

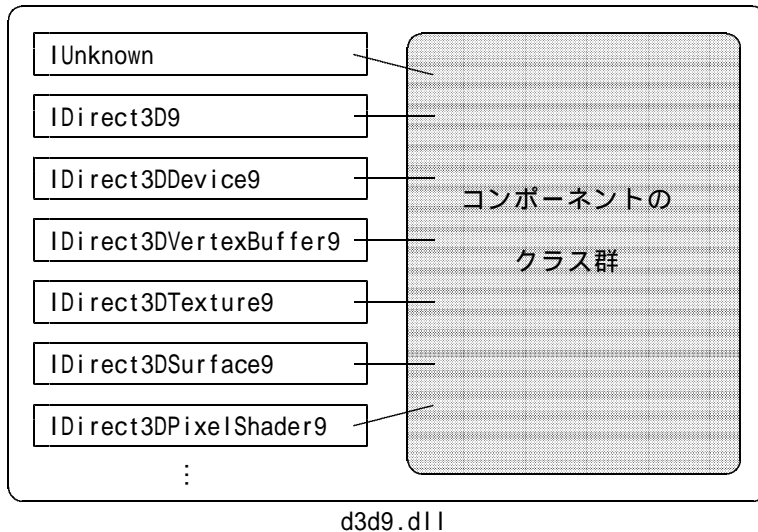
オブジェクト指向と ゲームプログラミング

DirectX Graphics編 - 第2回 Direct3Dオブジェクトの生成と解放

Direct3Dオブジェクトの生成

DirectX Graphicsの実体は、Direct3Dです。その機能を使用するには、Direct3D9コンポーネントのインスタンス(Direct3D9オブジェクト)を生成し、そのインタフェースを取得しなければなりません。

Direct3D9コンポーネントはd3d9.dllに格納されており、3Dデバイスやテクスチャなど複数のコンポーネントの集合として実装されています。Direct3D9コンポーネントのインスタンスを生成すると、最も基本的なインタフェースであるIDirect3D9が返されます。



Direct3D9オブジェクトの生成は、Direct3DCreate9関数で行います。引数には必ず「D3D_SDK_VERSION」を指定します。関数が成功すると、Direct3D9オブジェクトが生成され、そのインタフェースが返されます。

```
// Direct3D9オブジェクトの生成とインタフェースの取得
IDirect3D9* pD3D;
pD3D = Direct3DCreate9(D3D_SDK_VERSION);
```

Direct3Dオブジェクトの解放

生成または取得したCOMオブジェクトは、不用になったら解放しなければなりません。すべてのCOMオブジェクトには、解放するためのReleaseメソッドがあります。解放されたオブジェクトは再び初期化するまで使用できないので、インタフェースを格納していた変数にはNULLを代入し、無効なインタフェースであることを明示するようにします。こうしておくと、解放したインタフェースを再使用してしまい、不安定な動作になるという症状を防ぐことができます。

```
// Direct3Dオブジェクトの解放
if(pD3D != NULL) { // ポインタチェック。NULLの場合は解放しない(できない)
    pD3D->Release(); // Direct3Dオブジェクト解放
    pD3D = NULL; // NULLを代入し、無効なインタフェースであることを示す
}
```

課 題

CDXGraphics9クラスに、Direct3D9オブジェクトの生成と解放を行う処理を追加しましょう。

- (1) CDXGraphics9クラスに、Direct3D9オブジェクト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。

```
IDirect3D9* m_pD3D; // Direct3D9インタフェース
```

- (2) CDXGraphics9クラスのインスタンス生成時に、メンバ変数に初期値が設定されるようにします。コンストラクタを以下のように変更し、初期化子を追加しましょう。

```
CDXGraphics9::CDXGraphics9() : m_pD3D(NULL)
{
}
```

- (3) DirectX Graphicsを初期化してCDXGraphics9クラスを使用可能にするInitialize関数と、DirectX GraphicsおよびCDXGraphics9クラスが使用していた資源を解放するRelease関数を作成します。以下のプロトタイプを適切な場所に追加しましょう。

```
bool Initialize();
void Release();
```

- (4) Initialize関数およびRelease関数を実装します。以下のプログラムをソースファイルに追加しましょう。

```
/*
 * 初期化
 */
bool CDXGraphics9::Initialize()
{
    Release(); // 解放せずにInitialize関数が呼ばれても問題が起きないように、まず解放しておく
    return true; // 成功
}

/*
 * 解放
 */
void CDXGraphics9::Release()
{
}
```

- (5) 以下のプログラムをデストラクタの適切な場所に追加しましょう。

```
Release();
```

デストラクタに解放処理を記述しておけば、Release関数を呼び忘れたとしても、CDXGraphics9クラスのインスタンスが破棄される時にデストラクタが呼び出されるため、資源の解放漏れ(リソースリーク)を防ぐことができます。

- (6) Direct3Dオブジェクトの生成を行うプログラムを追加します。Initialize関数の適切な場所に、以下のプログラムを完成させて追加しましょう。

```
// Direct3D9オブジェクトの生成とインタフェースの取得
// ここは各自考えましょう;
if(m_pD3D == NULL) {
    ::OutputDebugString("*** Error - Direct3D9オブジェクト生成失敗(CDXGraphics9_Initialize)\n");
    return false;
}
```

(7) Direct3Dオブジェクトの解放を行うプログラムを追加します。Release関数の適切な場所に、次のプログラムを完成させて追加しましょう。

```
// Direct3Dオブジェクト解放
if(m_pD3D != NULL) {
    ここは各自考えましょう;
    ここは各自考えましょう;
}
```

(8) オブジェクトの解放処理は、Direct3Dオブジェクト以外でもまったく同じ処理で行うことが多いので、テンプレートを作成しておきます。以下のプログラムを適切な場所に追加しましょう。

```
template <class T> inline void SafeRelease(T& x) { if(x != NULL) { x->Release(); x = NULL; } }
```

テンプレート関数SafeReleaseにより、Direct3Dオブジェクトの解放は以下のように記述できます。

```
SafeRelease(m_pD3D); // Direct3Dオブジェクトの解放
```

(9) ソースファイル"CGameApp.cpp"でCDXGraphics9クラスが認識できるように、適切なヘッダファイルをインクルードしましょう。

(10) アプリケーションの初期化を行うCGameApp::Initialize関数の適切な場所に、以下のプログラムを追加しましょう。

```
// DirectX Graphics初期化
if(DXGraphics().Initialize() == false)
    return false;
```

(11) アプリケーションの解放を行うCGameApp::Release関数の適切な場所に、以下のプログラムを追加しましょう。

```
DXGraphics().Release(); // DirectX Graphics解放
```