

ゲームプログラミング

DirectShow - 第2回 マルチメディアファイルの読み込み

マルチメディアファイルの読み込みと再生にはたくさんのDirectShowオブジェクトが必要です。それらの生成、解放について説明します。

マルチメディアファイルの再生に必要なインタフェース

DirectShowにはさまざまなオブジェクトがあります。マルチメディアファイルを読み込んで再生するだけでも、以下のオブジェクトのインタフェースを使用します。

IAMultiMediaStreamオブジェクト

マルチメディアストリームオブジェクトは、マルチメディアデータを再生またはキャプチャするためのデータの経路を管理します。経路の追加、ファイルの読み込み、再生、停止などのメソッドが提供されています。

IMediaStreamオブジェクト

メディアストリームオブジェクトは、マルチメディアストリームオブジェクトから出力されたデータの特徴にあわせたアクセス方法を提供します。

IDirectDrawMediaStreamオブジェクト

DirectDrawメディアストリームオブジェクトは、DirectDraw上の映像データを制御します。

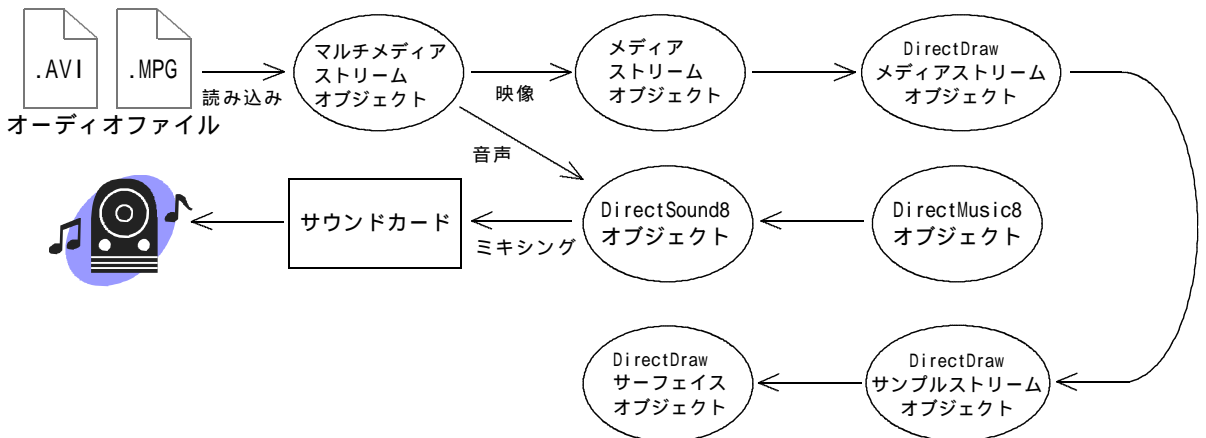
DirectDrawSampleStreamオブジェクト

DirectDrawサンプルストリームオブジェクトは、映像データに関連づけられたDirectDrawサーフェイスを設定または取得するメソッドを提供します。

DirectDrawオブジェクト

DirectDrawSurfaceオブジェクト

DirectSoundオブジェクト



マルチメディアファイルの読み込み

最初にデータの流れを管理するマルチメディアストリームオブジェクトを生成します。このオブジェクトは、DirectX Audioのオブジェクトと同じように、CoCreateInstance関数で生成します。この関数はCOMライブラリの初期化をしないと使用できないので、あらかじめCoInitialize関数で初期化しておきます。また、すべてのCOMライブラリの使用が終わったら、CoUninitialize関数でライブラリを解放する必要があります。

```
// マルチメディアストリームオブジェクト生成
IAMMultiMediaStream* piAMMStream = NULL; // マルチメディアストリームオブジェクト
CoCreateInstance(CLSID_AMMultiMediaStream, NULL, CLSCTX_INPROC_SERVER,
IID_IAMMultiMediaStream, (LPVOID*)&g_piAMMStream);
```

次に、Initializeメソッドでオブジェクトを初期化します。再生だけを行う場合は、1つ目の引数に"STREAMTYPE_READ"を指定し、読み取りモードにします。

```
// マルチメディアストリームオブジェクト初期化
piAMMStream->Initialize(STREAMTYPE_READ, 0, NULL);
```

初期化できたら、AddMediaStreamメソッドでDirectDrawとDirectSoundをメディアストリームに追加します。

AddMediaStreamメソッドの1つ目の引数は、追加するオブジェクトを指定します。2つ目の引数は、追加するオブジェクトの参照識別子(GUID)で、映像データの場合は"MSPID_PrimaryVideo"、オーディオデータの場合は"MSPID_PrimaryAudio"という変数があらかじめ定義されているので、この変数のアドレスを指定します。3つ目の引数はメディアストリームの動作フラグで、ビデオの場合は0、オーディオの場合はデフォルトレンダラ"AMMSF_ADDDEFAULTRENDERER"を指定します。4つ目の引数はNULLにします。

```
LPDIRECTDRAW7 lpDDDraw7; // DirectDraw7オブジェクト(初期化済みとする)
LPDIRECTSOUND8 lpDSound8; // DirectSound8オブジェクト(初期化済みとする)
```

```
// プライマリビデオ追加
piAMMStream->AddMediaStream(lpDDDraw7, &MSPID_PrimaryVideo, 0, NULL);
```

```
// プライマリオーディオ追加
piAMMStream->AddMediaStream(lpDSound8, &MSPID_PrimaryAudio, AMMSF_ADDDEFAULTRENDERER, NULL);
```

以上でファイルを読み込む準備が整ったので、OpenFileメソッドでファイルを読み込みます。ファイル名はDirectX Audioと同じようにUNICODE文字列で指定します。マルチバイト文字列の場合は、MultiByteToWideChar関数でUNICODE文字列に変換します。

```
TCHAR szFileName[MAX_PATH]; // 読み込むファイル名
```

```
// ファイル名をUnicodeに変換
WCHAR wszFileName[MAX_PATH];
MultiByteToWideChar(CP_ACP, 0, lpszFileName, -1, wszFileName, MAX_PATH);
```

```
// ファイル読み込み
piAMMStream->OpenFile(wszFileName, 0);
```

ファイルを読み込めばすぐに再生できますが、この状態では映像データがDirectShowの内部的なサーフェイスに転送されるようになっていました。任意のサーフェイスに転送するには、さらにいくつかのオブジェクトを生成する必要があります。

まず、プライマリビデオのメディアストリームオブジェクトを取得します。これは、GetMediaStreamメソッドで行います。1つ目の引数に"MSPID_PrimaryVideo"、2つ目の引数に取得したメディアストリームオブジェクトのインタフェースを格納する変数のアドレスを指定します。

```
// メディアストリームオブジェクト取得(プライマリビデオ)
IMediaStream* piPVMStream = NULL; // プライマリビデオメディアストリームオブジェクト
piAMMStream->GetMediaStream(MSPID_PrimaryVideo, &piPVMStream);
```

取得したオブジェクトのQueryInterfaceメソッドを呼び出し、DirectDrawメディアストリームオブジェクトを取得します。1つ目の引数に取得するインターフェイスの参照識別子"IID_IDirectDrawMediaStream"、2つ目の引数に取得したDirectDrawメディアストリームオブジェクトのインタフェースを格納する変数のアドレスをLPVOID*型にキャストして指定します。

```
// DirectDrawメディアストリームオブジェクト取得
IDirectDrawMediaStream* piDDMStream = NULL; // DirectDrawメディアストリームオブジェクト
piPVMStream->QueryInterface(IID_IDirectDrawMediaStream, (LPVOID*)&piDDMStream);
```

映像データが格納されるサーフェイスを取得するには、DirectDrawストリームサンプルオブジェクトを生成します。これはDirectDrawメディアストリームオブジェクトのCreateSampleメソッドで取得します。このメソッドの4つ目の引数に生成されたDirectDrawストリームサンプルオブジェクトのインタフェースを格納する変数アドレスを指定します。

```
// DirectDrawストリームサンプルオブジェクト生成
IDirectDrawStreamSample* piDDSSample = NULL; // DirectDrawストリームサンプルオブジェクト
piDDMStream->CreateSample(NULL, NULL, 0, &piDDSSample);
```

生成されたDirectDrawストリームサンプルオブジェクトのGetSurfaceメソッドを呼び出すと映像データが転送されるサーフェイスを取得できます。しかし、このメソッドはDirectDrawサーフェイスバージョン1を返すようになっています。DirectDrawサーフェイスバージョン7を取得したい場合は、QueryInterfaceメソッドを呼び出します。

GetSurfaceメソッドの1つ目の引数は生成したDirectDrawサーフェイスオブジェクトを格納するアドレス、2つ目の引数はRECT構造体変数のアドレスでサーフェイスのサイズが格納されます。

```
// DirectDrawSurfaceオブジェクト生成
LPDIRECTDRAWSURFACE lpdds;
RECT rcMovie;
piDDSSample->GetSurface(&lpdds, &rcMovie);

// DirectDrawSurface7オブジェクト取得
LPDIRECTDRAWSURFACE7 lpDDSDirectShow = NULL;
lpdds->QueryInterface(IID_IDirectDrawSurface7, (LPVOID*)&lpDDSDirectShow);

lpdds->Release(); // 古いバージョンのオブジェクトは不要なので解放する
```

以上でマルチメディアファイルを再生する準備が整いました。再生すると映像データは任意のサーフェイスに、音声データはDirectSoundに出力されます。

DirectShowオブジェクトの解放

DirectShowで生成、取得したオブジェクトは、これまでのオブジェクトと同じようにReleaseメソッドで解放します。

課題

マルチメディアファイルを読み込むDSLoadFromFile関数の作成と、DirectShowのオブジェクトを解放する処理を追加しましょう。

(1) マルチメディアファイルの読み込み時に、DirectDraw7オブジェクトとDirectShow用のサーフェイスオブジェクトが必要になります。DDUtils.cppの変数宣言でstaticをはずし、ほかのソースファイルから参照できるように変更します。

- 変更1 -

```
LPDIRECTDRAW7 g_lpDDraw7 = NULL;
LPDIRECTDRAWSURFACE7 g_lpDDSurface7[DDS_MAX] = {NULL};
```

(2) マルチメディアファイルの読み込み時に、DirectSound8オブジェクトが必要になります。DXAUtils.cppの変数宣言でstaticをはずし、ほかのソースファイルから参照できるように変更します。

- 変更2 -

```
LPDIRECTSOUND8 g_lpDSound8 = NULL;
```

(3) 以下のプログラムをDSUtils.cppの適切な場所に追加しましょう。この宣言により、DDUtil.cppおよびDXAUtils.cppで宣言されているグローバル変数(g_lpDDraw7, g_lpDDSurface7, g_lpDSound8)がDSUtils.cppから参照できるようになります。

- 追加1 -

```
extern LPDIRECTDRAW7 g_lpDDraw7;
extern LPDIRECTDRAWSURFACE7 g_lpDDSurface7[DDS_MAX];
extern LPDIRECTSOUND8 g_lpDSound8;
```

(3) マルチメディアファイルの読み込みと再生に必要なとなるオブジェクトのインタフェースを格納する変数を適切な場所に追加しましょう。

- 追加2 -

```
static IAMMultiMediaStream* g_piAMMStream = NULL; // マルチメディアストリームオブジェクト
static IMediaStream* g_piPVMStream = NULL; // メディアストリームオブジェクト(プライマリビデオ)
static IDirectDrawMediaStream* g_piDDMStream = NULL; // DirectDrawメディアストリームオブジェクト
static IDirectDrawStreamSample* g_piDDSSample = NULL; // DirectDrawストリームサンプルオブジェクト
```

(4)以下のプログラムは、マルチメディアファイルを読み込んで映像をDirectDraw、音声をDirectSoundに出力するように設定するDSLoadFromFile関数です。関数の仕様をよく読み、プログラムを完成させましょう。なお、この関数は次回作成するDSSStop関数も必要です。

DSLoadFromFile関数

- 説明 -

DSLoadFromFile関数は、マルチメディアファイルを読み込み、映像をDDS_DIRECTSHOWで指定されるサーフェイスに、音声をDirectSoundに出力するように設定します。

- パラメータ -

LPCSTR lpszFileName...読み込むマルチファイル名(読み込める形式は環境によって異なります)

- 戻り値 -

成功した場合はtrue、それ以外はfalseを返します。

- 追加7 -

```
/*
*****
*/
/*          ファイル読み込み          */
*****
*/
bool DSLoadFromFile(LPCSTR lpszFileName)
{
    DSRelease();
    DDRReleaseSurface(DDS_DIRECTSHOW);

    // マルチメディアストリームオブジェクト生成
    if(FAILED(????????????(CLSID_AmmMultiMediaStream, NULL, CLSCTX_INPROC_SERVER,
        IID_IAMMultiMediaStream, (LPVOID*)&g_piAMMStream))) {
        OutputDebugString("**** Error - マルチメディアストリームオブジェクト生成失敗(DSLoadFromFile)¥n");
        return false;
    }

    // マルチメディアストリームオブジェクト初期化
    if(FAILED(ここは各自考えましょう)) {
        OutputDebugString("**** Error - マルチメディアストリームオブジェクト初期化失敗(DSLoadFromFile)¥n");
        DSRelease();
        return false;
    }

    // プライマリビデオ追加
    if(FAILED(ここは各自考えましょう[ヒント: DirectDraw7オブジェクトはg_lpDDDraw7])) {
        OutputDebugString("**** Error - プライマリビデオ追加失敗(DSLoadFromFile)¥n");
        DSRelease();
        return false;
    }

    // プライマリオーディオ追加
    if(FAILED(ここは各自考えましょう[ヒント: DirectSoundオブジェクトはg_lpDSound8])) {
        OutputDebugString("**** Error - プライマリオーディオ追加失敗(DSLoadFromFile)¥n");
        DSRelease();
        return false;
    }

    // ファイル名をUnicodeに変換
    WCHAR wszFileName[MAX_PATH];
    ?????????????????(CP_ACP, 0, lpszFileName, -1, wszFileName, MAX_PATH);
    // ファイル読み込み
    if(FAILED(g_piAMMStream->????????(wszFileName, 0))) {
        OutputDebugString("**** Error - ファイル読み込み失敗(DSLoadFromFile)¥n");
        DSRelease();
        return false;
    }

    // メディアストリームオブジェクト(プライマリビデオ)取得
    if(FAILED(g_piAMMStream->????????(MSPID_PrimaryVideo, &g_piPVMStream))) {
        OutputDebugString("**** Error - プライマリビデオメディアストリーム取得失敗(DSLoadFromFile)¥n");
        DSRelease();
        return false;
    }

    // DirectDrawメディアストリームオブジェクト取得
    if(FAILED(g_piPVMStream->????????(IID_IDirectDrawMediaStream, (LPVOID*)&g_piDDMStream))) {
```

```

        OutputDebugString("**** Error - DirectDrawメディアストリーム取得失敗(DSLoadFromFile)¥n");
        DSRelease();
        return false;
    }

    // DirectDrawストリームサンプルオブジェクト生成
    if(FAILED(g_piDDMSample->?????????(NULL, NULL, 0, &g_piDDSSample))) {
        OutputDebugString("**** Error - DirectDrawストリームサンプル生成失敗(DSLoadFromFile)¥n");
        DSRelease();
        return false;
    }

    // DirectDrawSurfaceオブジェクト生成
    LPDIRECTDRAWSURFACE lpdds;
    if(FAILED(g_piDDSSample->?????????(&lpdds, NULL))) {
        OutputDebugString("**** Error - DirectDrawSurfaceオブジェクト生成失敗(DSLoadFromFile)¥n");
        DSRelease();
        return false;
    }

    // DirectDrawSurface7オブジェクト取得
    HRESULT hr = lpdds->?????????(IID_IDirectDrawSurface7,
                                   (LPVOID*)&g_lpDDSsurface7[DDS_DIRECTSHOW]);
    lpdds->Release();
    if(FAILED(hr)) {
        OutputDebugString("**** Error - DirectDrawSurface7オブジェクト取得失敗(DSLoadFromFile)¥n");
        DSRelease();
        return false;
    }

    return true;
}

```

(5) DSRelease関数ですべてのDirectShowオブジェクトを解放するようにプログラムを変更しましょう。
 なお、関数中のDSSStop関数は、次回作成します。

- 変更 2 -

```

/*****
/*
DirectShow解放
*/
*****/
void DSRelease()
{
    DSSStop();

    // DirectDrawストリームサンプル解放
    if(??? != g_piDDSSample) {
        g_piDDSSample->?????();
        g_piDDSSample = NULL;
    }

    // DirectDrawメディアストリーム解放
    if(??? != g_piDDMSample) {
        g_piDDMSample->?????();
        g_piDDMSample = NULL;
    }

    // プライマリビデオメディアストリーム解放
    if(??? != g_piPVMStream) {
        g_piPVMStream->?????();
        g_piPVMStream = NULL;
    }

    // マルチメディアストリーム解放
    if(??? != g_piAMMStream) {
        g_piAMMStream->?????();
        g_piAMMStream = NULL;
    }
}

```