

オブジェクト指向と ゲームプログラミング

DirectX Graphics 2D編 - 第1回 スプライト

スプライト

スプライトとは、キャラクタを表示するための機能のことで、背景やほかのスプライト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、スプライトはハードウェアで実現される機能ですが、DirectX Graphicsでは、ポリゴンとテクスチャを用い、擬似的に実現します。また、「Direct3DX ユーティリティ ライブラリ」は、この疑似スプライトのインタフェースを提供しています。3Dの機能を用いて画像を描画(レンダリング)するので「スケーリング」「回転」「アルファブレンディング」「ミラーリング」といった機能を使用することができます。

ID3DXSpriteインタフェースの生成と解放

スプライトの機能を使うには、ID3DXSpriteインタフェースを生成する必要があります。スプライトインタフェースを使うと、スプライトの描画が簡単に行えます。

スプライトインタフェースは、D3DXCreateSprite関数で生成します。

```
// スプライトインタフェース生成(pD3DDeviceは、Direct3DDevice9オブジェクト)
ID3DXSprite pD3DXSprite;
D3DXCreateSprite(pD3DDevice, &pD3DXSprite);
```

スプライトを使用する必要がなくなった場合は、Releaseメソッドを呼び出し、解放します。

```
// スプライトインタフェース解放
pD3DXSprite->Release();
```

ID3DXSpriteインタフェースのリセット処理

デバイスが消失したり、新しいプレゼンテーションパラメータに切り替える場合、IDirect3DDevice9::Resetメソッドを呼び出します。スプライトインタフェースを生成している場合は、リセットの前にID3DXSprite::OnLostDeviceメソッドを呼び出す必要があります。さらに、リセットの後にID3DXSprite::OnResetDeviceメソッドを呼び出さなければなりません。

```
// リセット処理(present_paramsは、プレゼンテーションパラメータ)
pD3DXSprite->OnLostDevice();

pD3DDevice->Reset(&present_params);

pD3DXSprite->OnResetDevice();
```

課 題

CDXGraphics9クラスに、スプライトを描画するための初期化および解放処理を追加しましょう。

(1) CDXGraphics9クラスに、スプライトインタフェースを管理するメンバ変数を追加します。以下のプログラムを適切な場所に追加しましょう。

```
ID3DXSprite*   m_pD3DXSprite;    // スプライトインタフェース
```

(2) CDXGraphics9クラスのインスタンス生成時に、スプライトインタフェースに初期値が設定されるようにします。コンストラクタに、以下の初期化子を追加しましょう。

```
m_pD3DXSprite(NULL)
```

(3) DirectX Graphicsの初期化時に、スプライトインタフェースを生成するようにします。以下のプログラムを完成させ、適切な場所に追加しましょう。

```
// スプライトインタフェース生成
if( (::????????????????(m_pD3DDevice, &m_pD3DXSprite) != D3D_OK) {
    ::OutputDebugString("*** Error - スプライトインタフェース生成失敗(CDXGraphics9_Initialize)¥n");
    Release();
    return false;
}
```

(4) スプライトインタフェースを解放するプログラムを作成し、CDXGraphics9::Release関数に追加しましょう。

(5) 以下のプログラムを適切な場所に追加しましょう。

```
// リセット前処理
m_pD3DXSprite->OnLostDevice();
```

(6) 以下のプログラムを適切な場所に追加しましょう。

```
// リセット後処理
m_pD3DXSprite->OnResetDevice();
```