

オブジェクト指向と ゲームプログラミング

DirectX Audio編 - 第11回 エフェクト

エフェクト

エフェクトとは、再生時に効果が現れるさまざまな音響効果のことです。コーラスやエコーといった標準エフェクトが提供されているほか、独自のカスタムエフェクトを作成することができます。

エフェクトはサウンドバッファに設定します。複数のエフェクトを同じサウンドバッファに設定したり、設定したエフェクトのパラメータを取得し、変更することができます。

DirectMusicのオーディオパスからIDirectSoundBuffer8インタフェースを取得することができるので、DirectMusicにエフェクトを加えることもできます。

エフェクトの設定

エフェクトの設定は、IDirectSoundBuffer8::SetFXメソッドで行います。エフェクトの設定を行うには、バッファの生成時にDSBCAPS_CTRLFXフラグを指定する必要があります。なお、DSBCAPS_CTRLFXフラグで生成されたバッファは、メモリを共有するバッファとすることができません。

IDirectSoundBuffer8::SetFXメソッド

- 説明 -

SetFXメソッドは、サウンドバッファのエフェクトを有効にします。複数のエフェクトを有効にすることもできます。バッファが再生中であつたり、ロックされていると失敗します。

- 書式 -

```
HRESULT SetFX(DWORD dwEffectsCount, LPDSEFFECTDESC pDSFXDesc, LPDWORD pdwResultCodes);
```

- パラメータ -

1つ目の引数(dwEffectsCount)は、エフェクトの数です。2, 3つ目の引数で指定する配列は、必ずこの値以上の要素数でなければなりません。この値を0にし、2, 3つ目の引数をNULLにすることにより、バッファからすべてのエフェクトを削除することができます。

2つ目の引数(pDSFXDesc)は、エフェクト情報を格納したDSEFFECTDESC構造体の配列のアドレスです。

3つ目の引数(pdwResultCodes)は、エフェクトの作成を試みた結果を受け取るDWORD配列のアドレスです。各々のエフェクトに対して結果が返されます。結果が必要ない場合はNULLを指定します。

- 戻り値 -

成功した場合はDS_OKまたはDS_INCOMPLETE、それ以外はエラーコードを返します。

エフェクトの情報を格納するDSEFFECTDESC構造体は、以下のメンバがあります。

メンバ	データ型	説明
dwSize	DWORD	この構造体のサイズ(バイト数)
dwFlags	DWORD	フラグ。エフェクトをハードウェアまたはソフトウェアに生成するように指定できるが、DirectX9.0では必ずソフトウェアに生成されるので、0を指定する
guidDSFXClass	GUID	エフェクトのGUID
dwReserved1	DWORD_PTR	予約されているため、0でなければならない
dwReserved2	DWORD_PTR	予約されているため、0でなければならない

guidDSFXClassメンバに指定できる標準エフェクトとして、以下のGUIDが定義されています。

GUID_DSFX_STANDARD_CHORUS	コーラス
GUID_DSFX_STANDARD_COMPRESSOR	コンプレッサ
GUID_DSFX_STANDARD_DISTORTION	ディストーション
GUID_DSFX_STANDARD_ECHO	エコー
GUID_DSFX_STANDARD_FLANGER	フランジ
GUID_DSFX_STANDARD_GARGLE	ガーグル
GUID_DSFX_STANDARD_I3DL2REVERB	Interactive3D-Level2リバーブ
GUID_DSFX_STANDARD_PARAMEQ	パラメトリックイコライザ
GUID_DSFX WAVES_REVERB	ウェーブリバーブ(16ビットのオーディオフォーマットのみ)

```

// エフェクトの設定(pDSBufferは初期化済みのDirectSoundBuffer8オブジェクト)
// 演奏を停止し、エフェクトを削除する
pDSBuffer->Stop();
pDSBuffer->SetFX(0, NULL, NULL);

// エフェクト(コーラス)を設定
// DSEFFECTDESC構造体設定
DSEFFECTDESC dsed;
ZeroMemory(&dsed, sizeof(dsed));
dsed.dwSize = sizeof(DSEFFECTDESC);
dsed.guidDSFXClass = GUID_DSFX_STANDARD_CHORUS; // コーラス

// エフェクト設定
pDSBuffer->SetFX(1, &dsed, NULL);

// 複数のエフェクトを設定
// DSEFFECTDESC構造体設定
DSEFFECTDESC dsed[3];
ZeroMemory(&dsed, sizeof(dsed));
dsed[0].dwSize = sizeof(DSEFFECTDESC);
dsed[0].guidDSFXClass = GUID_DSFX_STANDARD_CHORUS; // コーラス

dsed[1].dwSize = sizeof(DSEFFECTDESC);
dsed[1].guidDSFXClass = GUID_DSFX_STANDARD_DISTORTION; // ディストーション

dsed[2].dwSize = sizeof(DSEFFECTDESC);
dsed[2].guidDSFXClass = GUID_DSFX_STANDARD_ECHO; // エコー

// エフェクト設定
pDSBuffer->SetFX(3, dsed, NULL);

```

課 題

サウンドバッファクラスに、エフェクトを設定する機能を追加しましょう。

(1) SoundBuffer.hppに、エフェクトを指定するためのフラグを定義します。

```

/*****
/*
// エフェクトフラグ定義
*/
/*****
enum DXAFX {
    DXAFX_CHORUS = 1 << 0, // コーラス
    DXAFX_COMPRESSOR = 1 << 1, // コンプレッション
    DXAFX_DISTORTION = 1 << 2, // ディストーション
    DXAFX_ECHO = 1 << 3, // エコー
    DXAFX_FLANGER = 1 << 4, // フランジ
    DXAFX_GARGLE = 1 << 5, // ガーグル
    DXAFX_ENVREVERB = 1 << 6, // 環境リバーブ
    DXAFX_PARAMEQ = 1 << 7, // パラメトリックイコライザ
    DXAFX_WAVESREVERB = 1 << 8, // ミュージックリバーブ
    DXAFX_MAX = 9
};

```

(2) サウンドバッファの生成時に、DSBCAPS_CTRLFXフラグを追加し、エフェクトを制御できるようにプログラムを変更しましょう。

(3) エフェクトの設定はSetFX関数として作成します。SetFX関数には、有効にしたいエフェクトをエフェクトフラグを組み合わせで渡します。0の場合には、エフェクトを削除します。これをふまえると、関数のプロトタイプは以下ようになります。ISoundBufferクラスの適切な場所に追加しましょう。

```
virtual bool SetFX(const DWORD inFXFlags) = 0;
```

(4) CSoundBuffer::SetFX関数のプロトタイプを定義し、以下のプログラムの足りない部分を補って適切な場所に追加しましょう。

```
/*
 *                                     エフェクト設定
 */
bool CSoundBuffer::SetFX(const DWORD inFXFlags)
{
    Stop();

    // エフェクト削除
    m_pDSBuffer->SetFX(0, NULL, NULL);
    if(inFXFlags == 0)
        return true;

    // エフェクト設定
    int cnt = 0;
    DSEFFECTDESC EffectDesc[DXAFX_MAX];
    ::ZeroMemory(EffectDesc, sizeof(EffectDesc));
    for(int i = 0; i < DXAFX_MAX; i++) {
        EffectDesc[i].dwSize = sizeof(DSEFFECTDESC);
        EffectDesc[i].guidDSFXClass = GUID_NULL;
    }

    // コーラス
    if((inFXFlags & DXAFX_CHORUS) != 0) {
        EffectDesc[FXCnt].guidDSFXClass = ここは各自考えましょう;
        cnt++;
    }

    // コンプレッション
    if((inFXFlags & DXAFX_COMPRESSOR) != 0) {
        EffectDesc[FXCnt].guidDSFXClass = ここは各自考えましょう;
        cnt++;
    }

    // ディストーション
    if((inFXFlags & DXAFX_DISTORTION) != 0) {
        EffectDesc[FXCnt].guidDSFXClass = ここは各自考えましょう;
        cnt++;
    }

    // エコー
    if((inFXFlags & DXAFX_ECHO) != 0) {
        EffectDesc[FXCnt].guidDSFXClass = ここは各自考えましょう;
        cnt++;
    }

    // フランジ
    if((inFXFlags & DXAFX_FLANGER) != 0) {
        EffectDesc[FXCnt].guidDSFXClass = ここは各自考えましょう;
        cnt++;
    }

    // ガーグル
    if((inFXFlags & DXAFX_GARGLE) != 0) {
        EffectDesc[FXCnt].guidDSFXClass = ここは各自考えましょう;
        cnt++;
    }

    // 環境リバーブ
    if((inFXFlags & DXAFX_ENVREVERB) != 0) {
        EffectDesc[FXCnt].guidDSFXClass = ここは各自考えましょう;
        cnt++;
    }

    // パラメトリックイコライザ
    if((inFXFlags & DXAFX_PARAMEQ) != 0) {
        EffectDesc[FXCnt].guidDSFXClass = ここは各自考えましょう;
        cnt++;
    }

    // ミュージックリバーブ
    if((inFXFlags & DXAFX_WAVESREVERB) != 0) {
        EffectDesc[FXCnt].guidDSFXClass = ここは各自考えましょう;
        cnt++;
    }

    if(FAILED(m_pDSBuffer-> ここは各自考えましょう)) {
        ::OutputDebugString("**** Error - エフェクト設定失敗(CSoundBuffer_SetFX)%n");
        return false;
    }
}
```

```
    return true;
}
```

(5) CNullSoundBuffer クラスに、「何もしない」SetFX関数を実装しましょう。

(6) サウンドバッファの生成後に、以下のプログラムを実行し、エフェクトが有効になるかどうか確認しましょう。

```
m_pSE->SetFX(DXAFX_CHORUS | DXAFX_ECHO);    // コーラスとエコー
```

このほかのエフェクトもどのようにするか確認しましょう。