

## はじめに

次世代WindowsであるVistaでも3次元CGを用いたインターフェイスが採用されるなど、3次元CGはゲーム以外にも広がりつつあります。そのような3次元CGのアプリケーションは、WindowsではDirectXのモジュールのひとつであるDirect3Dを使って開発します。

ここでは、C#からも簡単に3次元CGのアプリケーション開発ができる「Managed DirectX」を使ったDirect3Dプログラミングの基本テクニックを紹介してゆきます。

Xは、もともとC++言語のようなアンマネージド環境用に開発されているため、そのままではマネージドコードを生成するC#からは使うことができません。

Managed DirectXは、C#(もしくはVB)からDirectXにアクセスするための機能のことで、.NET Frameworkのネームスペース内にクラスをまとめているため、C++よりも簡単にDirectXプログラムを書くことができます。しかも、C++と比べてパフォーマンスにおいて見劣りすることもほとんどありません。

このManaged DirectXを使うには、最新の「DirectX 9.0 SDK」をインストールする必要があります。マイクロソフトの「DirectX Developer Center」

(<http://www.microsoft.com/japan/msdn/directx/>) からSDKのインストールキットをダウンロードし、開発マシンに導入してください。

SDKには、C#用のサンプルもたくさんついてくるので、まずは、これらサンプルをいろいろと実行してみるとよいでしょう。

## Direct3Dのプログラミングフロー

### 基本プログラムとなるスケルトン作成

プログラミングを始めるにあたって、

## Managed DirectXとは

C#からDirectXを使うには、Managed DirectXという機能を使います。Direct

レベル >>> Level

1 2 3 4 5

ツール >>> Tool

- Visual C# 2005 Express
- DirectX 9.0 SDK

言語 >>> Language

- C#

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。

CHAPTER

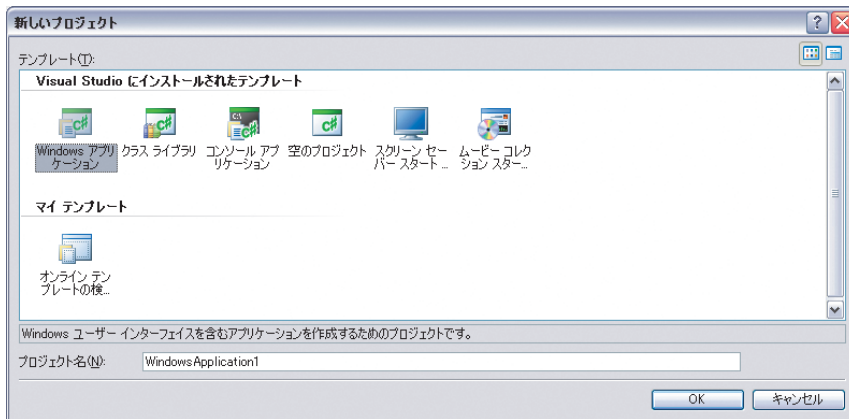
3

# 立体感のある表示に挑戦!

3Dゲームプログラミングの基本を学ぶ >>> N2Factory



図1：Windowsアプリケーションのプロジェクトを生成



リスト1：Direct3Dプログラムの基本ソースコード

```

using System;
using System.Drawing;
using System.ComponentModel;
using System.Windows.Forms;
using Microsoft.DirectX;
using Microsoft.DirectX.Direct3D;

namespace D3DProject01 {
    public partial class Form1 : Form
    public Form1() {}

    // デバイスの生成
    public bool CreateDevice() {
        return true;
    }

    // 図形の作成
    private void CreateModel() {}

    // カメラの視野角設定
    private void SetProjection() {}

    // カメラの位置設定
    private void SetCamera() {}

    // 図形の位置設定
    private void SetLocation() {}

    // 図形の描画
    private void Draw() {}

    // メイン関数
    static void Main() {
        using (Form1 frm = new Form1()) {
            if (!frm.CreateDevice()) {
                MessageBox.Show("Error :
                    CreateDevice");

                return;
            }

            // Form1の表示開始
            frm.Show();
            frm.CreateModel();

            // 描画処理ループ
            while (frm.Created) {
                frm.SetProjection();
                frm.SetCamera();
                frm.SetLocation();
                frm.Draw();
                Application.DoEvents();
            }
        }
    }
}

```

まず基本となるプロジェクトを作成します。

Visual C# 2005 Expressを起動し、メニューから「ファイル」-「新しいプロジェクト」を選択します。ダイアログボックスから「Windowsアプリケーション」テンプレートを選択して、プロジェクトを生成しましょう（図1）。

次に、生成されたプロジェクトに、参照設定で「Microsoft.DirectX」と「Microsoft.DirectX.Direct3D」を追加しておきます。また、今回のプログラムに必要な項目は削除しておきます。

最終的にプロジェクトの「参照設定」が図2のような状態になっていればOKです。

図2：整理後のプロジェクトのようす

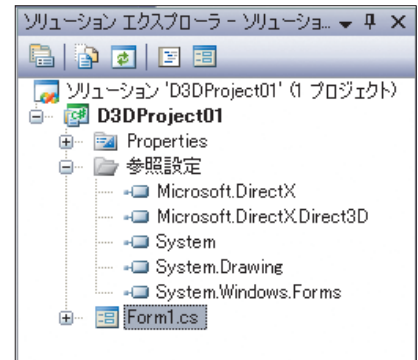
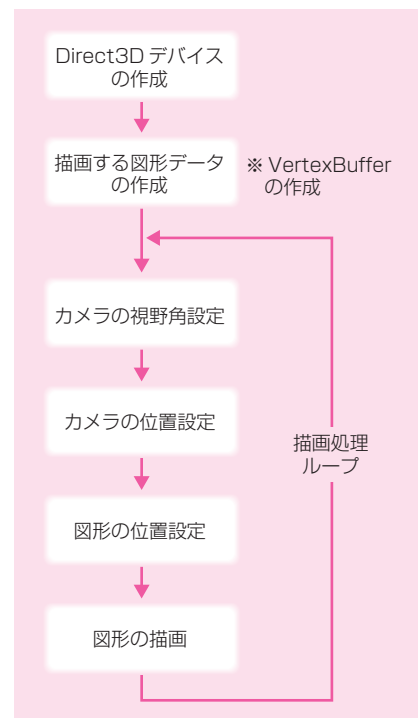


図3：Direct3Dプログラムの基本的な処理フローのパターン



さらに、プログラムのエントリーポイントが記述されている「program.cs」を削除して、代わりにMain関数をForm1のコード（Form1.cs）に追加しておきます。ネームスペースにはDirectXに関係するものを追加し、必要ないものは削除します。

これでDirect3DのプログラミングテンプレートがForm1.csとして出来上がりました（リスト1）。

まだ中身は実装していませんが、Di