CHAPTER 2

DirectXを使ったゲームプログラミング

DirectX Graphicsで2次元オブジェクトを描画してみよう

>>> 大澤 文孝 OSAWA, Fumitaka

はじめに

ゲームと言えば、Visual C++を使ってネイティブなコードで書いていくというイメージがあるかもしれません。

しかしDirectXの最新版となるDirectX 9.0には、.NET Framework環境からDirectXを操作できる「ManagedDirectX」が提供されており、Visual Basic .NETやC#などからもゲームを作成することができます。

DirectXは、.NET FrameworkのGDI＋を使って描画するのとは違い、画面への描画方法がやや特殊です。

本稿では、DirectXで2次元オブジェクトを描画する方法を例にとり、.NET FrameworkでDirectXを使うための基本を説明していきます。

DirectXに含まれるもの

DirectXは、ゲームやマルチメディア処理に必要となるAPIの集合体です。大きく次のコンポーネントにわけることができます。

DirectX Graphics

3Dグラフィックや2Dグラフィックを扱うAPIです。

開発者は、このAPIを使うことによって、ビデオカードに内蔵されている演算処理部を使って、高速な3D、2D処理ができます。

DirectDraw

2Dのグラフィックを操作するAPIです。

DirectDrawは、DirectX7からDirectX8になったときに、事実上「DirectX Graphics」に吸収され、いまでは使われなくなっています。

つまり言い換えれば、現在では、2DでもDirectX Graphicsを使うことになっています。

ただし、いまでもDirectX7でも動かせるプログラムを作りたい場合や、3D機能は不要なのでリソース量を削減したいといった場合などに使われることもあります。

DirectInput

マウス、キーボード、ジョイステックなど、さまざまな入力デバイスを制

レベル >>> Level

1 2 3 4 5

言語 >>> Language

- C#

ツール >>> Tool

- Visual C# 2005 Express
- DirectX 9.0 SDK

サンプル >>> Sample

この記事で取り上げたソースコードおよびサンプルプログラムは、
http://www.shoeisha.com/mag/windev/
からダウンロード可能です。

御するAPIです。

DirectPlay

通信機能を提供するAPIです。このAPIを使うと、TCP/IPやNetBEUIなどの通信プロトコルを用いて、ゲームで使われるデータを送受信することができます。ピアツーピアとクライアント／サーバーの両方の通信方式に対応しています。

またDirectPlay Voiceを使うと、音声通信もできます。

DirectSound、DirectMusic

音を鳴らすためのAPIです。3Dサウンドにも対応し、音源位置を指定してドップラー効果を出すことなどもできます。

DirectShow

マルチメディアを扱うためのAPIです。動画の再生時にフィルタをかけたり、再生中の動画フレームをキャプチャして抜き出したりする操作ができます。

本稿では、これらのAPIのうち、DirectX Graphicsの2Dに関する機能を説明していきます[注1]。

もちろん、2Dグラフィックを扱うときには、DirectX Graphicsを使わずに、.NET FrameworkのGDI＋を使っても同等な処理ができます。

しかしGDI＋では、拡大や縮小、回転、重ね合わせなどは、CPUで処理することになりますから、DirectX Graphicsを使った場合よりも低速になります。

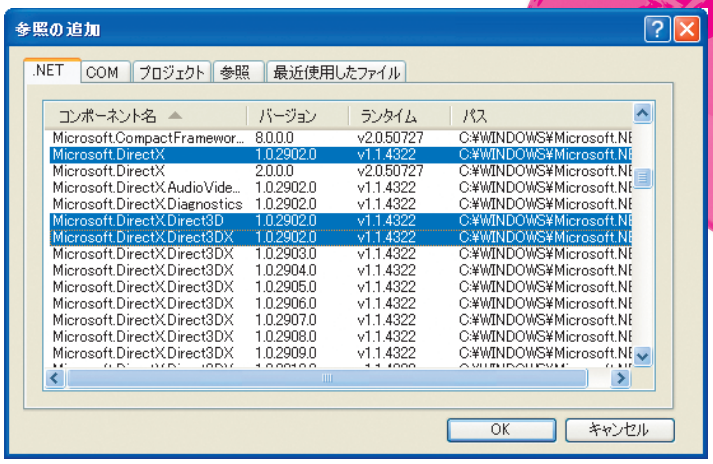
反面、DirectX Graphicsを使う場合には、

・DirectXがインストールされていることが必要
・実行時にハードウェアを制御するためのセキュリティ権限が必要

という欠点があります。

そのため用途によって、DirectX GraphicsにするかGDI＋にするかを使い分けるべきです。

図1：必要な参照設定



必要なソフトとプロジェクトの設定

DirectXの開発をするには、「DirectX 9.0 SDK」が必要です。DirectX 9.0 SDKは、「http://www.microsoft.com/japan/msdn/directx/」からダウンロードできます。

DirectX 9.0 SDKには、.NET Frameworkから扱うための「Managed DirectX」も含まれています[注2]。

本稿の執筆時点では、「DirectX SDK（June 2006）」が最新版なので、これをダウンロードしてインストールします。

プロジェクトの作成と参照設定

DirectXの開発をするときには、ふつうのWindowsアプリケーションとしてプロジェクトを作成します。DirectXを使うからといってプロジェクトに対して何か特別な設定を必要とするわけではありません[注3]。

DirectXの開発をするときに必要なのは参照設定です。本稿で必要となる参照設定は、次の3つです（図1）[注4]。

Microsoft.DirectX

DirectX全般に関するものです。

Microsoft.DirectX.Direct3D

DirectX Graphicsに関するものです。

Microsoft.DirectX.Direct3DX

後述するスプライト機能など、DirectX Graphicsの付加的なクラスが格納されています。

これらのコンポーネントに含まれる名前空間は、次の通りです。

```
using Microsoft.DirectX;
using Microsoft.DirectX.Direct3D;
```

注1）DirectX Graphicsの扱いは、2Dも3Dもあまり変わりません。言い換えると2Dは、「Z軸がないもの」として扱われた特殊な3D描画であるとも言えます。そのため本稿の内容の一部は、3Dの場合にもあてはまります。

注2）すでに古いバージョンのDirectX SDKがインストールされているときには、アンインストールしてからインストールしてください。なおDirectX自身（ランタイム）は、一度インストールすると、それを削除することはできません。

注3）もし詳細なデバッグをしたいのなら、プロジェクトのプロパティの［デバッグ］で、［アンマネージコードデバッグを有効にする］にチェックを付けておくとよいでしょう。

注4）環境によっては、異なるバージョンのDirectXがインストールされていることがあります。その場合、これら3つのバージョンを同じものにしないと実行時にエラーが出ることがあります。