

## Flash 3D Basics の日本語訳

本ドキュメントは、flashmagazine.com サイトで公開されている記事「Flash 3D Basics」を  
ヒム・カンパニー 永井勝則が自主的に日本語に訳したものです。

<http://www.himco.jp/>

[knagai@himco.jp](mailto:knagai@himco.jp)

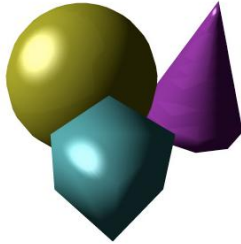
(2009/11)

本ドキュメントの原文は

[http://www.flashmagazine.com/Tutorials/detail/flash\\_3d\\_basics/](http://www.flashmagazine.com/Tutorials/detail/flash_3d_basics/)

です。

## Flash 3D の基本



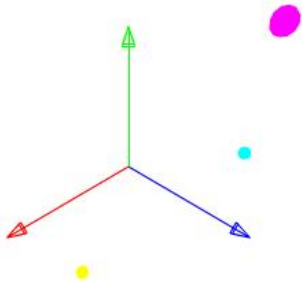
本記事は、これから新たに 3D コンピュータグラフィックスを始めようとするみなさんを対象に、その用語を紹介し、Flash で 3D を作成する際に必要になる事柄を解説します。このチュートリアルは、メッシュやテクスチャ、シェーダなどの用語をご存知ない方が、Papervision3D や Away3D、Sandy といった Flash 3D エンジンを取おうとするときに参考になります。本記事ではコンピュータの 3D に関する基本を扱うだけなので、プログラミングの知識は必要ありません。

チュートリアルは初心者向けですが、初心者以外の方にも参考になる事柄が含まれています。サンプルはすべて Away3D で作成され、そのソースコードはダウンロードできます。ソースコードを自分で調べることは最良の学習方法でもあるので、サンプルがどのように作成されているのか関心がある場合には、ソースコードをダウンロードし、解読してみてください。本記事の最後ではサンプルの使用方法を解説しています。またすべてのサンプルを動作させるには画面上をマウスオーバーする必要があります。このページのすべての Flash ファイルを一度に再生すると、ブラウザがクラッシュしてしまうからです。その理由もすぐに分かります。

では始めましょう。3D は単純です。3D はみなさんが自分自身の目で見、慣れているものです。そのためみなさんの脳は 3D の動作をよく理解しています。コンピュータにおける 3D も実世界の 3D とほとんど変わりません。違うのはただ、コンピュータ上では目にするものはすべて誰かによって作成されたもので、2次元で表示されるということだけです。

### 3次元

みなさんはおそらく、学校で習った数学のグラフを覚えておられるでしょう。線は X 軸と Y 軸を使って描きましたが、数学の先生を満足させるためだけのようでもありました。2D のグラフは X と Y 座標のセットを使い、その間を結ぶことで線を引きますが、3D のグラフでは X と Y のほかに Z 座標を使って線を引きます。



ムービー: Triaxe.as

このサンプルの画面上をクリックしドラッグしてみてください。ここでは三つ又(X、Y、Z 軸を表す矢印のついた線)と3つの球を追加しています。この球は三つ又が動く様子を理解しやすくするためのもので、これがないと実際に回転している感じが得られにくくなります。3D オブジェクトは移動させない(またはアニメーションさせない)限り、2D の描画とまったく変わらないものであることを覚えておいてください。

## 頂点

3D 空間の点は頂点と呼ばれます。頂点を定義するには、3つの各軸に関する位置を定義する必要があります。3D 空間の座標はつねに X、Y、Z の順に並べます。値は、2D の点と同様に、“原点”を参照して作成されます。Flash でグラフィックを描画するとき、そのグラフィックの位置は、ステージの左上隅を基準にした X と Y として与えられます。3D ワールドにも同じような原点があり、X、Y、Z 値はこの共通する参照を基準にします。

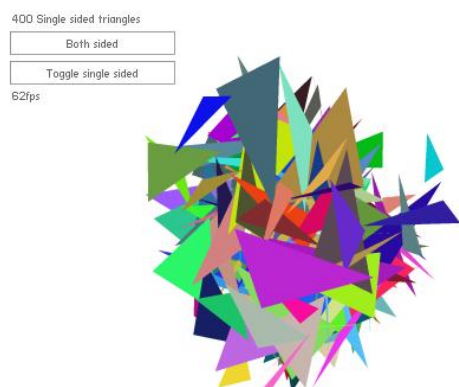
## 三角形

三角形は、描画可能な最小の 3D オブジェクトです。三角形を作成するには、3つの頂点が必要です。この3つの頂点の位置を変更すると、三角形を3軸に沿ったどの方向にも向けることができます。これにより三角形は非常に柔軟になります。三角形がほとんどの 3D ソフトウェアで基本的な固まりの作成に使用される理由はここにあります。

3D ソフトウェアでは通常、こういった三角形は“フェース”(サーフェスの略)と呼ばれます。この呼び方にはもう一つ、三角形は通常1方向しか向かない、つまり一方から見えるのは片側だけであるという、よい理由があります。ユーザーに表示されるのは、仮想のカメラの方を向いているフェースだけです。

ほとんどの 3D エンジンでは、可視的なサイド(側)をひっくり返したり、両サイドから見えるようにフェースを両面(ダブルサイド)にすることができます。フェースが両サイドから見えるように設定されている場合には、それ

を処理するためのパワーは少しだけよけいにかかります。これは大きな問題ではないように思えるかもしれませんが、三角形が多数必要になるときは顕在化します。

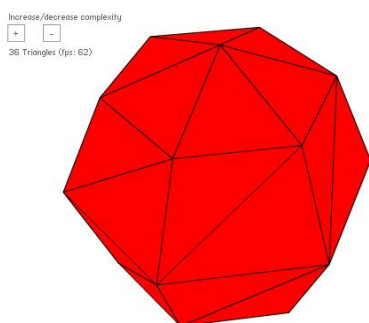


ムービー: Triangles.as

サンプルの上をマウスでなぞって再生させてみてください。ここでは、片面と両面を切り替えることのできる三角形を 500 個ランダムに作成しています。両面にするとフレームレートが落ちることに注目してください(フレームレートは fps(frames per second)で計測しています)。さらに複雑なことには、処理能力はコンピュータによって異なるので、Flash におけるリアルタイム 3D では、[異なるハードウェア](#)上での多くのテストが必要になります。

### 3D モデル

3D 空間に三角形を多数描画し、それらを互いに隣接して配置することで、モデルのようなものを作成することができます。結果として作成されるモデルは通常、メッシュまたはワイヤーフレームと呼ばれます。三角形では曲がったサーフェースは描画できませんが、三角形を多く追加し、先細に見えるサーフェースにすることで、曲がったように見せることができます。



ムービー: SphrerTri.as

+ ボタンをクリックすると、球の三角形の密度の高低を変える（三角形の数を増減する）ことができます。球の解像度を大きくすればするほど、球が三角形を元に見えているように見えなくなります。また三角形を追加していくにつれて、アニメーションがスローダウンすることにも気づかれるでしょう。レンダリングする三角形が多くなればなるほど、アニメーションの実行スピードは遅くなります。

この球は 3D エンジンでよく使用される多くのプリミティブの1つです。そのほかよく使用されるプリミティブには平面、ボックス、シリンダー、コーンなどがあります。3D モデルにはプリミティブを使用することもできますが、三角形を描画することから始めてゼロから作成することも可能です。しかしながらモデルの作成には専用のソフトウェアを使用した方がはるかに容易です。

### 3D モデルソフトウェア

3D は大きな産業で、さまざまな作業に適した多くのプログラムがあります。3D モデルを作成するソフトウェアには、Swift3D、Blender、Strata、Maya、3DS Max、Lightwave 3D、Softimage、Electric Image、Cinema4D、FormZ、Rhino、SketchUp、そのほかの多くのソフトウェアがあります。機能や価格に関しても多岐にわたり、単純なエフェクトの作成に使用するものからハリウッドで使用されるものまでさまざまです。

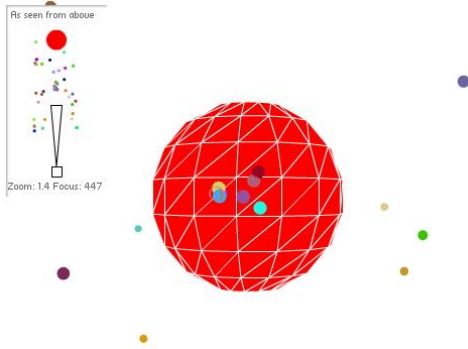
個人的には、わたしは 3DS Max (高価) と SketchUp (フリー) を使っていますが、Flash コミュニティではオープンソースツールの Blender が多く使われていることを知っています。Blender はフリーなので価格の面では比較になりませんが、3D エンジンにもとづいた ActionScript で操作できるさまざまな形式を書き出すことができます ([Dennis Ippel に感謝](#))。ただしどのツールを選択した場合でも、相応の時間をかけてそのツールを学ぶ必要があります。3D モデルの作成には面倒で退屈な作業がとれない、ソフトウェアの操作自体も複雑です。オンラインで見つけたチュートリアルを利用したり、すぐれた参考書を読んで理解を進めてください。そうするとモデリングの腕が上がっていくことになるのですが、そのときにはつねにモデルの複雑性にも目を向けるように留意してください。モデリングするソフトウェアでは立派に見えても、それは Flash によるリアルタイム表示には複雑すぎてしまう場合が多くあるのです。

### ビューとカメラ

画面上の 3D は実際には 3D ではありません。実際の 3D では2つの目で異なる事物を見る必要があり、そのためには特殊なハードウェアがない限り、容易なことではありません。これを解決するには、3D 空間を 2D に“翻訳”する必要があります。これは仮想的なカメラを使ってビューに連結させることで行います。ビューは 3D 空間の投影を作る矩形を言います。このビューの幅と高さを設定することで、結果として作成されるイメージの見え方が決まります。旧来のテレビのアスペクト比(幅と高さの比率)は 4:3 で、最新の HD

(高解像度)テレビのそれは 16:9 です。

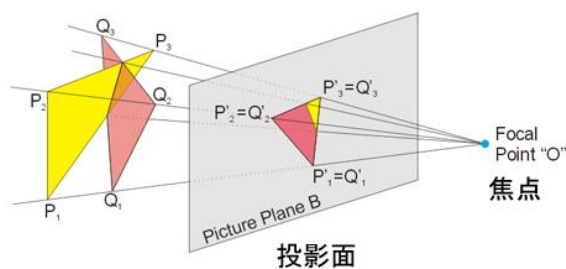
3D モデルをどのように描画するかはカメラの属性で決まります。次のサンプルで試してみてください。



ムービー: Camera.as

このサンプルのカメラ位置は固定されていますが、マウスの縦方向の移動でズームを変更し、横方向の移動でフォーカスの幅(視野、Field Of View、FOV と呼ばれます)を変えることができます。一般的なカメラレンズとまったく同じように、フォーカス値が低いほど(マウスが左にあるほど)広い領域が見え、高いほど見える領域は狭くなります。これは“*As seen from above*”(俯瞰図)で示しています。このサンプルは技術的に 100%正しいものではありませんが、その大まかな概念は得られるでしょう。マウスを左上隅に移動させていくことで、映画「スターウォーズ」に出てきたような“ハイパードライブ”(超光速航法)効果を見ることができます。右上隅に移していくと、3D モデルは平面的に見えます。下に移動させると大きな赤い球にズームインしていきます。

仮想的 3D シーンの 2D への変換処理は 3D 投影と呼ばれます。三角形は仮想カメラを通して見、平面上に投影されます。3D グラフィックスの投影には透視投影と正投影(直交射影)という2種があります。簡単にいうと、透視投影はカメラが“目”になりますが、正投影ではそうはなりません。三角形の位置の“翻訳”(置き換え)はカメラの設定によって決まります。

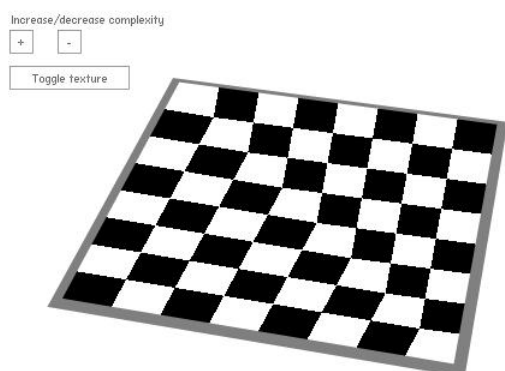


これは行列数学によって行われますが、本チュートリアルを超えてのトピックです。数学に長けもっと学びと思われる方は[ここをクリック](#)してください。とはいえ 3D エンジンを作ろうとでも思わない限り、平均的な Flash ユーザーにはこれは無縁でしょう。

## テクスチャ

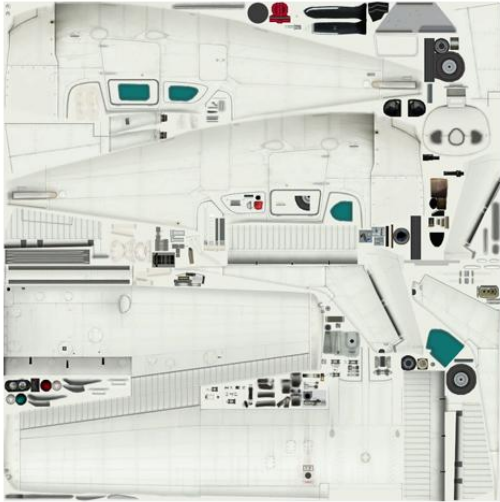
ワイヤーフレームのモデルは勉強にはなりますが、モデルにサーフェースにテクスチャを加えると、もっとそれらしく見えるようになります。テクスチャを追加する処理は、すべての三角形が大きなイメージの小さな一部を示す必要があるため、かなり複雑な作業です。それぞれの三角形は X、Y、Z 軸に沿って回転していることが考えられるので、テクスチャを傾斜させて透視を補う必要があるのです。

三角形の数が少なすぎると、テクスチャが歪んで見えます。この根本的な形状を改良するには三角形を追加します。するとテクスチャの見え目がよくなります。これはボタンをクリックすることで確認できます。



## ムービー: Textures.as

テクスチャは、UV 座標と呼ばれるルールにしたがって三角形に適用されます。この座標によって、イメージのどの部分がどの三角形に割り当てられるのかが決まります。UV 座標ではビットマップのさまざまな部分を指し示すことができるので、たった1つのビットマップを使うだけで飛行機全体の外観を簡単に作成できます。



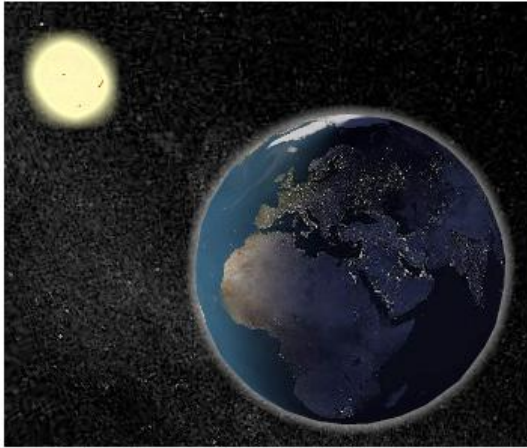
このイメージは UV 座標によって、異なる部分が飛行機の翼や本体、車輪、内装やそのほかの部分にマッピングされます。標準的なプリミティブは UV 座標のプリセットを持っていますが、飛行機のような複雑なモデル用の UV 座標を作成するには、3D モデリングソフトウェアを使用します。

3D モデルにマッピングできるテクスチャにはいくつかの種類があります。

- カラーマッピング: 単色または複数色でサーフェスを色づけします。
- ビットマップマッピング: モデルをビットマップでくみます。
- バンプマッピング: ビットマップマッピングに似ていますが、グレースケールイメージを使って、テクスチャのサーフェスに“バンプ(でこぼこ)”を追加します。複雑な計算を行わずにモデルを強化できます。(サンプル)
- 環境マッピング: ビットマップマッピングに似ていますが、オブジェクトの回転によって変化する別のビットマップにもとづく“反射”を作成します。(サンプル)
- ノーマルマッピング: モデルのサーフェスを滑らかにし、さらに高解像度に見えるようにします。バンプマッピング効果の作成に利用できます。(サンプル)

この分野に関して現行の 3D エンジンではその機能が異なるので、特殊な必要性がある場合には調べる必要があります。ざっと言うと、Papervision3D がサポートするのはカラー、ビットマップ、バンプ、環境で、Away3D がサポートするのはカラー、ビットマップ、環境、ノーマルです。

覚えておくべきことは、適切なテクスチャの使用がモデルをよいものにするということです。Web 上には高品質なテクスチャが数多く存在し、その気になれば本当に多くの素材が手に入ります。しかし予算が潤沢な場合には商用で提供しているベンダーから購入することもできます。



[Heaven and Earth チュートリアル](#)からのイメージ

## ライトとシャドウ (光と陰)

リアルさに関していうと、ライトとシャドウにかなうものはありません。われわれは実世界でこれに慣れているので、もしライトとシャドウがなかったら途端に気づいてしまいます。一般的にライトには、オムニライト(全方向を均等に照らす)、ディレクショナルライト(太陽光のように平行に照らす)、アンビエントライト(周辺光)などがあります。ライトとシャドウに関し、現在の Flash の 3D エンジンではこれを堅固にサポートしているものはありません。というもそのためには多くの処理能力が必要になるからです。これを行う小さなデモは存在しますが、いずれにせよ実験の範疇を超えるものではありません。

Flash Player 10 と新しい Pixel Bender シェーダエンジンは、Flash にライトとシャドウの搭載を可能にするものです。しかし今の時点ではこれらの機能はまだおまけに過ぎません。

## シェーダ

プレーンなテクスチャをつけた球は見た目もよく、多くの目的に利用できることは容易に想像できますが、見え方はあくまでも平面的です。シェーダは 3D モデルに奥行きと豊かさを加えます。シェーダには[フラット](#)、[グロウ](#)、[フオン](#)、バンプ、[セル/カートゥーン](#)などの種類があります。これらは、モデルの基本的なテクスチャにシャドウや詳細、ハイライト、反射、屈折などを追加します。Away3D でライトと、3D オブジェクトに影響を与えるシェーダとの組み合わせを見るには、Ralph Brooker による[このすぐれたサンプル](#)があります。

Papervision3D の貢献者の 1 人である Ralph Hauwert は、今後のリリースに組み込まれるかもしれない[シェーダに関する実験](#)を行っています。現時点では Flash で使用できるほとんどのシェーダは単純なフオンシェーダですが、Flash Player 10 の 3D 機能と Pixel Bender の機能が実装されると、これに変化が現れるかもしれません。

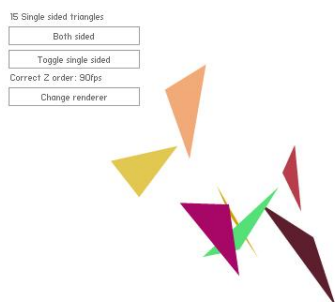
## レンダリング

レンダリングは、ここまで述べてきたすべてを結びつけ、投影イメージを出力する処理のことです。通常 3D はリアルタイムレンダリングとプリレンダリングに分けられます。Flash での使用に関していうと、通常はリアルタイムレンダリングが使用されますが、プリレンダリングも [可能です](#)。現行の 3D エンジンのタイプはすべてリアルタイムレンダリングです。

ほとんどの 3D ソフトウェアでは異なる種類のレンダラーが提供されています。光がガラスで屈折する高度なレンダリングを提供するものもありますが、リアルタイムレンダリングではそれは不可能です。リアルタイムレンダリングでは、すべてがトレードオフ(あっちを立てればこっちが立たず)で、再生を滑らかにするためのトリックが必要になります。よく使用されるのは、仮想カメラから見えない三角形はレンダリングしないというトリックです。この処理はカリングと呼ばれます。またつねに技術的に完全である必要がない場合には、すべての三角形を完全にソートしないことで“だます”トリックもあります。

多くのエンジンでは、状況によって異なるレンダラーが使用できる機能が提供されています。精度が必要で複雑な形状(計算)を使用しない場合には、非常に見栄えのよい結果を得ることができます。形状(計算)が複雑な場合には、最高速のレンダラーが使用できるようにモデルを作成するようにします。Away 3D には、Basic(高速)、Correct Z オーダー(三角形のソートを強制的に行う)、Intersecting Objects(正確に表示するため、オブジェクトをもっと多くの三角形に自動的に再分割(サブディバイド)する)という3つのレンダラーがあります。Papervision3D の GreatWhite ブランチでは、Basicと Lazy レンダラーが提供されています。Sandy 3D エンジンはその方法が異なり、4つの異なるソート方法が提供されています。これらのエンジンでは今後、さらにレンダラーが追加されていくと思われます。

レンダラーによってフレームレートに与える影響が異なることを例示するため、前の Triangles サンプルに戻りましょう。”renderer”ボタンをクリックしてレンダリング方法を変えて結果を見てください。



ムービー: [Renderers.as](http://Renderers.as)

結果から分かるように、Intersection Objects レンダラーが相当 CPU インテンシブです。交差するオブジェクトの数が多ければ多いほど、fps は下がります。3D レンダリングについてさらに学びたい場合には、[この Wikipedia ページ](#)が参考になります。

## Flash 3D エンジン

Flash で使用できる 3D エンジンについてはすでに触れました。以下は現在使用可能なエンジンの[リスト](#) (Carlos Pinho による)の一部です。

- Papervision3D (オープンソース)
- Sandy 3D (オープンソース)
- Away 3D (オープンソース)
- Wire Engine 3D (オープンソース)
- Electric 3D (商用)
- Alternative Platform (商用/非商用についてはフリー)

Carlos のリストではまたこれら以外のエンジンも取り上げられているので、たとえば物理エンジンを探したり、3D ゲームを作成する方法を調べたりしたい場合に利用できます。

上記リストの中ではおそらく初めの3つが最も興味深く、汎用的でしょう。Sandy は初めての Flash 用 3D エンジンで、ActionScript 2.0 にもとづいていたので動作はかなり鈍重でした。現在は ActionScript 3.0 に書き直され、大きな進展を見せています。Papervision3D についてはオンラインで数多くのチュートリアルが公開されているので、Flashmagazine としては Away 3D に焦点を当て、この偉大なエンジンを多くの人々の注目を集めるよう、一役買うことにします。Away 3D は当初から Papervision 3D をベースにしているのでよく似ていますが、Papervision3D の現バージョンでは不可能ないくつかの機能も備えています。Flash や Flex での Away 3D の使用方法を学びたい場合にはわれわれのチュートリアルを訪れてください。そのほかのエンジンについて知りたい場合には、[Nicolas Antille の記事](#)が参考になります。

Flash Player 10 は現在ベータ段階で、3D 操作の速度を上げるいくつかの機能や([サンプル](#))、グラフィックの表示スピードや正確性を大いに高める Pixel Shader の機能を備えています。とはいえUnityなど専用の 3D プラグインの品質には遠く及ばないので、驚くようなことは期待できないでしょう。

## サンプルファイルの実行

ダウンロードしたファイルの使用法を学ぶには、[Away 3D の設定チュートリアル](#)を参照してください。これらのサンプルを実行するには、Away 3D のソースコードと MinimalComps (サンプルファイルで使用してい

る UI コンポーネント)をダウンロードしてインストールしておく必要があります。MinimalComps は[キース・ピーターズ](#) (通称 Bit-101) が作成したコンパクトで簡単に使える UI コンポーネントのセットで、[minimalcomps](#) ページから最新版がダウンロードできます。ZIP ファイルを解凍したらプロジェクトフォルダにコピーします。また[Cover.as ファイル](#)も必要です。