

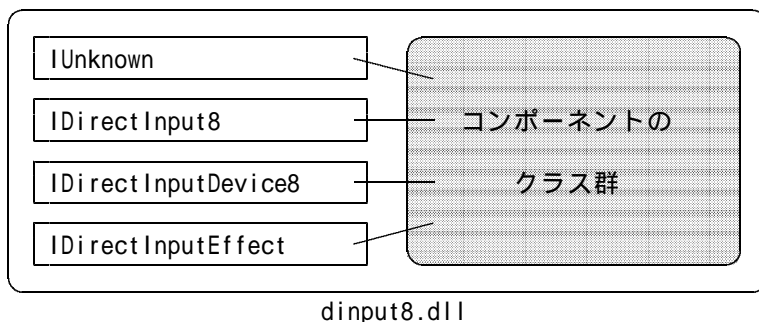
オブジェクト指向と ゲームプログラミング

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

DirectInputオブジェクトの生成

DirectInputの機能を使用するには、DirectInput8コンポーネントのインスタンス(DirectInput8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。

DirectInput8コンポーネントはdinput8.dllに格納されていて、入力デバイスやエフェクト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。DirectInput8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IDirectInput8が返されます。



DirectInput8オブジェクトの生成は、CoCreateInstance関数で行います。DirectInput8コンポーネントでは、1つ目の引数(クラスID)に「CLSID_DirectInput8」、4つ目の引数(インタフェースID)に「IID_IDirectInput8」を指定します。

```
// DirectInput8オブジェクトの生成とインタフェースの取得
IDirectInput8* pDIInput;
CoCreateInstance(CLSID_DirectInput8, NULL, CLSCTX_INPROC_SERVER,
IID_IDirectInput8, (LPVOID*)&pDIInput);
```

関数が成功すると、5つ目の引数にインタフェースが格納されます。しかし、このままではDirectInputの機能を使用することはできません。Initializeメソッドを呼び出し、DirectInput8オブジェクトを初期化する必要があります。

```
// DirectInput8オブジェクト初期化(hInstanceはインスタンスのハンドル)
pDIInput->Initialize(hInstance, 0x0800);
```

Initializeメソッドの1つ目の引数には、アプリケーションインスタンスのハンドルを指定します。ここで指定したアプリケーションとDirectInputが関連づけられます。2つ目の引数には、DirectInputのバージョンを指定します。「0x0800」を指定すると、バージョン8として初期化されます。

dinput.dllは、DirectInput8Create関数をエクスポートしています。この関数は、DirectInput8オブジェクトを生成し、初期化まで行うので非常に便利です。しかし、該当するバージョンがサポートされていないと、警告ダイアログボックスを表示してアプリケーションを強制終了してしまいます。

```
// DirectInput8オブジェクトの生成と初期化( dinput8.libが必要)
IDirectInput8* pDIInput;
DirectInput8Create(hInstance, 0x0800, IID_IDirectInput8, (LPVOID*)&pDIInput, NULL);
```

DirectInput8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、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。

```
// DirectInput8オブジェクトの解放
if(pDIInput != NULL) { // ポインタチェック。NULLの場合は解放しない(できない)
    pDIInput->Release(); // DirectInput8オブジェクト解放
}
```

```

    pDInput = NULL;          // NULLを代入し、無効なインタフェイスであることを示す
}

```

課 題

CDInput8クラスに、DirectInput8オブジェクトの生成と初期化、解放を行う処理を追加しましょう。

- (1) CDInput8クラスに、DirectInput8オブジェクトを管理するメンバ変数と、DirectInputに関連づけるウィンドウのハンドルを保持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。

```

IDirectInput8*   m_pDInput;          // DirectInput8インタフェース
HWND            m_hWnd;

```

- (2) CDInput8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。

```

CDInput8::CDInput8() : m_pDInput(NULL), m_hWnd(NULL)
{
}

```

- (3) DirectInputを初期化してCDInput8クラスを使用可能にするInitialize関数と、DirectInputおよびCDInput8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(const HWND hWnd);
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 CDInput8::Initialize(const HWND hWnd)
{
    Release();

    m_hWnd = hWnd;          // ウィンドウハンドルの保存(デバイスオブジェクトを生成するときに使います)

    return true;
}

/*****
/*                                     解 放                                     */
/*****
void CDInput8::Release()
{
    m_hWnd = NULL;
}

```

- (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、CDInput8クラスのインスタンスが破棄されるとき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) CDInput8クラスのコンストラクタの適切な場所に、COMライブラリの初期化を行うプログラムを追加しましょう。

(7) CDInput8クラスのデストラクタの適切な場所に、COMライブラリの解放を行うプログラム追加しましょう。

(8) DirectInput8オブジェクトの生成と初期化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。

```
// DirectInput8オブジェクトの生成とインタフェースの取得
if( (::  ここは各自考えましょう != S_OK) {
    ::OutputDebugString( "*** Error - DirectInput8オブジェクト生成失敗(CDInput8_Init)¥n" );
    return false;
}

// DirectInput8オブジェクト初期化
if( m_pDInput->?????????( (HINSTANCE)::GetWindowLong( hWnd, GWL_HINSTANCE ), DIRECTINPUT_VERSION )
    != DI_OK ) {
    ::OutputDebugString( "*** Error - DirectInput8オブジェクト初期化失敗(CDInput8_Init)¥n" );
    Release();
    return false;
}
```

(9) DirectInput8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、以下のプログラムを完成させて追加しましょう。

```
// DirectInput8オブジェクト解放
if( m_pDInput != NULL ) {
    这里是各自考えましょう;
    这里是各自考えましょう;
}
```

(10) オブジェクトの解放処理は、DirectInput8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、DirectInputオブジェクトの解放は以下のように記述できます。

```
SafeRelease(m_pDInput); // DirectInput8オブジェクトの解放
```

(11) ソースファイル"CGameApp.cpp"でCDInput8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。

(12) 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。

```
// DirectInput初期化
if( DInput().Initialize(GetHWnd()) == false )
    return false;
```

(13) 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。

```
DInput().Release(); // DirectInput解放
```