

ゲームプログラミング

DirectDraw - 第2回 初期化と解放

DirectDrawを使用するには、DirectDrawオブジェクトを生成する必要があります。DirectDrawオブジェクトはDirectDrawそのものを表しており、DirectDraw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。

DirectDrawオブジェクトの生成

DirectDrawの機能を使用するには、DirectDraw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、DirectDrawコンポーネントの基本であり、DirectDraw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。

DirectDraw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。DirectDraw7オブジェクトは、DirectDrawCreateEx関数で生成します。成功すると、メモリにDirectDraw7オブジェクトが生成され、そのインタ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。このインタフェースをとおして、DirectDraw7オブジェクトを操作します。

```
LPDIRECTDRAW7 lpDDraw7 = NULL; // DirectDraw7オブジェクト
if(DD_OK != DirectDrawCreateEx(NULL, (LPVOID*)&lpDDraw7, IID_IDirectDraw7, NULL)) {
    // 生成失敗
}
```

DirectDrawCreateEx関数

- 説明 -

DirectDrawCreateEx関数は、DirectDraw7オブジェクトを生成します。

- パラメータ -

1つ目の引数は、使用するグラフィックカードのドライバ識別子(GUID:グローバルユニーク識別子)または以下の値を指定します。

NULL	現在使用されているドライバを使用します。
DDCREATE_EMULATIONONLY	DirectDrawのすべての機能をHELで実行します。ハードウェアはまったく使用されなくなります。

2つ目の引数は、生成されるオブジェクトのインタフェースを受け取る変数のアドレスです。LPDIRECTDRAW7型の変数のアドレスをLPVOID*型にキャストして指定します。

3つ目の引数は、取得するインターフェイスのIDです。必ず"IID_IDirectDraw7"を指定します。

4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DD_OK、それ以外はエラーの原因をエラーコードで返します。

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DirectDrawを解放するDirectDrawRelease関数があるとすれば、

```
DirectDrawRelease(lpDDraw7); // lpDDraw7は初期化済みのDirectDraw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。

```
lpDDraw7->Release(); // lpDDraw7は初期化済みのDirectDraw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、

```
lpDDraw7.Release(); // lpDDraw7は初期化済みのDirectDrawオブジェクト
```

とするべきですが、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。

```
// DirectDraw7オブジェクト解放
if(NULL != lpDDraw7) { // 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)
    lpDDraw7->Release(); // DirectDraw7オブジェクト解放
    lpDDraw7 = 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、DDInit関数にDirectDraw7オブジェクトを生成する処理、DDRelease関数にDirectDraw7オブジェクトを解放する処理を追加しましょう。なお、DDInit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 LPDIRECTDRAW7 g_lpDDraw7 = NULL;
```

- 追加 2 -

```
OutputDebugString("*** Error - DirectDraw7オブジェクト生成失敗(DDInit)¥n");
```