

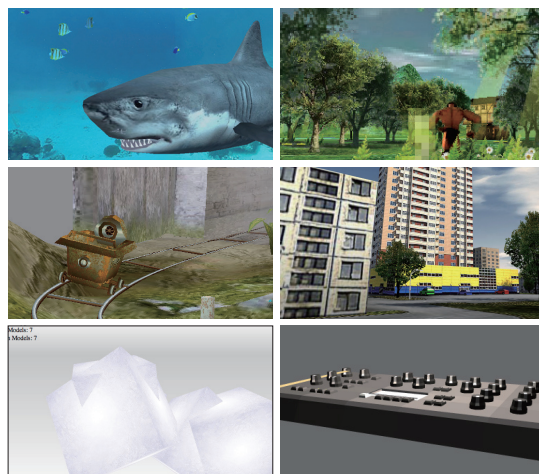


ActionScript ライブラリではじめる Flash 3D

SERIES #09

ライブラリで手軽に 3D 表現 : 概論

数年前であれば、Web上でリアルタイムな3D表現を行うためには、「Shockwave3D」など、専用のランタイム環境が必要でした。しかし、「Papervision3D」や「Away3D」などのFlash用3Dライブラリの登場によって、Flashでもリアルタイムな3D表現が可能となり、ここ1〜2年の間にFlash3Dコンテンツが爆発的に増えました。3Dライブラリは新しい表現ができるということもあって、ActionScriptライブラリの中でもっとも人気の高い分野の一つとなっています。本連載でもいよいよ、3Dライブラリについて取り上げていきましょう。1回目となる今回は、3Dライブラリを利用する上で押さえておくべき基礎知識について解説します。



複雑な処理はライブラリが引き受けてくれる

Flash Player 9 と ActionScript 3.0 の登場で処理速度が劇的に高速化したことを背景に、「3D エンジン」と呼ばれるライブラリがいくつも開発されました。そして、その多くがオープンソースで公開されたこともあって、3D表現を多用したFlashコンテンツが国内でも多く見られるようになりました。ゲームなどのエンターテインメント分野ではすっかり定着した表現ですが、Webブラウザの中で、しかもFlashコンテンツとして動くというのはインパクトがありますよね。

さらに注目なのは、それらが3Dアプリケーションなどで作られた「予定されたアニメーション」ではなく、「ユーザーがマウスやキーボードを操作して、物体をいろいろな角度から眺めたり、空間内を自由に飛び回ったりできる、リアルタイムかつインタラクティブな要素を持った3Dコンテンツである」という点です。

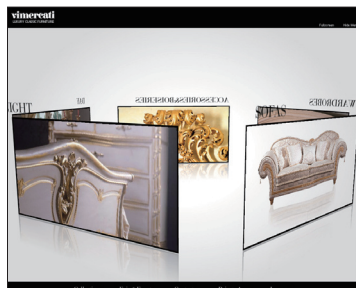
これまででは、多様な知識と技術力を持たない限り手が出せなかった領域ですが、ライブラリの力を得ることで、基本的な記述方法さえマスターすれば、ある程度のことは誰にでも手軽に実現できるようになりました。3D描画に必要な演算、画像処理、データ管理などの機能を複合的に連動させるというややこしい部分は、すべて3Dライブラリが引き受けてくれます。「机の上に幅30cm、高さ15cm、奥行き25cmの箱が置いてあって、それを斜め上から俯瞰で見た感じで。箱の色はオレンジがいいなあ。あとさー、魚眼レンズで見た感じとかになると面白いかなあー」といったアイデアを、ササッと絵にしてくれる頼もしい相棒です。

もちろん、漫然と使っているだけではなく、ライブラリを利用することを通じて数学的な概念への理解を深めていくことや、3D表現の効果的な使い

どころを見極めるセンスを磨くことも重要です。とはいえ、まずは無邪気に3Dライブラリで遊んでみてはいかがでしょうか。3Dライブラリと上手に付き合えるようになるための基礎知識を身につけて、表現の幅を広げる3Dプログラミングへの第一歩を踏み出しましょう！



HelloEnjoy

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

Vimercati Meda

<http://www.vimercatimeda.it/>

3Dライブラリは、ゲームコンテンツだけでなく、タイポグラフィやサイトのナビゲーションなど、さまざまな用途で使われています



代表的な 3D ライブラリ

3D ライブラリは、ActionScript 3.0 の登場でその数が大幅に増え、多様な選択肢が生まれました。その大半はオープンソースですが、なかにはパッケージ製品も存在します。ここでは、代表的な 3D ライブラリを紹介していきます。

オープンソースで配布されているライブラリは、ライセンスによってその利用条件が規定されています。たとえば、MIT-License の Papervision3D

と Apache License 2.0 の Away3D、そして特に利用制限をしていない Sandy については、成果物を Web コンテンツとして公開してもソースコードの開示は強制されず、比較的自由に使えます。しかし、GPL の WireEngine3D に関しては商用、非商用に関わらずソースコードを開示する義務が発生します。各ライブラリを利用する際には、それぞれのライセンスに注意しましょう。



Papervision3D

ActionScript 3.0 による 3D 表現の可能性を世に知らしめた人気ライブラリ。いろいろな人が Web 上でサンプルコードなどを公開しているので、情報量が多く、簡単な使い方ならすぐに習得できるのが魅力です。



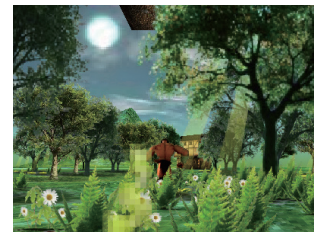
Papervision3D official demo
<http://www.papervision3d.org/>

ライセンス MIT-Licence
 公式サイト <http://blog.papervision3d.org/>



Sandy 3DENGINE

ActionScript 2.0 版から脈々と開発が続く、ActionScript 向け 3D ライブラリの老舗的存在。Flash という制約の中で 3D ライブラリに求められる一通りの機能が実装されています。



Protopop walk demo
http://www.flashsandy.org/demos/protopop_walk/

ライセンス 非商用、商用問わず基本的に無償で利用可 (ドネーションウェア)
 公式サイト <http://www.flashsandy.org/>



Away3D

Papervision3D から派生し、現在は完全独自の 3D エンジンとして開発が続けられています。頂点やポリゴン制御、ベクトル操作など、気の利いた機能が用意されています。



RailAway Express
<http://www.closier.nl/playground/railaway-express.html>

ライセンス Apache Licence2.0
 公式サイト <http://away3d.com/>



Altanativa3D

ゲーム制作を前提として作られた、高速度かつ多機能なライブラリです。パッケージ製品だけあって非常に完成度が高く、Flash での 3D 表現における一つのベンチマークとなる存在になっています。



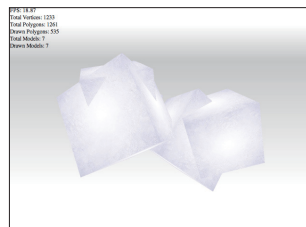
ALTCITY
<http://alternativaplatform.com/en/demos/altcity/>

ライセンス 非商用なら無償で利用可 (商用の場合は有償)
 公式サイト <http://alternativaplatform.com/en/>



Wick3D

今回紹介した中では最も新しい 3D ライブラリ。ほとんど情報がないのですが、後発だけにかなり洗練された印象で、特にシェーディング (陰影を付ける処理) の美しさと高速な処理が際立っています。



BSP sorting demo
<http://www.derschmale.com/2008/10/04/wick3d-update-bsp-sorting/>

ライセンス LGPL
 公式サイト <http://code.google.com/p/wick3d/>



WireEngine3D

ライセンスがネックなのか国内ではあまり普及していないようですが、これも高い性能を持ったライブラリです。今回紹介したなかで唯一 LWO (LightWave3D NewTek 社の 3D アプリケーションの形状データ) の読み込みに対応しています。



Loading a LWO file with a Virus Synthesizer
<http://3key.at/we3d/forum/demos/virus/>

ライセンス GPL
 公式サイト <http://www.3key.at/we3d/forum/>

基本知識：3D ライブラリが提供する主な機能

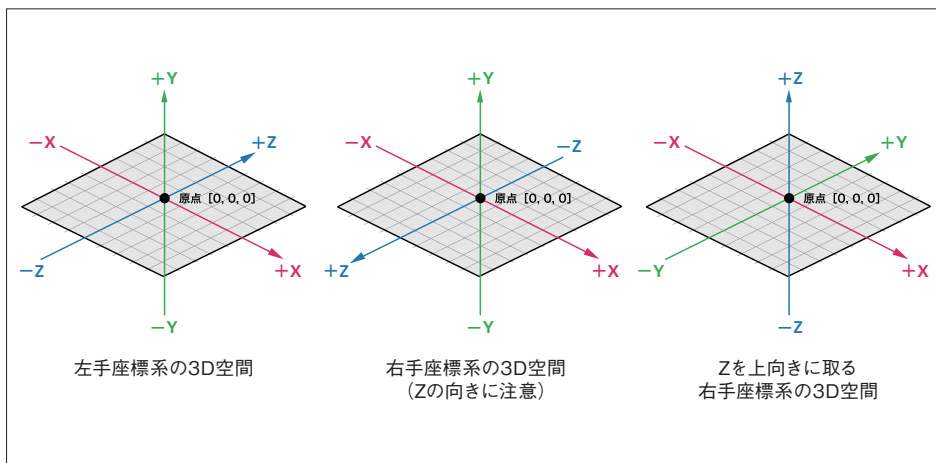
3D ライブラリが提供する基本的な機能から解説します。ほとんどの機能はライブラリ自身が必要に応じて使用するものなので、これらをすべて理

解しないとコードが書けないということはありませんが、基本的な概念を理解するために一つひとつの機能について把握しておきましょう。

3D 空間の提供

3D を表現するには横軸と縦軸に加えて、「奥行き」を表す軸が必要です。基本的にFlashには奥行きがなく (Flash CS4 の 3D 機能については次ページ参照)、それを補うために 3D 処理専用の空間 (シーンやワールドと呼びます) を生成する機能が用意されています。3D 空間は、原点で交わる「X (横軸)」「Y (縦軸)」「Z (奥行き)」の 3 つの軸によって定義されます。各座標の向きの決め方には、大きく「左手座標系 (Z を増加させると奥に向かう)」と「右手座標系 (Z を増加させると手前に近づく)」があり、ライブラリごとにいずれかを採用しています。

AS 用 3D ライブラリでは、左手座標系を採用しているものが多いようです (Papervision 3D は右手座標系に設定することもできます)。はじめはピンと来ないかもしれませんが、覚えておくと後々役立ちます。



空間内部での「視点」となるカメラの提供

ライブラリが提供する 3D 空間の中の様子は、画面上に直接表示することができません。このため空間内部の様子を見るためのカメラ (視点) を設置する機能が用意されています。カメラは空間内を自由に移動したり、好きな方向に向けたりすることができます。カメラに写らない範囲は画面に表示されないことに注意してください。

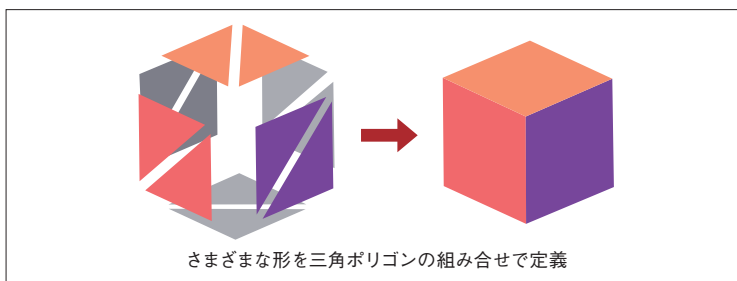
座標変換と、変換に必要な演算

3D 空間内に配置される形状の正体は、それを形づくる頂点の座標情報の集合体です。これを空間上の位置、カメラから見た位置、最終的に画面に表示する際の位置を次々に変換していくのが「座標変換機能」で、地味な存在ですが 3D ライブラリの中核を担う重要な機能です。

ポリゴンの描画

形を定義する頂点座標を元に、各頂点を繋いで「面」を描画する機能です。面が描画されて初めて画面にその姿が映るわけですが、この面のことを「ポリゴン」と呼びます。AS 用 3D ライブラリでは、座標変換された頂点座標を元に、`beginFill()` や `lineTo()` などを用いてベクターシェイプとして描画します。この「3D 座標で演算 → 座標変換 → 2D のベクターシェイプ化」が、3D ライブラリの仕事の大半を占める処理です。

AS 用 3D ライブラリのポリゴンは、頂点を 3 つ繋いでできる三角ポリゴンをもっとも基本とし、すべての形を三角形の集合体で表現します。球体などの曲面を表現する場合は、より多くのポリゴンが必要になります。



マテリアル・テクスチャマッピング

その形状の表面の様子を視覚的に表現するための機能です。形状の表面 (ポリゴン) の表示方法を定める機能を「マテリアル」、ポリゴンの表面に画像を貼り付けることを「テクスチャマッピング」と呼びます。マテリアルにはいくつかの種類があり、3D ライブラリによっては MovieClip や flv をポリゴンの表面に貼り付けることができます。

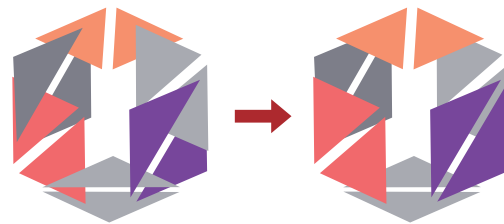
左から、ワイヤーフレーム+カラーマテリアル、MovieClip、FLV を貼り付けたポリゴン



Zソート (陰面消去)

前述したようにAS用3Dライブラリのポリゴンはベクターシェイプ、大ざっぱに言うならば三角形が描かれたMovieClipですので、オブジェクトの形状に合わせてこれらの前後関係を正しく指定する必要があります。なんだか難しそうですが、これもライブラリが処理してくれます。

AS用3Dライブラリでは、ポリゴンを構成する3つの頂点のZ座標の平均値を割り出し、より奥に配置されるものから順番に描画していく「Zソート」という方法が採用されています。また、前後関係があるということは、物体の陰に隠れて見えない部分も当然でできます。この部分を描画しても無駄になるだけなので、見えない部分は描かないことで最適化を図っています。



ポリゴンの前後関係を整え、適切な見栄えになるように処理

クリッピング

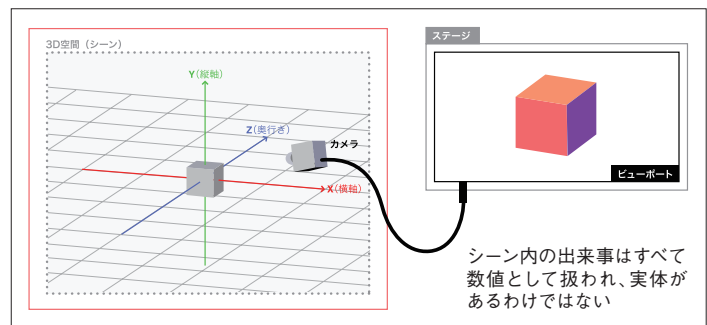
カメラに映らない範囲 (たとえばカメラの裏側とか) は画面に表示されないと言いましたが、表示されていないというだけで実際はさまざまな処理が行われています。これを意図的に計算の対象外にする

のが「クリッピング」と呼ばれる機能です。画面に表示されるポリゴンが増えるほど負荷がかかるので、無駄を減らすための工夫が施されています。

3D ライブラリの動作イメージ

3D ライブラリが実際に動作している際の状況は、右のようなイメージです。どの3Dライブラリを使った場合でも概ね共通しています。3つの座標軸 (X, Y, Z) で構成される3D空間 (以下「シーン」) 内に、画面に表示したい形状 (図の例だと立方体) を配置し、それを同じくシーン内に配置されたカメラによって撮影、その状況をFlashステージの表示リストに追加された「ビューポート」(カメラに接続された外部モニターのような役割を持つもの) に投影することで、3D表示を実現しています。このうち、「空間内をカメラで撮影 → ビューポートに投影」の処理を「レンダリング」と呼び、さまざまな演算が行われます (前述したポリゴン描画、座標変換、Zソート、クリッピングなど)。

レンダリングを繰り返し実行してシーン内の変化を連続して見せることで、形状をアニメーションさせたり、カメラワークを付けたりといったリアルタイム3D表示を実現しています。



動作イメージのポイント

- シーン自身とシーン内に配置されたものは、すべて概念に基づく数値でしかなく、直接見ることができない。
- 目にすることができるのは、カメラを通じてレンダリングされたもの。
- レンダリングを連続で行うことでリアルタイム表示ができる。

まとめ

冒頭で書いた通り、3Dライブラリとは、「こちらの指示に従って立体的な絵を描いてくれる相棒」です。コンテンツを制作する際のライブラリへの指示は、すべてスクリプトで以下の順に記述して伝えることになります。

- シーンとカメラの定義
- ビューポートの定義
- 形状の定義
- レンダリング処理

もっとシンプルに、「初期設定して、形の見栄えと振るまいを指示すれば3D表現ができる」と考えるといいでしょう。次回はPapervision3Dを使って、実際にコードを書きながら簡単なデモを作成し、3Dライブラリを扱う具体的な作法について解説します。

Flash CS4 / Flash Player 10 の 3D 機能とライブラリの関係

column

Flash CS4 の新機能として搭載された3D機能によって、Z座標指定やXYZ軸による回転など、平面 (2D) であるMovieClipを3D変形・アニメーションできるようになりました。しかし、これだけでなく3Dライブラリを使って作られたコンテンツと同様のものが簡単に作れるようになったかという点、そうではありません。3Dアプリケーションのようなモデリング機能が備わったわけではないので、ライブラリを使わなかった場合、立方体一つ作るだけでも一筋縄ではいかないのです。

Flash CS4 が提供する3D関連機能は、あくまでも部品に過ぎません。そのため、各機能をまとめあげた処理体系を構築する必要性が生じます。汎用性が高く効率的な処理体系を作り上げるためには、3Dライブラリを自作するほどの労力がかかります。ですから、3DライブラリはFlash CS4の登場によってその必要性が失われるのではなく、Flash CS4の3D機能をより高度に扱うための手段として、むしろ今後も進化していくでしょう。

なお、P116からの「Flash Lab.」連載では、Flash CS4の3D機能について詳しく解説しています。