

ゲームプログラミング

DirectInput - 第2回 初期化と解放

DirectInputを使用するには、DirectInputオブジェクトを生成する必要があります。DirectInputオブジェクトはDirectInputそのものを表しており、DirectInputがサポートする機能は、このオブジェクトに備えられているメソッドを呼び出すことで使用することができます。

DirectInputオブジェクトの生成

DirectInputの機能を使用するには、DirectInputオブジェクトを生成する必要があります。このオブジェクトは、DirectInputコンポーネントの基本であり、DirectInputに関するほとんどのオブジェクトは、このオブジェクトから生成または取得します。

DirectInputオブジェクトにはいくつかのバージョンがあり、バージョンによって生成方法が異なります。最新のDirectInput8オブジェクトは、DirectInput8Create関数で生成します。生成に成功すると、メモリにDirectInput8オブジェクトが生成され、そのインターフェースのアドレス(ポインタ)を取得することができます。このインターフェースをとおして、DirectInput8オブジェクトを操作します。

```
LPDIRECTINPUT8  lpDInput8 = NULL;
if(DI_OK != DirectInput8Create(hInstance, 0x0800,
                               IID_IDirectInput8, (LPVOID*)&lpDInput8, NULL)) {
    // 生成失敗
}
```

DirectInput8Create関数

- 説明 -

DirectInput8Create関数は、DirectInput8オブジェクトを生成します。

- パラメータ -

1つ目の引数は、アプリケーションインスタンスのハンドルです。このハンドルはWinMain関数がプログラム起動時に受け取る1つ目の引数です。ウィンドウのハンドルからGetWindowLong関数で取得することもできます。

2つ目の引数は、使用するDirectInputのバージョンです。"DIRECTINPUT_VERSION"を指定すると、ヘッダファイル(dinput.h)で定義されているバージョンになります。0x0800を指定するとバージョン8になります。

3つ目の引数は、取得するインターフェースのIDで、DirectInput8の場合は"IID_IDirectInput8"を指定します。

4つ目の引数は、生成されるオブジェクトのインターフェースを受け取る変数のアドレスです。DirectInput8の場合は、LPDIRECTINPUT8型の変数のアドレスをLPVOID*型にキャストして指定します。

5つ目の引数は、将来の互換性のためにあり、必ずNULLにする必要があります。

- 戻り値 -

関数が成功するとDI_OK、それ以外はエラーの原因をエラーコードとして返します。

オブジェクトの操作

DirectXのオブジェクトは、すべてC++のオブジェクトと同じように、メソッドを呼び出す形式で行います。従来のAPIはC言語の関数と同じ形をしています。たとえば、仮にDirectInputを解放するDirectInputRelease関数があるとすれば、

```
DirectInputRelease(lpDInput8); // lpDInput8は初期化済みのDirectInputオブジェクト
```

のように、操作対象を引数で渡す形式になっています。DirectXは、C++のクラスオブジェクトのような形式になっているため、上記の関数呼び出しは、すべてメソッド呼び出しになります。

```
lpDInput8->Release(); // lpDInput8は初期化済みのDirectInputオブジェクト
```

操作対象であるオブジェクトから"->"演算子でメソッドを呼び出します。C++はこのように、オブジェクトに備えられたメソッドを呼び出して操作します。本来、オブジェクトのメソッドを呼び出す場合、構造体と同じように"."(ドット)演算子を使うので、

```
lpDInput8.Release(); // lpDInput8は初期化済みのDirectInputオブジェクト
```

とすべきですが、DirectXのオブジェクトへの操作はすべて、インタフェースのアドレス(すなわちポインタ)を通して間接的に行うので、"->"演算子でメソッドを呼び出します。なお、メソッドとはオブジェクトに備えられた関数で、メンバ関数と呼ばれることもあります。

オブジェクトの解放

生成または取得したDirectXオブジェクトは、プログラム終了までに必ず解放しなければなりません。すべてのオブジェクトには解放するためのReleaseメソッドがあり、これを呼び出すと解放されます。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、オブジェクトを格納していた変数には無効を表すNULLを代入し、無効なオブジェクトであることを明示するようにします。このようにおくと、解放したオブジェクトを初期化なしに再使用してしまい、不安定な動作になってしまうというバグを防ぐことができます。

```
// DirectInput8オブジェクト解放
if(NULL != lpDInput8) { // ポインタチェック、NULLの場合は解放しない(する必要がない)
    lpDInput8->Release(); // DirectInput8オブジェクト解放
    lpDInput8 = NULL; // NULLを代入し、無効なオブジェクトであることを示す
}
```

課題

以下の - 追加 1 - をあるソースファイルに追加し、DIInit関数にDirectInput8オブジェクトを生成する処理、DIRelease関数にDirectInput8オブジェクトを解放する処理を追加しましょう。なお、DIInit関数でオブジェクトの生成に失敗した場合は、- 追加 2 - を実行しデバッグメッセージを出力したあと、関数の仕様に従った値を返すように作成しましょう。

- 追加 1 -

```
static LPDIRECTINPUT8 g_lpDInput8 = NULL;
```

- 追加 2 -

```
OutputDebugString("*** Error - DirectInput8オブジェクト生成失敗(DIInit)¥n");
```