる。

### 2-3 Flashの再生環境

Flashで制作した映像の再生環境は、完全普及しているといってよい。前述したとおり、日本にある98.3%のパソコンに、「FlashPlayer6」以降がインストールされている。この普及率の高さにはいくつか原因がある。そのうち最大の要因は、パソコンに詳しくない人でも、知らないうちにインストールできるということである。「FlashPlayer」がインストールされていないパソコンで、SWF形式やFLV形式のファイルを掲載しているWebページを閲覧すると、インストールするようにユーザが誘導される。ただし携帯電話の場合は、ユーザがインストールするわけではない。「DoCoMo」,「AU」のいずれの機種にも、「FlashLite」（機能を限定して軽量にした携帯版FlashPlayer）がプリインストールされている。「FlashPlayer」の利点は以下の4点である。

1. ダウンロードモジュールの小ささ
2. プラットフォームを多き
3. ベクトル画像によるアニメーションを再生できる
4. ビットマップ画像による映像も再生できる

上記の1. に関して言及すると、最新の「FlashPlayer9.0.16」でも、ファイルサイズは946KBである。現在では、インターネットを通じてクライアントアプリケーションをインストールするのが当たり前になった。したがって、ファイルサイズの小ささは重要である。2. に関しては、パソコン、PDA、PHS、携帯電話など、現在、「FlashPlayer」が動かないデバイスはないといっても過言ではない。3. と4. に関しては、当初の「FlashPlayer」は、ベクトルアニメーションしか再生できなかった。ところが最近の「FlashPlayer」は、FLVというビデオ形式をサポートするようになった。Webの動画配信サイト「YouTube」にも「FlashPlayer」が採用され、「QuickTimePlayer」のシェアを一気に奪うことになった。また、最近Yahooなどで見られる、音がでるバナー広告もFlashのFLV形式である。

### 2-4 FlashPlayer以外のクライアントモジュール

他の無料でダウンロードされるクライアントモジュールに関して言及する。Adobe社の「PDF」も、ダウンロードに関しては、同じ仕組みを持つ。しかし、Windows版「AcrobatReader7.0.8」<sup>(17)</sup>ダウンロードモジュールのファイルサイズは29.8MBである。Flashの実行環境に比べて、実に30倍の差がある。また、「AcrobatReader」には、Webアプリケーション開発環境が整備されていない。「Java」に関しては、アプリケーション開発環境は、完全に整備されている。また「Java」の実行環境は、Unix版、Windows版、Mac版はもちろ

ん、PDA版、携帯電話用まで存在する。プラットフォームを問わないという点で、Flashに類似している。しかし「Java アプレット」の実行環境である「JavaVM」は、「Sun microsystems」社のサイトで手動ダウンロードする必要がある。その点で、コンピュータに詳しくないユーザに負担をかけることになる。ダウンロードモジュールのサイズも小さいとはいえない。Windows版「JavaVM」は「Java Runtime Environment Version5.0 Update6」ダウンロードモジュールで16MBである。また、「Java アプレット」は、その一般性、機能の豊富さゆえに実行速度が遅いという欠点を持つ。特に画像を扱うことに限れば「FlashPlayer」に及ばない。実行速度や開発環境の点では、「ActiveX コントロール」は、利点を持つ。しかし、「ActiveX」は、Microsoft社の技術であり、Windowsにしか搭載されておらず、ブラウザでの利用は「InternetExplorer」に限られる。また、オペレーティングシステムの機能を利用しローカルのファイルにアクセスできるため（例えば、オンラインウィルススキャンなどはこの仕組みを利用している）、悪意のあるソフトウェアも開発できてしまう。開発元の「Microsoft」社も、この「ActiveX」の危険性を認めている。セキュリティの観点から、WindowsXPSP2の「InternetExplorer」では、「ActiveX」のダウンロードをブロックする設定になっている。「WindowsMediaPlayer」は、Windowsにバンドルされているのが特徴である。しかし、インタラクティブなアプリケーション制作のための開発環境と呼べるものではなく、純粋な動画（WMV、MPEG形式）、音楽（WMA、MP3形式）の再生Playerである。これは、「RealPlayer」についても同様である。

## 2-5 SWF 形式

Flashが出力するSWF形式は、デジタルカメラやスキャナで取り込んだビットマップ画像とは異なり、複雑形状でなければ、ビットマップ画像のファイルサイズより小さくなる（ただし、ビットマップ形式とは異なり、圧縮はほとんどできない）。このSWF形式のファイルのサイズの小ささが、GIFアニメーションに対して、大きなアドバンテージを持つことになった。現在のWebサイトでは、非常に小さいものにしかGIFアニメは使用されていない。SWFファイルは、以下のように数値的な情報を保持している形式である。

```

$s = new SWFShape();
$f = $s-> addFill($g, SWFFILL_RADIAL_GRADIENT);
$f-> scaleTo(0.03);
$s-> setRightFill($f);
$s-> movePenTo(-30, -30);
$s-> drawLine(60, 0);
$s-> drawLine(0, 60);
$s-> drawLine(-60, 0);

```

```
$s-> drawLine(0, -60);
```

「FlashPlayer」は、SWF ファイルに保存された数値情報を元に計算して、パソコンなどの画面に描画する。この「計算」という要素が入るため、Flashの再生速度はパソコンのCPUパワーに異存する。ところが、近年のパソコンはCPUパワーが急速に上がった。複雑なベクトル画像によるアニメーションの再生でも滑らかに行われる。CPUパワーの増大も、「Flash-Player」が普及した一因である。

### 3 Flashのグラフィックス

Flashではベクトル画像を描ける。ベクトル画像の特徴は、データ量が少ないことと、解像度に依存しないことである。他にベクトル画像を描ける2DCGグラフィックスソフト（3DCGソフトはその性質上、必ずベクトル画像を扱う）は、有名なものとしては、「Adobe」社の「Illustrator」がある。「Illustrator」に慣れている学生には、「Illustrator」で描いたグラフィックスを読み込ませても良い。他のグラフィックスツールにない特徴として以下の4点があげられる

1. 線と塗りが、合体したり、分割したり、型抜きされたりする
2. 描いた後の線と塗りを、マウスでドラッグして変形できる
3. 「ブラシツール」で描いた塗りに対して、補正が自動的にかかる
4. 「シンボル」の生成の仕方、および「シンボル」の編集モードへの入り方が、それぞれ

2通りある

これらの特徴を教えれば、他の2DCGソフトに慣れていない学生でも、それなりの絵が描けるようになる。特に注意したい点は、4. である。以下でも述べるが、「シンボル」と「インスタンス」の関係を理解させなければならない。ただし、アニメーションやRIAの制作を視野にいれなければ、「シンボル」の使用は必須というわけではない。「Flash」でのグラフィックスの描き方の最小限の指導内容は以下の通りである。

1. 「線ツール」、「楕円ツール」、「矩形ツール」で、大まかな形状を線で描く
  2. 線を「選択ツール」でドラッグ&ドロップして修正
  3. 線で囲まれた箇所を「バケツツール」や「ブラシツール」で塗りつぶす
  4. 細かい箇所はさらに線を追加して、「バケツツール」や「ブラシツール」で塗り合わせる
- 上記の手順は、大まかなものであり、Flashでグラフィックスを描く際に、教えるべき事項は以下の節で述べる。

#### 3-1 ユーザインターフェースに関する項目

Flashのユーザインターフェースは、使いづらいというユーザが多い。しかし、これはむしろ「Macromedia」社のインターフェース設計が独特であるといったほうがよい。他の「Ma-

cromedia」社のソフトを使った経験のない学生には、常に「プロパティインスペクタ」に注目するという指示を与えるとよい。以下に重要な操作を挙げる。

- 「Ctrl キー」 + 「Z キー」で操作の取り消し（100回まで可能）
- 「Ctrl キー」 + 「Y キー」で操作の取り消し自体の取り消し
- いかなるツールが選択されていても、「Space キー」を押しながらドラッグすると、「ハンドツール」で画面移動ができる
- 「ハンドツール」をダブルクリックすると、「ステージ」が限界まで広く表示される
- いかなるツールが選択されていても、「Ctrl キー」と「+キー」の同時押しで拡大表示、「Ctrl キー」と「-キー」の同時押しで縮小表示ができる
- 「虫めがねツール」で、ある部分を囲むようにドラッグすると、その部分が拡大される
- 「虫めがねツール」をダブルクリックすると、100%表示になる
- 「プロパティインスペクタ」で、現在選択されている「オブジェクト」の属性情報を得られるだけでなく、「オブジェクト」の属性を変更することができる

### 3-2 描画に関する項目

Flash の描画機能は独特なもので、「Illustrator」に慣れ親しんだユーザは戸惑いやすい。特に、Flash の「ペンツール」は、現在に至るまで使い勝手が悪く、「ペンツール」を使いたい場合は、「Illustrator」の「ペンツール」で作ったグラフィックスを読み込む手法が常套手段とされてきた。Flash での描画を教育する場合は、「ペンツール」ではなく、基本的なグラフィックスを変形させることを主眼に置いたほうがよい。

- 「ステージ」に描かれたグラフィックスだけが、ムービーとしてパブリッシュされる
- 線と塗りは全く別々に扱われる
- 線と塗りは「シェイプ」になる
- 「線ツール」、「直線ツール」、「ペンツール」、「鉛筆ツール」が、線を描くツールである
- 「ブラシツール」が、塗りを描くツールである
- 「楕円ツール」や「矩形ツール」は、線と塗りを同時に描くツールである。
- 「楕円ツール」や「矩形ツール」で、「Shift キー」を押しながらドラッグすれば、正円や正方形の「シェイプ」を描ける
- 「ソフトエッジ」は塗りにしか適用できない
- 同じレイヤー上の「シェイプ」は、他の「シェイプ」を重ねて、選択を解除することで、色が同じならば「合体」させられる。色が違うならば「分割」、「型抜き」される
- 「シェイプ」の種類が「線」である場合は、「合体」と「型抜き」は起きない
- 複数の線を重ねると、線と線の区切りごとに分割された線として認識される

- 「インクボトルツール」で、塗りをクリックすると、塗りの周りに線を描ける
- 「バケツツール」で、閉じた線、あるいは閉じた塗りの中に、塗りを描ける
- 「カラーミキサーパレット」で塗りの色に「ライブラリ」にあるビットマップ画像を設定できる
- 塗りに設定したビットマップ画像は「塗りの変形ツール」で拡大縮小ができる
- 「ステージ」に置いたり、「ライブラリ」に読み込んだりすることができるビットマップ画像は、JPEG、GIF、BMP、PICT、PNG形式である
- ビットマップ画像を「シェイプ」に変換するには、「分解」を選択する
- 「テキスト」を「シェイプ」に変換するには、「分解」を2回選択する
- 「シェイプ」は「矢印ツール」でドラッグすることにより変形できる
- 「シェイプ」の変形の際に、角を作りたい場合は、「Ctrl キー」を押しながらドラッグする
- 「エンベロープ」を使用すれば、「シェイプ」を「ペンツール」で歪曲させることができる
- 「消しゴムツール」のオプションの「流し消す」でつながった部分だけを消去できる
- 「消しゴムツール」は、「レイヤー」をロックしない限り、全ての「レイヤー」の「オブジェクト」に影響を及ぼす
- 「矢印ツール」で選択して「BackSpace キー」または「Delete キー」を押しても、「オブジェクト」を消去できる
- 「Ctrl キー」+「Shift キー」+「V キー」で、「オブジェクト」を同じ位置にペーストする

### 3-3 シンボルに関する項目

「シンボル」の作り方は、既に描いた絵を「シンボル」として登録する方法（この場合、ステージにある絵は瞬時に「インスタンス」になる）と、新規に「シンボル」を作成する方法の2通りがある。「シンボル」の編集モードへの入りも、「ライブラリ」の中の「シンボル」をダブルクリックする方法と、「ステージ」上の「インスタンス」をダブルクリックする方法がある。また、「インスタンス」に個別に「カラー」や「アルファ」を設定できることも「シンボル」と「インスタンス」の関係を理解させる際に混乱を生む原因になるので、特に注意すべきである。なお、「ステージ」に配置された「シンボル」は、全て「インスタンス」である。書籍によっては、「シンボル」の「インスタンス」を指して、「シンボル」と呼んでいる。（ただし、「ステージ」上に配置された「シンボル」ならば、間違いとはいえない）これは大きな間違いであり、「シンボル」を階層化するときや、スクリプトを書く際に、混乱の元になる。本論文では、以下「シンボル」の「インスタンス」を「インスタンス」と書く。

- 「オブジェクト」を「シンボルに変換」すると、「シンボル」として「ライブラリ」に登録

される

- 「F11キー」で「ライブラリ」を表示する
- 「オブジェクト」を選択してから「F8キー」で「シンボル」に変換する
- 「シンボル」には、「ムービークリップ」、「ボタン」、「グラフィック」の3種類がある
- 「シンボル」の編集は、「ライブラリ」に登録されている「シンボル」をダブルクリックするか、「ステージ」上の「インスタンス」をダブルクリックする
- 「シンボル」自体を編集すると、全ての「インスタンス」に編集結果が反映される
- 「シンボル」を「ステージ」に配置すると「インスタンス」になる
- 「ステージ」上に配置された、「シェイプ」、「テキスト」、「グループ」、「インスタンス」は全て「オブジェクト」である
- 複数の「オブジェクト」を「1つのオブジェクト」として扱いたい場合は、「グループ化」や「シンボル変換」を行う
- 「オブジェクト」は「Ctrl キー」を押しながらドラッグ&ドロップすることで、コピーできる
- 「オブジェクト」は「Shift キー」を押しながら選択することで、複数同時選択ができる
- 「オブジェクト」のピクセル単位の移動は、「カーソルキー」を使って行う
- 「オブジェクト」を厳密に変形するには、「変形ウインドウ」を使用する
- 「グループ」、「テキスト」、「インスタンス」は、他の「オブジェクト」から切り抜きや合体などの影響を受けない
- 「グループ」、「テキスト」、「インスタンス」は、他の「オブジェクト」より手前に表示される
- 「グループ」、「テキスト」、「インスタンス」は、「重ね順」で、表示の優先度を変更できる
- 「グループ」、「テキスト」、「インスタンス」には「変形点」があり、小さい「○」で表示される
- 「変形点」は、ActionScript では、一切扱うことができない
- 「グループ」、「テキスト」、「インスタンス」には「基準点」があり、小さい「+」で表示される
- 「グループ」、「テキスト」、「インスタンス」の拡大縮小、回転、歪曲、伸縮は、「変形点」を基準として行われる
- 「グループ」、「テキスト」、「インスタンス」の位置は、「基準点」を基準として「プロパティインスペクタ」に表示される
- 「グループ」、「インスタンス」を編集する場合は、ダブルクリックしてから編集する
- 「インスタンス」を編集することは「シンボル」自体を編集することになり、ステージ上の全ての「インスタンス」に編集結果が反映される
- 「インスタンス」を「分解」すると、「インスタンス」ではなくなる（もとの「シンボル」

との関係がなくなる)

- 「シンボル」の中に「シンボル」を入れることにより「シンボル」を階層化できる
- 「インスタンス」はデータ量をほとんど増やさない
- 各「インスタンス」には、変形、回転角度、カラー、アルファなどを個別設定できる
- 「インスタンス」を選択して「プロパティインスペクタ」の「シンボルの入れ替え」を選択すると、既存の「シンボル」（正確には、「シンボル」の「インスタンス」）と置き換えることができる
- 他のファイルの「ライブラリ」に登録されている「シンボル」も使用できる
- 「ライブラリ」で「シンボル」を削除すると、その「シンボル」の「インスタンス」も自動的に削除される
- 「ライブラリ」には、「シンボル」以外にも、サウンド、ビットマップ画像、ビデオ映像を読み込める
- ビットマップ画像は、「シンボル」に変換して、「インスタンス」として、ステージに配置しなければ、アルファの設定ができない
- 「ビットマップのトレース」で、ビットマップ画像をベクトル画像に変換できる

### 3-4 パブリッシュに関する項目

パブリッシュでは、通常、SWF形式とSWFが埋め込まれたHTML形式のファイルを出力する。ただし、Flashは、「ステージ」のサイズ以外の大きさを、SWF形式のファイルのパブリッシュできない。したがって、最初の段階で、「ステージ」の「幅」と「高さ」を設定する必要がある。「幅」と「高さ」の設定は、なにも「オブジェクト」を選択しない状態の「プロパティインスペクタ」で行える。

- 「テキスト」で「デバイスフォント」を使用すれば、OSのシステムフォント（再生環境によって異なる）が使用されるため、データ量を減らせる
- 「パブリッシュ設定」で「AI」、「JPEG」、「GIF」、「PNG」形式で書き出せば、「Illustrator」、「Photoshop」で加工できる
- 「パブリッシュ設定」で「Windows プロジェクト」形式で書き出せば、「FlashPlayer」がインストールされていないパソコンでも実行できる
- 「パブリッシュ設定」で「wmf」形式で書き出せば、Officeアプリケーションのクリップアートとして使用できる
- Flashのソースファイル（FLA形式のファイル）は、そのバージョンか、1つ前のバージョンの形式でしか保存できない。例えば、「Flash8」では、「FlashMX」（Flash6）で読み込めるような形式では保存できない。もし、そのような必要がある場合には、「MX2004」（Flash7）形式で一旦保存し、「FlashMX2004」を起動し読み込ませ、MX形

式で保存しなおすという手順を踏む

- 「パブリッシュ設定」で、「読み込みの禁止」にチェックを入れれば、SWF ファイルを Fla ファイルに読み込めなくなる
- Web ページに表示された SWF ファイルは、ブラウザによってキャッシュされてしまう。したがって、HTML のソースファイルで、キャッシュを止めない限り、SWF ファイルのダウンロードは防げない

#### 4 Flash のアニメーション

Flash を使用すれば、イラストの下手な学生でも、滑らかに複雑な動きのアニメーションを制作できるというわけではない。これが、3DCG ソフトウェアと Flash との大きな違いである。3DCG では、「ボーン」や「インバース・キネマティクス」など、関節があるものを滑らかに動かすためのツールが揃っている。動きを指定すれば、コンピュータの計算により、アニメーション画像を出力する。Flash では、このような補助機能は、「キーフレーム」間を単純に補間するトゥイーン機能しかない。何枚かの「キーフレーム」用の画像は、必ずグラフィックスとして描かなければならない。一般的に動きが複雑であればあるほど、必要とされる「キーフレーム」画像は増えることになる。Flash でアニメーション制作する際に教えるべき事項は、以下の節で述べる。

##### 4-1 Flash を用いたアニメーション制作

Flash になじみのない学生は、アニメーションといえば、ストーリーに沿った、キャラクターやカメラのダイナミックな動きを連想するはずである。しかし、映画や TV でみるようなアニメーションを Flash で制作することは難しい。キャラクターが歩いたり走ったりするアニメーションは原画を描くのが難しく、そのような場面の原画は経験豊富なアニメーターが担当する。まず、Flash を学ぶ学生には、TV アニメーションの固定観念を捨てさせ、バナー広告で使用されるモーショントイポグラフィから制作させるとよい。モーショントイポグラフィは、テキストが動くアニメーションである。しかし、「レイヤー」、「タイムライン」、「フレーム」、「キーフレーム」、「空白キーフレーム」、「ムービークリップ」、「モーシントゥイーン」などの Flash 動画における重要事項を学ぶのには適している。イラストを描かなくていいため、動かすことに集中できるからである。また、イラストを描くのが苦手な学生でも、綺麗な作品を制作できるのが特徴である。

モーショントイポグラフィを学ばせた後に、いわゆる「モーシヨングラフィック」を制作させるとよい。イラストやグラフィックデザインを動かす簡単なアニメーションは、従来のキャラクターアニメーションと区別するために「モーシヨングラフィック」と呼ばれる場合が多い。モーシヨングラフィックでは、グラフィックデザインとモーシヨンドesign両方のセンスが要求される。ただし、最初はアニメーションさせることを考えさせず、静止画を描かせる。静止

画のイラストを眺めさせていると、画像の中で「ここが動いたら面白い」という部分が見えてくるはずである。動かす部分は最小限に抑えさせる。また、動かすといっても、「キーフレーム」における複雑な変化を描かなくてはいけないものは避けるべきである。例えば、「背景の一部が流れていく動き」などが適切である。このような動きであれば、以下のスリーステップで実現できる。

1. 動かしたいものを「シンボル」に変換
2. 2つの「キーフレーム」で、「シンボル」の位置を決める
3. 始めの「キーフレーム」の「プロパティインスペクタ」で「モーショントゥイーン」を設定

単純な「モーショントゥイーン」のみでは、アニメーションとしての完成度は低い。しかし、もともとの静止画のクオリティが高ければ、それなりの作品が制作できる。なお、Flash単体では、3DCGアニメーションは作成できない。3Dの物体を描く機能がFlashにはないからである。ただし、「Electric Rain」社の「Swift3D」では、ベクターレンダリングエンジンを搭載しているため、3D空間で、モデリングを行った物体を、ベクトル画像として出力できる。出力されたベクトル画像を、Flash（MX以降）に取り込むプラグインもリリースしている。

#### 4-2 フレームアニメーションとトゥイーンアニメーション

Flashにおけるアニメーションの手法には、「フレームアニメーション」と「トゥイーンアニメーション」がある。「フレームアニメーション」は「フレーム・バイ・フレーム」とも呼ばれ、1フレームごとに少しずつ異なるグラフィックスを使う手法である。Flashでは、「キーフレーム」に新たなグラフィックスを描いていく。基本的に、小学生が教科書の端に書いて遊ぶ「パラパラアニメ」と同じであり、全ての場面でこの手法を用いればFlashを使う意味はない。グラフィックスを並べて、再生タイミングを調節できるソフトであれば、どんなソフトを使っても同じであり、いずれにせよ膨大な手間がかかることになるからである。また、人間が走る動きなどの複雑なアニメーションは、アニメーターでなければ実現不可能である。複雑な動きを演出するために「少しずつ異なったグラフィックスを描く」ことは、熟練したアニメーターにとっても難しい。Flashを用いる利点は、「フレームアニメーション」と「トゥイーンアニメーション」の使い分けができることにある。

「トゥイーンアニメーション」では、「キーフレーム」の間のグラフィックス（アニメーション用語でいう「中割り」）は、計算により自動的に描かれる。それにより、手間とデータ量は大きく減る。「トゥイーンアニメーション」には、「モーショントゥイーン」と「シェイプトゥイーン」があり、「モーショントゥイーン」は、基本的には「インスタンス」に、「シェイプトゥイーン」は必ず「シェイプ」に対して設定する。ただし、「モーショントゥイーン」の単独使用では、単純な動きしか実現できない。「モーショントゥイーン」を使用する場合は、まず動かしたい物体の各部分を、「ムービークリップシンボル」として、バラバラに描くことが重

要である。バラバラに部品を描き、複雑なアニメーションを実現する手法は以下の通りである。

1. 各部品を「ムービークリップシンボル」とする
2. 各「ムービークリップシンボル」のタイムラインで「モーショントウイーン」あるいは「フレームアニメーション」を設定
3. 各「ムービークリップシンボル」を「インスタンス」として、別の「ムービークリップシンボル」の「タイムライン」に配置、必要があれば、その「タイムライン」で、再び「モーショントウイーン」あるいは「フレームアニメーション」を設定

この手順で、アニメーションを入れ子にして、階層化することができる。階層化されたアニメーションでは、ムービーの中でムービーを再生することになる。ただし、「ムービークリップ」は、「タイムライン」の最後に「stop();」というスクリプトを書かない限り、無限にループするので注意が必要である。「ムービークリップ」は「シンボル」であるから、階層化だけでなく、同じアニメーションを使いまわすため使用できる。この「ムービークリップシンボルの使いまわし」と「階層化アニメーション」は、Flashを用いたアニメーション制作の常套手段となっている。なお、「ムービークリップ」の階層化の様子を調べたい場合は、「ムービーエクスプローラ」を表示する。

#### 4-3 アニメーションに関する項目

「キーフレーム」に「インスタンス」を置いて、「モーショントウイーン」を設定させるのが基本である。微妙な動きは、「モーショントウイーン」の前後に「キーフレーム」を挿入して、フレームアニメーションで実現させる。いずれにせよ、最初にFPS (Frame Per Second) の設定が必要である。FPSとは、1秒に何フレーム再生するかである。TVは30、映画は24である。Flashは、初期状態では12FPSであり、滑らかなアニメーションを制作したい場合は、値の変更が必要である。特に滑らかな動きが必要な場合は60FPS程度にするが、CPU速度が遅いマシンでは処理落ちしやすい。通常は30FPS程度に設定する。

- Flash立ち上げ時に、はじめから表示されている「タイムライン」が「\_rootタイムライン」であり、必ず一つしかない
- 「フレームの挿入」で、「タイムライン」上に「フレーム」を挿入する
- グラフィックスは必ず、「キーフレーム」に描かれる
- 再生タイミングの変更は、「キーフレーム」の移動によって行う
- 「キーフレーム」の後は、新たな「キーフレーム」か「空白キーフレーム」を挿入しない限り、「キーフレーム」の画像が表示されつづける
- 「キーフレーム」、「空白キーフレーム」は、「タイムライン」上でドラッグ&ドロップすることにより移動できる
- アニメーションを設定しない部分は、1つの「キーフレーム」に描いておけばよい

- 「F5キー」で「フレーム」の挿入, 「Shift キー」 + 「F5キー」で「フレーム」を削除, 「F6キー」で「キーフレーム」の挿入, 「Shit キー」 + 「F6キー」で「キーフレーム」の削除, 「F7キー」で「空白キーフレーム」の挿入
- 「レイヤー」を「ガイドレイヤー」に変更すれば, パブリック時に表示されなくなる
- 「レイヤー」には重ね順があり, 上の「レイヤー」にあるグラフィックスほど優先的に（手前に）表示される
- 「レイヤー」が増えて「タイムライン」が見つらなくなったときは, 「レイヤーフォルダ」を使用する
- 「マスキレイヤー」の塗りがある部分にのみ, 「マスク」の対象となる「レイヤー」が表示される。「直線ツール」, 「鉛筆ツール」, 「ペンツール」で描いた線は, いくら太くても「マスク」にならない
- 「オニオンスキン」を使用すれば, 複数の「キーフレーム」を表示でき, 「オニオンスキーマーカー」の左端と右端をドラッグすれば, 表示範囲を調節できる
- 「フレームアニメーション」を作成するときは, 「オニオンスキン」あるいは「オニオンスキニアウトライン」を使用する
- 「複数フレームの編集」を使用すれば, マーカーの範囲内にある「キーフレーム」のグラフィックスをまとめて編集できる
- 「インスタンス」, 「グループ」および「テキスト」が選択されたとき, 「インスタンス」の周囲に, 青いバウンダリーボックスが表示される
- 周囲に青いバウンダリーボックスが表示されている「オブジェクト」（すなわち, 「インスタンス」, 「グループ」, 「テキスト」）には, 「モーショントゥイーン」が設定できる
- 「モーショントゥイーン」は, 開始時点の「タイムライン」上にある「キーフレーム」で設定する
- 「モーショントゥイーン」は, カラー, アルファに関する中割り画像も生成できる
- 「モーショントゥイーン」の「イー징」は「イン」が加速で, 「アウト」が減速である
- 「モーショントゥイーン」の「開始キーフレーム」と「終了キーフレーム」には, 同じ「シンボル」の「インスタンス」が配置されていなければならない
- 「モーショントゥイーン」で, 「ガイドレイヤー」を使用すれば, 線に沿って移動させることができる。「インスタンス」, 「グループ」, 「テキスト」の変形点を, 「ガイドレイヤー」上の線の両端にあわせる
- 「モーショントゥイーン」設定時に, 「パスに沿って回転」にチェックを入れるとパスに合わせて向きが変わるトゥイーンが設定できる
- 1つの「レイヤー」につき, 1つの「オブジェクト」にしか「モーショントゥイーン」を設定できないが, 「モーショントゥイーン」を設定した「ムービークリップインスタンス」は, 1つのレイヤーにいくつでも置ける

- 「ムービークリップインスタンス」は、「Enter キー」を押す簡易プレビューでは再生されない
- 「ムービークリップインスタンス」を、簡易プレビューで再生したい場合は、「プロパティインスペクタ」で、「グラフィックインスタンス」に変える
- 複数の「オブジェクト」を同時選択して、「レイヤーに配分」を選択すると、「オブジェクト」の数だけ「新規レイヤー」が作られ、各レイヤーの第1フレームに「オブジェクト」が配置される
- 「シェイプトゥイーン」が設定できるのは、「シェイプ」のみである
- 「シェイプトゥイーン」でうまくモーフィングしない場合は、「シェイプヒント」を付ける
- 「ライブラリ」に読み込めるサウンドファイルは、WAV、AIFF、MP3形式である
- サウンドは必ず、「キーフレーム」に設定される
- サウンドの形式には、「イベントサウンド」と「ストリーミングサウンド」がある
- 「イベントサウンド」は、「タイムライン」に依存せず、再生が始まったら最後まで再生される
- 「イベントサウンド」には、「イベント」、「スタート」、「ストップ」がある。「スタート」は、同じサウンドが再生されている場合は、重ねて再生しない。「ストップ」は、同じサウンドを停止する。「イベント」は、同じサウンドを重ねて再生する
- 「ストリーミングサウンド」は、サウンドを配置した「タイムライン」の長さに依存する。「タイムライン」が空白になれば、そこで再生も止まる

#### 4-4 アニメーションのパフォーマンスに関する項目

ベクトル画像によるアニメーションは、利用者の環境に大きく左右される。フレームレート (fps) はあくまで目安にしかならず、再生スピードはCPUの処理速度に依存する。ターゲットとするCPUで再生がもたつくようであれば、描画に負担がかかる処理を減らす必要がある。誤解されやすいのであるが、Flashのファイルサイズと表示速度には関連がない。ファイルサイズが小さい場合はダウンロードの時間が短くなるが、必ずしもアニメーション表示が速いというわけではない。

- 「モーショントゥイーン」を積極的に使い、「キーフレーム」を少なくする。フレームアニメーションを使うでも、「キーフレーム」ごとの変化量をなるべく抑える
- 「シェイプトゥイーン」を使用しない
- ビットマップ画像をアニメーションさせない
- グラデーション、アルファを使用しない
- 「修正」メニューから曲線の最適化を選択し、線の形状を単純化する
- 文字を「分解」して、「シェイプ」にする

## 5 Flash の ActionScript

もともとアニメーション制作ツールだった Flash が、RIA 制作ツールとして使われるようになったのは、「ActionScript」が存在したからである。Flash で、インタラクティブなコンテンツを制作する場合、「ActionScript」は避けて通れない。プログラミング経験者にとって、「ActionScript」を覚えるのは難しくない。しかし、「タイムライン」、「キーフレーム」といった Flash の基礎事項を知らなければ、実際にプログラミングすることはできない。逆にプログラミング経験のない学生が、本格的なオブジェクト指向言語（クラス設計、継承など）として使いこなすには、相当な量のトレーニングが必要である。教育機関で「ActionScript」を教える際には、内容の絞り込みが必要である。5-8 を除く以下の節では、学生に直接教育する事項ではなく、教員が知っておくべき事項を述べる。5-8 では、インタラクティブなコンテンツを制作するために必要なメソッド（命令）を挙げる。

### 5-1 アクション

「ActionScript」になる前の「アクション」と呼ばれたスクリプト言語は、Flash のムービーに、インタラクティブ性を加えるための最小限の機能であった。具体的には、「タイムライン」の制御、「ボタンシンボル」の「インスタンス」に対するイベントの取得、「ムービークリップシンボル」の「インスタンス」のマウスやキーボードでの移動などである。これらのスクリプトは、プログラム経験のない Web デザイナーやグラフィッカーによって書かれていた。なぜ、プログラム経験がなくても「アクション」のスクリプトを書けるのかという理由は、「フレーム」や「インスタンス」に対して、直接スクリプトを記述することができたからである。プログラミングで最も難しいのは、現実世界の現象を抽象化して手続きに落とし込むことである。その点で、Flash でのスクリプティングは、もともとアニメーションがある状態で行うため抽象度が低いといえる。「アクション」によるスクリプティングは、プログラミングの経験のない学生にとっても、直感的に理解しやすい。

### 5-2 ActionScript と ECMAScript

「ActionScript」は、「Flash5」から導入されたオブジェクト指向プログラミング言語である。「Flash5」と「FlashMX」における「ActionScript」は、「ActionScript1.0」であり、「FlashMX2004」と「Flash8」では、「ActionScript2.0」である。「ActionScript2.0」では、class 文とデータ型の宣言が加わっているが、「ActionScript1.0」で書かれたコードもそのまま動く。いずれも、「JavaScript」の国際標準規格である「ECMA-262」に準拠しているので、構文や基本的なクラスは「JavaScript」と同一である。

ECMA とは、1961年に「European Computer Manufacturer Association」（欧州電子計算機工業会）として、欧州のコンピュータメーカーを中心に設立された組織である。1994年に、「ECMA International」と改名され、国際的な標準化組織となった。「ECMA-262」に準拠し

た Script 言語は、「ECMAScript」と呼ばれる。ただし「ECMAScript」はあくまで仕様である。実装には、「ActionScript」の他に、「Netscape」社の「JavaScript」,「Microsoft」社の「Jscript」がある。そもそも、「JavaScript」と「Jscript」の仕様が異なっていたため、「ECMAScript」として標準化されたという経歴がある。「ECMAScript」は、手続き型の言語、すなわちC言語系であり、「ActionScript」もC言語に似ている。したがって、「ActionScript」を学ぶことは、「JavaScript」や「Jscript」はもちろん、「C」,「C++」,「Perl」,「Java」など他のC言語系言語の習得にも有益であるといえる。来春発売予定の「Flash9」で採用予定である「ActionScript3.0」では、クラスやメソッドの見直しにより、より「Java」に近い言語になる。

「ActionScript」はSWF（Flashの実行ファイル）に書き出されると、バイトコードという命令に変換される。このバイトコードを解釈して実行するのが、「Flash Player」（SWF ファイルを再生するためのプラグイン）の「仮想マシン」である。この仕組みは「Java」と同じである。仮想マシンを用いることによって、パソコン、PDA、携帯電話などデバイスを選ばず、Flashが実行できることになる。逆の見方をすれば、各デバイスに対応した「Flash-Player」をインストールしなければならないということになる。

### 5-3 ActionScriptの問題点

「ActionScript」は成長し続けてきた言語であるといえる。当初は、プログラムの概念になじみがないデザイナーやグラフィッカーを対象とした、言語とは呼べないものであった。それは、「Flash4」までは「アクション」と呼ばれていたことから明らかである。「アクション」は、アニメーションを制御し、インタラクティブなコンテンツを制作するために設計された。そのため、ごく少数の関数が用意されているだけであった。

ところが、Flashはアニメーション制作ツールに留まらず、Webのユーザインターフェースの制作に使用されることが増えていった。それにより、Flashの開発者の層が広がり、従来のプログラマーも参入してきたのである。「Flash5」では、「ECMAScript 第3版」<sup>(19)</sup>に準拠した、「ActionScript1.0」が採用されることになった。さらに、「FlashMX2004」では、「ECMAScript 第4版」<sup>(20)</sup>に準拠した、「ActionScript2.0」が採用された。もともと「ECMAScript」は、「JavaScript」と「JScript」を標準化したものである。「アクション」から「ActionScript」への変更は、従来からWebプログラミングを行ってきたプログラマーにとっては、確かに有益であったといえる。

「ActionScript」の問題点は、スクリプト間の互換性を保ってきたところにある。Flash h1～Flash4の「アクション」、Flash5～FlashMXの「ActionScript1.0」、FlashMX2004～Flash8の「ActionScript2.0」という3つの言語の混在が許されている。しかも、同じ言語でも、Flashのバージョンにより、使用できるオブジェクトやメソッドが異なる。この混在は、時として深刻な問題を引き起こす。例えば、最新バージョンのFlash8でもFlash1からの古いアク

ションが存在する。例えば、「gotoAndPlay()」はアクションとしても、メソッドとしても存在する。メソッドとして使用する場合には、「ターゲットパス.gotoAndPlay()」と記述し、アクションとして使用する場合には、「gotoAndPlay()」と記述する。ただし、シーンを移動できるのは、アクションの場合のみで、

```
gotoAndPlay("scene2",1);
```

と記述しなければならない。上記の文の意味は「シーン2の1フレームに移動して再生」であるが、これを以下のように記述すると動作しなくなる。

```
my_mc.gotoAndPlay("scene2",1);
```

上記で「my\_Mc」は、「ムービークリップ」のインスタンスに対するターゲットパスであり、「gotoAndPlay()」は、「ムービークリップ」のメソッドである。このとき、エラーメッセージを出さないで、SWFファイルにコンパイルされてしまうのも、混乱を引き起こす原因となっている。

もう一つの問題点は、本格的なプログラミング言語となる過程で、当初の「手軽さ」を失ってしまったということである。それでも、現在の「ActionScript2.0」では、「ActionScript1.0」や以前の「アクション」の記法や、命令も使える。しかし、来春リリース予定の「Flash9」の「ActionScript3.0」では、厳密な型宣言が必須となっただけでなく、「on(release) {}」などの「オブジェクトアクション」や、「onEnterFrame」などの「イベントハンドラメソッド」も使えなくなる。

#### 5-4 ActionScript の記述箇所

「ActionScript」は、「フレーム」か「シンボル」の「インスタンス」に記述される。ただし、「グラフィックスシンボル」の「インスタンス」にはスクリプトを記述できない。「ActionScript」の種類は、それが記述される場所によって以下の3つに分類される。

1. フレームアクション
2. ボタンアクション
3. ムービークリップアクション

「フレームアクション」は、「タイムライン」中の「キーフレーム」に記述される。「キーフレーム」は、通常の「キーフレーム」（●で表される）でも「空白キーフレーム」（○で表される）であってもよいが、スクリプト専用のレイヤーを作成して、「空白キーフレーム」への記述が推奨されている。

「ボタンアクション」は「ボタンシンボル」の「インスタンス」に、「ムービークリップアクション」は「ムービークリップ」の「インスタンス」に記述される。「ボタンアクション」と「ムービークリップアクション」をまとめて「オブジェクトアクション」と呼ぶ。

ただし、「ムービークリップアクション」には特別の注意が必要である。「ムービークリップ」は独自の「タイムライン」を持つ。したがって、「ムービークリップ」へのスクリプトの

記述は、「ムービークリップシンボル」の「タイムライン」中の「フレーム」に記述する場合と、「ムービークリップインスタンス」に直接記述する場合の2通りがある。前者は「フレームアクション」、後者は「ムービークリップアクション」に分類される。「ムービークリップシンボル」から生成された「インスタンス」は、全て共通の「フレームアクション」を持つことになる。「ムービークリップインスタンス」のコピーを行っても、「ムービークリップアクション」はコピーされない。

### 5-5 ActionScript のイベントハンドラ

スクリプトは、必ずなんらかのイベントが起きたときに実行される。「ActionScript」が記述される場所は3箇所ある。書く場所によって、イベントを捕まえる機能（ハンドラ）が違う。「フレームアクション」の場合は、イベントハンドラは明示的に記述する必要がない。再生ヘッドが、フレームに移動した時点でスクリプトが実行される。「オブジェクトアクション」では、「インスタンス」が、「ムービークリップ」か「ボタン」で、イベントハンドラが異なる。「ムービークリップインスタンス」に記述する場合は「onClipEvent」、「ボタンインスタンス」に記述する場合は「on」（「onハンドラ」は、「FlashMX」以降は、「ムービークリップインスタンス」にも記述できるようになった）であり、それぞれ以下のように記述する。

ムービークリップインスタンスのイベント処理：

```
onClipEvent (イベント名) {ステートメント}
```

ボタンインスタンスのイベント処理：

```
on (イベント名) {ステートメント}
```

上記の「onClipEvent」ハンドラは、「Flash5」で導入されたイベントハンドラであり、「アクション」パネルの「アクション」-「ムービークリップ制御」カテゴリにある。「onハンドラ」は「Flash2」で導入されたハンドラであり、「アクション」-「ムービーコントロール」カテゴリにある。これらのハンドラは「ActionScript1.0」までの記述方式であるが、「ActionScript2.0」でも有効である。

ただし「FlashMX」以降では、「イベントハンドラメソッド」が実装され、イベントハンドラを、「フレームアクション」として記述することができるようになった。

### 5-6 ActionScript のクラス

「ActionScript」における「クラス」は、「シンボル」と大きな類似性がある。「シンボル」は「インスタンス」に対して、「タイムライン」やグラフィックスを提供する。それに対し、「クラス」は、「インスタンス」に対して、プロパティ、イベントハンドラ、メソッドを提供する。「ActionScript」では、すべての「クラス」が各々異なる prototype というオブジェクトを備えていて、各クラスのプロパティやメソッドは、prototype オブジェクトに取められている。「インスタンス」は、対応するクラスの prototype から、プロパティやメソッドを継承

する。それらのプロパティの値を変更したり、メソッドを使用したりすることで、プログラムを作成する。ただし、メソッドに関しては、カスタム関数として、プログラマーがインスタンスに定義することもできる。

クラスには、「ECMA」が定めた基本クラスの他にも、MovieClip クラスなど、様々なクラスが存在する。また、プログラマーがカスタムクラスを設計することもできる。

### 5-7 ActionScript のコーディングスタイル

「Macromedia」社は、現在では、「インスタンス」に「ActionScript」を書くことは推奨していない。推奨されているコーディングスタイルは、「メインタイムライン」の第1フレームにまとめて「フレームアクション」を書くというものである。このコーディングスタイルは、以下の3点を考慮したものである。

1. プログラマーとデザイナーの分業
2. 複数のプログラマーによる共同開発
3. Flash のコード実行のタイミング

スクリプトが「インスタンス」と「フレーム」に分散しては、上記の1. と2. は困難になる。上記の3. は、Flash の「タイムライン」の存在を考慮したものである。例えば、「タイムライン」の後方の「フレーム」で関数を定義した場合、その前方の「フレーム」からは、関数を呼び出せないことになる。上記の3点は、大規模なプログラムを書く際には重要である。しかし、「タイムライン」の制御は「ボタンインスタンス」か「フレームアクション」として書き、「ムービークリップ」の制御は、「ムービークリップインスタンス」に書くという、明快さは失われることになる。「アクション」および「ActionScript1.0」がデザイナーにある程度受け入れられたのは、従来のプログラミング言語にない、「画像に対して直接スクリプトを記述する」という特徴があったからである。

「Flash9」で採用される「ActionScript3.0」は、本格的なRIA 開発言語として設計されているため、「インスタンス」にスクリプトを書くことはできなくなる。なお、「ActionScript2.0」では、「アクション」及び「ActionScript1.0」のコーディングスタイルも許されている。

なお、「Flash MX2004」からは、スクリプト記述の際のノーマルモードがなくなったので、「FlashMX」で教育する場合も必ずエキスパートモードで行うべきである。

### 5-8 インタラクティブコンテンツ制作のためのメソッド

手続き型プログラミングでは、コンピュータに対する命令文を書くことが中心である。「ActionScript」では、「インスタンス」へのターゲットパスを指定して、「.」の後に、メソッドを書く。そのメソッドのうち、特に重要なものを以下に挙げる。以下のメソッドと、いくつかのイベントハンドラ、プロパティだけでも、ゲームやサウンドノベルなどのインタラクティ

ブコンテンツは作成できる。

- `stop()` で再生ヘッドを停止させ、`play()` で再生ヘッドを再び動かす
- `gotoAndPlay()` で再生ヘッドを指定したフレームへ移動して、続きを再生する
- `gotoAndStop()` で再生ヘッドを指定したフレームへ移動して、止める
- `nextFrame()`、`prevFrame()` で、次のフレーム、一つ前のフレームへ再生ヘッド移動して、止める
- `getURL()` で、リンク先をブラウザで表示する
- `trace()` で、変数の値を別ウィンドウに表示する（デバッグに有効）
- `duplicateMovieClip()` で、ムービークリップインスタンスを複製する
- `attachMovie()` で、ムービークリップインスタンスを生成する
- `startDrag()` で、ムービークリップインスタンスをドラッグ可能にし、`stopDrag()` でドラッグを停止する
- `hitTest()` で、ムービークリップインスタンスの衝突判定を行う
- `setMask()` で、マスクをムービークリップインスタンスに動的に設定する
- `createTextField()` で、テキストフィールドをムービークリップインスタンスに動的に作成する
- `setRGB()` で、ムービークリップインスタンスの色を変更する
- `swapDepths()` で、ムービークリップインスタンスの重なり順を変更する
- `loadMovie()` で、SWF ファイルをムービークリップインスタンスに読み込む
- `isDown()` で、キーボードが押されたことを判定する
- `random()` で、0 から 1 の範囲の乱数を発生させる
- `floor()` で、小数点以下を切り捨てる

## まとめ

映像処理や画像処理の分野では、趣味層やセミプロ層が厚かったが、アニメーションを個人で制作することは、ほとんどなかった。Flash が登場して、ベクトル画像によるアニメーション制作が個人で行えるようになった。「ファイルサイズが小さい」という Flash の特徴がネットワーク時代に適合し、Flash は Web アニメーション制作ソフトのスタンダードとなった。また、近年では、Web の広告バナーのほとんどは Flash で制作されている。「動いているモノ」を追いかけるといふ生物の習性からしても、Flash の動画が使われるという流れは必然のものであったといえる。しかし、アニメーションやバナーの制作ツールとしてだけ使用されてきたのではない。Flash は、初期のバージョンから、スクリプト言語を備えていた。近年は、スクリプト言語の能力が上がり、Web 上のゲーム、サウンドノベルから、アプリケーションのユーザインターフェースまで、Flash のスクリプト言語を使って制作されることが多くなっ

てきた。Flash は、このような多用途性ゆえ、教育内容も絞り込まなくてはならない。本論文では、Flash に関して、歴史、特徴、グラフィックス、アニメーション、プログラミングの5分野において、教育内容を提示した。グラフィックス、アニメーションを2年次で、プログラミングを3年次で教育するのが適切である。Illustrator を Flash の前に学ばせると、ベクトルグラフィックスに関して教えることは少なくなる。Flash 独特の特徴に関してだけ教えればいいので、効率がよいといえる。

**【参考 URL】**

- [ 1 ] [http://www.adobe.com/products/player\\_census/flashplayer/version\\_penetration.html](http://www.adobe.com/products/player_census/flashplayer/version_penetration.html)
- [ 2 ] <http://www.defide-ix.com/top.html>
- [ 3 ] <http://www.kaeruotoko.com>
- [ 4 ] <http://www.coj.co.jp/company/index.html>
- [ 5 ] <http://www.bb-anime.tv/bj>
- [ 6 ] <http://www.microsoft.com/japan/windows/windowsmedia>
- [ 7 ] <http://japan.real.com/realmusic>
- [ 8 ] <http://www.apple.com/jp/quicktime>
- [ 9 ] <http://www.adobe.com/jp/products/flashlite>
- [10] <http://www.adobe.com/jp/products/flex>
- [11] <http://www.ifour.co.jp/product/flashmaker2>
- [12] <http://www.swishzone.jp>
- [13] <http://www.geocities.jp/coa9999/parafla.html>
- [14] <http://www.dstorm.co.jp/products/swift3d>
- [15] <http://www.youtube.com>
- [16] <http://video.google.com>
- [17] <http://www.adobe.com/jp/products/acrobat/readstep2.html>
- [18] <http://www.java.com/ja/download/manual.jsp>
- [19] <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [20] <http://www.mozilla.org/js/language/es4/index.html>