オブジェクト指向と ゲームプログラミング

▶ コンポーネント編 － 第９回　MCIWnd ◀

MCI

MCI(Media Control Interface)は、マルチメディアデバイスやマルチメディアファイルを、特定のデバイスに依存しないで制御できるようにするためのAPIです。MCIを利用することにより、異なるマルチメディアデバイスおよびマルチメディアファイルでも共通の関数を使って制御することができます。

MCIWnd

MCIには、関数名の先頭が「MCIWnd」で始まるものがあります。これらの関数は、マルチメディアファイルを扱うためのウィンドウ(MCIウィンドウ)を生成するのが特徴で、このウィンドウ内でマルチメディアファイルに対してさまざまな操作を行うことができます。生成されるウィンドウの形状やサイズも指定でき、ファイル操作用のツールバーも備えることができます。
MCIウィンドウを生成するのがMCIWndCreate関数です。これらのMCIWndから始まる関数を使用する場合は、ヘッダファイル<vfw.h>とスタティックライブラリ"vfw32.lib"が必要になります。

MCIWndCreate関数

－説明－
MCIWndCreate関数は、MCIWndウィンドウクラスを登録して、MCIサービスを使うためにMCIWndウィンドウを生成します。また、MCIデバイスまたはファイル(AVIファイルなど)を開き、MCIWndウィンドウと関連付けます。

－書式－
HWND MCIWndCreate(HWND *hwndParent*, HINSTANCE *hInstance*, DWORD *dwStyle*, LPSTR *szFile*)

－パラメーター－
１つ目の引数(hwndParent)は、親ウィンドウのハンドルを指定します。このデバイスコンテキストと互換性のあるビットマップが生成されます。

２つ目の引数(hInstance)は、MCIウィンドウと関連付けるプログラムのインスタンスハンドルを指定します。

３つ目の引数(dwStyle)は、ウィンドウスタイルのフラグで指定します。CreateWindowEx関数で使用するウィンドウスタイルのほかに、MCIWndウィンドウで使うスタイルも指定できます。よく使うMCIウィンドウスタイルは、以下のものです。

MCIWNDF_NOERRORDLG	MCIエラーを表示しません
MCIWNDF_NOMENU	メニューを非表示にし、ポップアップメニューにアクセスできないようにします
MCIWNDF_NOOPEN	メニューの「開く」と「閉じる」コマンドを非表示にし、これらの選択肢にアクセスできないようにします
MCIWNDF_NOPLAYBAR	ツールバーを非表示にし、アクセスできないようにします

４つ目の引数(szFile)は、オープンするMCIデバイスまたはファイルの名前を指定します。たとえば、"cdaudio"を指定すると音楽CDをオープンすることができます。

－戻り値－
関数が成功すると、MCIウィンドウのハンドルが返ります。関数が失敗すると、０が返ります。

```
// MCIウィンドウを生成し、ムービーを開く
HWND   hMCIWnd = MCIWndCreate(hWnd, hInstance, 0, "Movie\\Opening.avi");
```

MCIWndCreate関数では、MCIウィンドウの表示座標を指定することができません。表示座標を指定したい場合は、CreateWindowEx関数でMCIウィンドウを生成します。MCIウィンドウの生成は、通常のウィンドウとは若干手順が異なり、以下のようになります。

(１)MCIWndRegisterClass関数でMCIウィンドウのクラスを登録
(２)CreateWindowEx関数でMCIウィンドウを生成(ウィンドウクラス名"MCIWND_WINDOW_CLASS"を使用)
(３)MCIWndOpen関数でファイルを開く

```
// MCIウィンドウのクラスを登録
MCIWndRegisterClass();

// MCIウィンドウ生成(クライアント座標(0, 16)にMCIウィンドウを生成)
HWND   hMCIWnd = CreateWindowEx(WS_EX_TOPMOST, MCIWND_WINDOW_CLASS, "MCIWND",
                                WS_CHILD | WS_VISIBLE,
                                0, 16, 0, 0,
                                hWnd, NULL, hInstance, NULL);

// MCIウィンドウ生成成功
MCIWndOpen(hMCIWnd, "Movie\\Opening.avi", 0);       // ファイルを開く
MCIWndSetZoom(hMCIWnd, 200);                        // ２倍に拡大
MCIWndPlay(hMCIWnd);                                // 再生開始
```

CreateWindowEx関数の５つ目と６つ目の引数がMCIウィンドウの表示座標になります。７つ目と８つ目の引数では、ウィンドウのサイズを指定することができますが、MCIウィンドウのサイズはムービーなどのイメージの大きさに合わせて自動的に設定されるため、これらの値は無視されます。ムービーなどのイメージの拡大率は、MCIWndSetZoomマクロで変更することができます。

MCIウィンドウにマルチメディアデバイスまたはマルチメディアファイルをオープンして関連づけるのがMCIWndOpen関数です。オープンしたデバイスおよびファイルは、不用になったらMCIWndCloseマクロでクローズします。このマクロはデバイスおよびファイルを閉じるだけです。MCIウィンドウ自体を破棄する場合は、MCIWndDestroyマクロを呼び出します。

※MCIWndRegisterClass(パラメータなし)
MCIウィンドウクラス(登録名MCIWND_WINDOW_CLASS)を登録します。成功すると０、それ以外の場合は０以外の値を返します。

※MCIWndOpen(MCIWndのハンドル，オープンファイル名，フラグ)
ファイルとそれにあわせたデバイスを初期化してオープンし、MCIウィンドウと関連付けます。ファイルを再生に使用する場合は、フラグに０を指定します。

MCIWndマクロ

MCIウィンドウを制御するために、さまざまなマクロが用意されています。これらのマクロは、関数とほとんど同じように使用できます。よく使うのは以下のマクロです(カッコ内はパラーメータです)。

MCIWndPlay(MCIWndのハンドル)…ファイルを再生します。
MCIWndStop(MCIWndのハンドル)…再生を停止します。
MCIWndSetSpeed(MCIWndのハンドル，スピード)…スピードを変更します。1000が標準です。
MCIWndSetVolume(MCIWndのハンドル，ボリューム)…ボリュームを変更します。1000が標準です。
MCIWndSetZoom(MCIWndのハンドル，倍率)…ムービーのサイズを変更します。100が標準サイズです。
MCIWndClose(MCIWndのハンドル)…MCIウィンドウとMCIデバイスをクローズします。
MCIWndDestroy(MCIWndのハンドル)…MCIウィンドウとMCIデバイスを破棄します。
MCIWndGetPosition(MCIWndのハンドル)…再生位置を取得します。
MCIWndGetEnd(MCIWndのハンドル)…終了位置を取得します。

課　　題

MCIウィンドウを管理するクラスCMCIWndを作成しましょう。

(１)CMCIWndクラスのヘッダファイル(MCIWnd.hpp)を以下のように作成しましょう。

－MCIWnd.hpp－

```
/*
=============================================================================================
                                オブジェクト指向ゲームプログラミング
     Programmed by Hibikino software. Copyright (c) 2004 Hibikino software. All rights reserved.
=============================================================================================

  【対象ＯＳ】
     Microsoft Windows98/2000以降

  【コンパイラ】
     Microsoft VisualC++ 6.0J ServicePack6

  【プログラム】
     MCIWnd.hpp
                MCIウィンドウクラスヘッダ

  【履歴】
     * Version    1.00       2004/04/dd  hh:mm:ss

=============================================================================================
*/

#pragma once

/*********************************************************************************************/
/*                                        インクルードファイル                                 */
/*********************************************************************************************/
#include <windows.h>

/*********************************************************************************************/
/*                                      MCIウィンドウクラス定義                                */
/*********************************************************************************************/
class CMCIWnd {
public:
    CMCIWnd();
    ~CMCIWnd() { Destroy(); }

    bool Create(LPCTSTR inFileName, const HWND hWnd, const int inPosX, const int inPosY,
                const LONG inZoom);
    void Destroy();

    bool Play();
    void Stop();

    void Suspend();
    void Resume();

    bool IsPlayCompleted() const;

private:
    HWND    m_hMCIWnd;
    TCHAR   m_FileName[MAX_PATH + 1];
    LONG    m_Zoom;
    LONG    m_EndPos;
    LONG    m_CurrentPos;

    CMCIWnd(const CMCIWnd&);
    CMCIWnd& operator= (const CMCIWnd&);
};
```

(２)CMCIWndクラスのメンバは、以下のとおりです。

CMCIWnd コンストラクタ

CMCIWndオブジェクトを構築します。

~CMCIWnd デストラクタ

CMCIWndオブジェクトを解放します。

Create 生成

指定されたウィンドウの子ウィンドウとしてMCIウィンドウを生成し、指定されたデバイスまたはファイルを開いて関連づけます。

書式
```
bool Create(LPCTSTR inFileName, const HWND hWnd, const int inPosX, const int inPosY,
            const LONG inZoom);
```

Return	成功：true　それ以外：false
inFileName	ファイル名
hWnd	親ウィンドウのハンドル
inPosX	MCIウィンドウのｘ座標
inPosX	MCIウィンドウのｙ座標
inZoom	ムービーの拡大率

Destroy 破棄

MCIウィンドウを破棄します。

Play 制御

ファイルの先頭から再生を開始します。

書式
```
bool Play();
```

Return	成功：true　それ以外：false

Stop 制御

ファイルの再生を停止します。

Suspend 制御

再生を一時停止します。

Resume 制御

一時停止から復帰します。

IsPlayCompleted 判定

再生が終了したかどうか調べます。

書式
```
bool IsPlayCompleted() const;
```

Return	再生終了：true　それ以外：false

m_hMCIWnd メンバ変数

CMCIWndオブジェクトが保持するMCIウィンドウのハンドルです。

書式
```
HWND m_hMCIWnd;
```

m_FileName メンバ変数

オープンされているファイル名です。

書式
```
TCHAR m_FileName[MAX_PATH + 1];
```

m_Zoom メンバ変数

オープンされているファイルの拡大率です。

書式
```
LONG m_Zoom;
```

m_EndPos メンバ変数

オープンされているファイルの終了位置を保持します。

書式
```
LONG m_EndPos;
```

m_CurrentPos メンバ変数

一時停止をした時点での再生位置を保持します。

書式
```
LONG m_CurrentPos;
```

(３)CMCIWndクラスのソースファイル(MCIWnd.cpp)を以下のように作成しましょう。

－MCIWnd.cpp－

```
/*
================================================================================================
                                オブジェクト指向ゲームプログラミング
    Programmed by Hibikino software. Copyright (c) 2004 Hibikino software. All rights reserved.
================================================================================================

  【対象ＯＳ】
     Microsoft Windows98/2000以降

  【コンパイラ】
     Microsoft VisualC++ 6.0J ServicePack6

  【プログラム】
     MCIWnd.cpp
                MCIウィンドウクラスヘッダ

  【履歴】
     * Version    1.00       2004/04/dd  hh:mm:ss

================================================================================================
*/

/*************************************************************************************************/
/*                                        インクルードファイル                                   */
/*************************************************************************************************/
#include "MCIWnd.hpp"
#include <※ここは各自考えましょう>

/*************************************************************************************************/
/*                                        静的リンクライブラリ                                   */
/*************************************************************************************************/
#pragma comment(lib, "※ここは各自考えましょう")

/*************************************************************************************************/
/*                                      デフォルトコンストラクタ                                 */
/*************************************************************************************************/
CMCIWnd::CMCIWnd() : m_hMCIWnd(NULL), m_Zoom(0), m_EndPos(0), m_CurrentPos(-1)
{
    ::ZeroMemory(m_FileName, sizeof(m_FileName));
}

/*************************************************************************************************/
/*                                      MCIウィンドウ生成                                        */
/*************************************************************************************************/
bool CMCIWnd::Create(LPCTSTR inFileName, const HWND hWnd, const int inPosX, const int inPosY,
                     const LONG inZoom)
{
    return true;
}

/*************************************************************************************************/
/*                                          MCIクローズ                                          */
/*************************************************************************************************/
void CMCIWnd::Destroy()
{
}

/*************************************************************************************************/
/*                                            再　生                                             */
/*************************************************************************************************/
bool CMCIWnd::Play() const
{
    return true;
}


/*************************************************************************************************/
/*                                            停　止                                             */
/*************************************************************************************************/
void CMCIWnd::Stop() const
{
```

```
}

/*****************************************************************************************************/
/*                                          サスペンド                                               */
/*****************************************************************************************************/
void CMCIWnd::Suspend()
{
}

/*****************************************************************************************************/
/*                                          復　帰                                                   */
/*****************************************************************************************************/
void CMCIWnd::Resume()
{
}

/*****************************************************************************************************/
/*                                        再生終了判定                                               */
/*****************************************************************************************************/
bool CMCIWnd::IsPlayCompleted() const
{
    return true;
}
```

(４)Create関数の足りない部分を補い、プログラムを完成させましょう。

```
bool CMCIWnd::Create(LPCTSTR inFileName, const HWND hWnd, const int inPosX, const int inPosY,
                     const LONG inZoom)
{
    Destroy();

    const HINSTANCE   hInstance = (HINSTANCE)::GetWindowLong(hWnd, GWL_HINSTANCE);

    WNDCLASSEX   wcx;
    // ウィンドウクラスが登録済みか調べる
    if(::GetClassInfoEx(hInstance, MCIWND_WINDOW_CLASS, &wcx) == 0) {
        // MCIウィンドウクラス登録
        if(::??????????????????() == 0) {
            ::OutputDebugString("*** Error - MCIウィンドウクラス登録失敗(CMCIWnd_Create)\n");
            return false;
        }
    }

    // MCIウィンドウ生成
    m_hMCIWnd = ::??????????????(::GetWindowLong(hWnd, GWL_EXSTYLE) & WS_EX_TOPMOST,
                                 ????????????????????, "MCIWND",
                                 WS_CHILD | WS_VISIBLE |
                                 MCIWNDF_NOMENU | MCIWNDF_NOPLAYBAR | MCIWNDF_NOERRORDLG,
                                 inPosX, inPosY, 0, 0,
                                 hWnd, NULL, hInstance, NULL);
    if(m_hMCIWnd == NULL) {
        ::OutputDebugString("*** Error - MCIウィンドウ生成失敗(CMCIWnd_Create)\n");
        Destroy();
        return false;
    }

    // ファイルオープン
    if(??????????(m_hMCIWnd, inFileName, 0) != 0) {
        ::OutputDebugString("*** Error - ファイルオープン失敗(CMCIWnd_Create)\n");
        Destroy();
        return false;
    }

    ????????????(m_hMCIWnd, inZoom);                // 拡大率設定
    m_Zoom   = inZoom;                              // 拡大率保存
    m_EndPos = MCIWndGetEnd(m_hMCIWnd);             // 終了位置取得
    ::strncpy(m_FileName, inFileName, MAX_PATH);    // ファイル名保存

    return true;
}
```

(５)Destroy関数の足りない部分を補い、プログラムを完成させましょう。

```
void CMCIWnd::Destroy()
{
    if(m_hMCIWnd != NULL) {
        MCIWndStop   (m_hMCIWnd);
        MCIWndClose  (m_hMCIWnd);
        ????????????(m_hMCIWnd);
        m_hMCIWnd = NULL;
    }

    ::ZeroMemory(m_FileName, sizeof(m_FileName));

    m_Zoom       = 0;
    m_EndPos     = 0;
    m_CurrentPos = -1;
}
```

(６)Play関数の足りない部分を補い、プログラムを完成させましょう。

```
bool CMCIWnd::Play()
{
    if(m_hMCIWnd == NULL) {
        ::OutputDebugString("*** Error - MCIウィンドウ未生成(CMCIWnd_Play)\n");
        return false;
    }

    m_CurrentPos = -1;
    if(????????????(m_hMCIWnd) != 0)
        return false;

    return true;
}
```

(７)Stop関数の足りない部分を補い、プログラムを完成させましょう。

```
void CMCIWnd::Stop()
{
    if(m_hMCIWnd == NULL) {
        ::OutputDebugString("*** Error - MCIウィンドウ未生成(CMCIWnd_Stop)\n");
        return;
    }

    m_CurrentPos = -1;
    ????????????(m_hMCIWnd);
}
```

(８)Suspend関数を以下のように作成しましょう。

```
void CMCIWnd::Suspend()
{
    if(m_hMCIWnd == NULL) {
        ::OutputDebugString("*** Error - MCIウィンドウ未生成(CMCIWnd_Suspend)\n");
        return;
    }

    m_CurrentPos = MCIWndGetPosition(m_hMCIWnd);
    MCIWndStop (m_hMCIWnd);
    MCIWndClose(m_hMCIWnd);
}
```

(９)Suspend関数を以下のように作成しましょう。

```
void CMCIWnd::Resume()
{
    if(m_hMCIWnd == NULL) {
        ::OutputDebugString("*** Error - MCIウィンドウ未生成(CMCIWnd_Resume)\n");
        return;
    }
    if(m_CurrentPos < 0) {
```

```
        ::OutputDebugString("*** Error - サスペンド無効(CMCIWnd_Resume)\n");
        return;
    }

    if(MCIWndOpen(m_hMCIWnd, m_FileName, 0) != 0) {
        MCIWndSeek(m_hMCIWnd, m_EndPos);
        return;
    }

    MCIWndSetZoom(m_hMCIWnd, m_Zoom);
    MCIWndPlayFrom(m_hMCIWnd, m_CurrentPos);
    m_CurrentPos = -1;
}
```

(10)IsPlayCompleted関数の足りない部分を補い、プログラムを完成させましょう。

```
bool CMCIWnd::IsPlayCompleted() const
{
    if(m_hMCIWnd == NULL)
        return true;

    if(??????????????????(m_hMCIWnd) != m_EndPos)
        return false;

    return true;
}
```

(11)CMCIWndクラスを使ってマルチメディアファイルを再生してみましょう。CTestSceneクラスの適切な場所に、以下のようにプログラムを追加、変更しましょう。

・CTestSceneクラスのメンバに追加

```
CMCIWnd   m_Movie;
```

・CTestScene::CTestScene関数を以下のように変更

```
CTestScene::CTestScene() : m_Movie()
{
    // ここに、機能テストの初期化処理を記述します
    m_Movie.Create("Movie\\Opening.mpg", GameApp().GetHWnd(), 0, 48, 200);
    m_Movie.Play();

    FPSTimer().Reset();        // タイマリセット
}
```

・CTestScene::ActivateProc関数を以下のように変更

```
int CTestScene::ActiveProc()
{
    // ここに、機能テストのメイン処理を記述します
    // 内部処理

    // 描画処理
    if(FPSTimer().IsSkip() == false) {
        HDC   hDC = ::GetDC(GameApp().GetHWnd());
        FPSTimer().DrawFPS(hDC);
        ::ReleaseDC(GameApp().GetHWnd(), hDC);
    }

    return 0;
}
```