

XNA Game Studio ゲームプログラミング

2D編 - 第1回 スプライト

スプライト

- ・XNAでは、2Dグラフィクスの描画はすべてスプライトで行う
- ・本来のスプライトはキャラクタをハードウェアで高速に表示する機能
- ・XNAでは「ポリゴン」+「テクスチャ」で擬似的に再現
- ・3Dの機能を用いるので「拡大縮小」「回転」「半透明」「zバッファによる重なり」をサポート

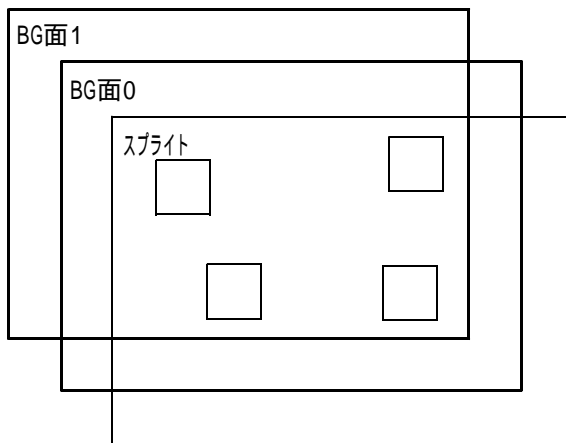
概要

スプライトとは、キャラクタを高速に表示するための機能のことで、背景などほかの画像を壊すことなく独立して制御でき、透過色や重なり順序を指定することができます。

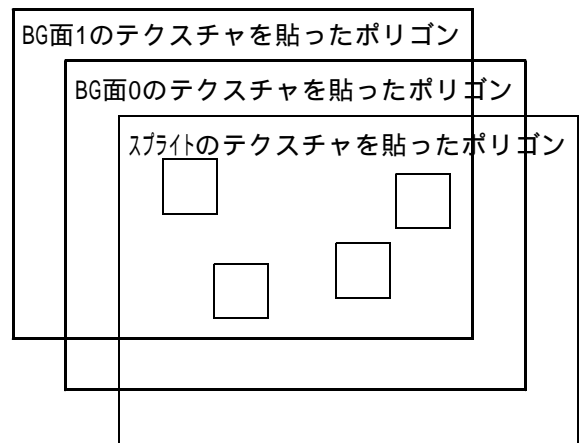
本来のスプライトは、ハードウェアで実現される機能で、背景画像(BG面)とは別に、多数の小さな画像を用意しておき、画面に合成して表示するものです。複数の背景やスプライトと重ね合わせて合成表示ができるようになっており、ほかのスプライトと重なったときに、どれを前面に表示するかが指定できるようになっています。CPUに負担をかけずにキャラクタを表示することができます。

現在では、CPUが高速化され、グラフィック機能も3Dに傾斜しているため、ハードウェアでスプライトをサポートしている機種はほとんどなく、ポリゴンとテクスチャを用いたり、SDKやライブラリでスプライトを擬似的に扱えるようにしています。

XNAでも、2Dグラフィクスの機能(スーパーファミコンのような旧来のスプライトおよびBG機能)は実装されていないため、すべての2D表示は、ポリゴンを平面的に並べ、そこにテクスチャを貼り付けて擬似的に再現する疑似スプライトとなります。3Dの機能を用いて描画(レンダリング)するので「スケールング」「回転」「アルファブレンディング」といった機能を使用することができます。



スーパーファミコンやセガサターンはBG面とスプライトをハードウェアで合成して画面に表示



XNAではすべて「ポリゴン」+「テクスチャ」で表示重なり順序は描画する順序やzバッファで指定

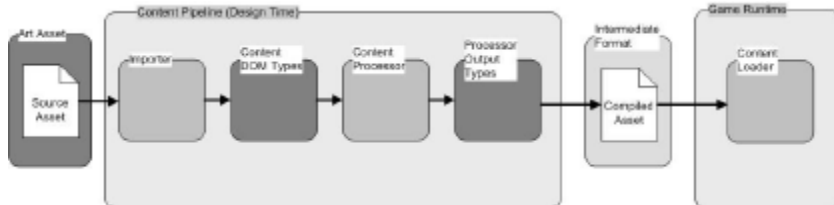
スプライトの読み込み

- ・スプライト(2Dグラフィクス)の読み込みはコンテンツパイプラインを用いると便利
- ・コンテンツパイプラインは、画像やサウンドなどのリソースを簡潔に効率よく扱うための仕組み
- ・コンテンツパイプラインを用いてリソースを読み込む方法は以下のとおり
 1. ファイルをプロジェクトの"Content"へ追加する
 2. ContentManagerクラスのLoadメソッドで読み込む

概要

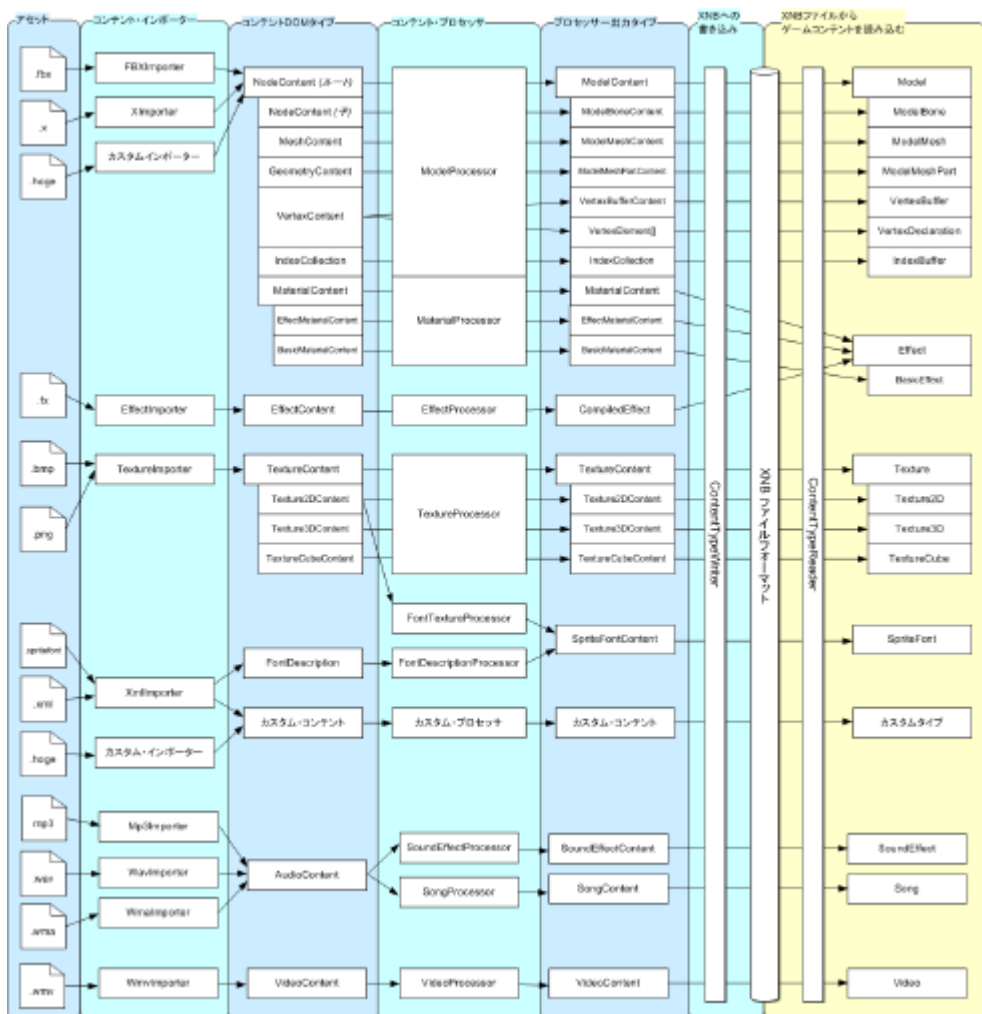
XNAには2Dグラフィクス、3Dモデル、サウンドといったリソースを簡単、効率よく扱うためにコンテンツ・パイプラインという仕組みがあります。コンテンツにリソースを登録すると、ビルド時にそれらがコンテンツ・パイプラインによって処理され、WindowsやXbox360で効率良く使えるデータ形式(.xnb)に変換されるようになっています。また、独自形式のフォーマットを扱えるようにカスタマイズすることもできます。

コンテンツの読み込みはContentManagerクラスのLoadメソッドで行います。引数はアセット名(コンテンツに登録したリソースの名前)で、プロジェクトのフォルダまたはContentManagerのインスタンスを生成したときに指定したフォルダが基点となります。



コンテンツに登録したリソースの流れ。コンテンツ・パイプラインによってビルド時に"Compiled Asset"(XNB形式)に変換。さらに、実行時にXNA Frameworkの"Content Loader"により、各機種が扱いやすい内部形式に変換される。

データ変換過程



サポート形式

2D画像 : .bmp, .png, .jpg, .tga, .dds, .dib, .hdr, .pfm, .ppm
モデル : .x, .fbx
マテリアル : .fx
サウンド : .xap, 3.0以降[wav, .mp3, .wma]
動画 : 3.1以降[.wmv]

コーディング例

```
// フィールド変数へ追加
Texture2D plyTxtr, bgTxtr; // 画像を管理
    ⋮

public GameMain()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content"; // コンテンツの初期フォルダの設定
    ⋮

protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    // TODO: use this.Content to load your game content here
    plyTxtr = Content.Load<Texture2D>("player1"); // プレイヤー 1 画像読み込み
    bgTxtr = Content.Load<Texture2D>(@"bg/bg1"); // BG 1 画像読み込み
    ⋮
}
```

課題

画像を読み込んでみましょう。

(1) キャラクタの画像を準備してください。

(2) 画像ファイルをコンテンツに登録します。

1. ソリューションエクスプローラの "Content" で右クリックし、「追加(D) 既存の項目(G)...」と選んでください。
2. 「既存項目の追加」ダイアログで追加したいファイルを選択します。
3. 「追加(A)」ボタンをクリックすると、選択したファイルがコンテンツとして追加されます。(エクスプローラからファイルをまとめて選び、「Content」へドラッグ&ドロップする方法でも可能)

(3) コンテンツを読み込むプログラムを追加します。以下のプログラムを適切な場所に追加しましょう。

```
1. Texture2D chrTxtr; // キャラクタ画像を管理

2. // キャラクター画像読み込み
   chrTxtr = Content.Load<Texture2D>("アセット名");
```

(4) ソリューションをビルドし、実行してみましょう。エラーや例外が発生しなければ成功です。

(5) キャラクタのほかに、背景用の画像を準備し、読み込むプログラムを作成しましょう。