ゲームプログラミング

▶　DirectX Audio－第１１回　エフェクト　◀

DirectSoundでは、サウンドバッファにエフェクトを設定することができ、再生時にさまざまな音響効果を加えることができます。

エフェクト

エフェクトは再生時に効果が現れるさまざまな音響効果のことで、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もできます。
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時に設定することもできます。設定したエフェクトからパラメータを取得し、変更することができます。また、DirectMusicのオーディオパスからDirectSoundBuffer8オブジェクト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、DirectSoundBuffer8オブジェクトの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。

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ったり、ロックされていると失敗します。また、あらかじめCoInitialize関数が呼び出されいる必要があります。

－パラメータ－
１つ目の引数は、有効にする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つ目の引数は、有効にする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。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、それ以外はエラーの原因をエラーコードで返します。

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GUIDとして、以下の標準エフェクト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(lpDSBuffer8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
lpDSBuffer8->Stop();
lpDSBuffer8->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
lpDSBuffer8->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
lpDSBuffer8->SetFX(3, dsed, NULL);
```

○ 課題 ○

サウンドバッファにエフェクトを設定する処理を追加しましょう。

(１)ヘッダファイルDXAUtils.hに、エフェクトを指定するためのフラグを定義します。
－追加１－

```
// エフェクトフラグ
enum DXA_FX {
    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)以下のプログラムは、セカンダリバッファのエフェクトを設定するDXASetBufferFX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。

DXASetBufferFX関数
－説明－
DXASetBufferFX関数は、指定されたセカンダリバッファの音量を設定します。

－パラメータ－
const DXA_SB dxaBuf…セカンダリバッファの指定。DXA_SB列挙体を指定。DXA_SB_MAXを指定すると、すべてのセカンダリバッファに対してエフェクトを設定する
DWORD dwFXFlags…設定するエフェクトフラグ(DXA_FX列挙型)を組み合わせて指定。0を指定した場合、エフェクトが削除される

－戻り値－
成功した場合はtrue、それ以外はfalseを返す。dxaBufがDXA_SB_MAXの場合は、つねにtrueを返す。

```
/*****************************************************************************************************/
/*                                    セカンダリバッファエフェクト設定                                 */
/*****************************************************************************************************/
bool DXASetBufferFX(const DXA_SB dxaBuf, const DWORD dwFXFlags)
{
    if(DXA_SB_MAX != dxaBuf && NULL == g_lpDSBuf8[dxaBuf]) {
        OutputDebugString("*** Error - バッファ未生成(DXASetBufferFX)\n");
        return false;
    }

    // 再生の停止とエフェクトの削除
    if(DXA_SB_MAX != dxaBuf) {
        g_lpDSBuf8[dxaBuf]->????();
        g_lpDSBuf8[dxaBuf]->※ここは各自考えましょう
    } else {
        for(int i = 0; i < DXA_SB_MAX; i++) {
            if(NULL != g_lpDSBuf8[i]) {
                g_lpDSBuf8[i]->????();
                g_lpDSBuf8[i]->※ここは各自考えましょう
            }   // if
        }   // for
    }   // if
    if(0 == dwFXFlags)
        return true;

    // DSEFFECTDESC構造体設定
    int             nFXCnt = 0;  // エフェクト数
    DSEFFECTDESC    dsEffectDesc[DXAFX_MAX];
    ZeroMemory(dsEffectDesc, sizeof(dsEffectDesc));
    for(int i = 0; i < DXAFX_MAX; i++)
        dsEffectDesc[i].dwSize = sizeof(DSEFFECTDESC);

    // コーラス
    if(0 != (dwFXFlags & DXAFX_CHORUS)) {
        dsEffectDesc[nFXCnt].guidDSFXClass = ※ここは各自考えましょう
        nFXCnt++;
    }
    // コンプレッション
    if(0 != (dwFXFlags & DXAFX_COMPRESSOR)) {
        dsEffectDesc[nFXCnt].guidDSFXClass = ※ここは各自考えましょう
        nFXCnt++;
    }
    // ディストーション
    if(0 != (dwFXFlags & DXAFX_DISTORTION)) {
        dsEffectDesc[nFXCnt].guidDSFXClass = ※ここは各自考えましょう
        nFXCnt++;
    }
    // エコー
    if(0 != (dwFXFlags & DXAFX_ECHO)) {
        dsEffectDesc[nFXCnt].guidDSFXClass = ※ここは各自考えましょう
        nFXCnt++;
    }
    // フランジ
    if(0 != (dwFXFlags & DXAFX_FLANGER)) {
        dsEffectDesc[nFXCnt].guidDSFXClass = ※ここは各自考えましょう
        nFXCnt++;
    }
    // ガーグル
    if(0 != (dwFXFlags & DXAFX_GARGLE)) {
        dsEffectDesc[nFXCnt].guidDSFXClass = ※ここは各自考えましょう
        nFXCnt++;
    }
    // 環境リバーブ
    if(0 != (dwFXFlags & DXAFX_ENVREVERB)) {
        dsEffectDesc[nFXCnt].guidDSFXClass = ※ここは各自考えましょう
        nFXCnt++;
    }
    // パラメトリックイコライザ
    if(0 != (dwFXFlags & DXAFX_PARAMEQ)) {
        dsEffectDesc[nFXCnt].guidDSFXClass = ※ここは各自考えましょう
```

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

    // エフェクト設定
    if(DXA_SB_MAX != dxaBuf) {
        if(FAILED(g_lpDSBuf8[dxaBuf]->※ここは各自考えましょう)) {
            OutputDebugString("*** Error - エフェクト設定失敗(DXASetBufferFX)\n");
            return false;
        }
    } else {
        for(int i = 0; i < DXA_SB_MAX; i++) {
            if(NULL != g_lpDSBuf8[i])
                g_lpDSBuf8[i]->※ここは各自考えましょう
        }   // for
    }   // if

    return true;
}
```

(４)サウンドバッファ１にWaveファイルを読み込み、以下のプログラムを実行してエフェクトが正しく設定されているかを確認しましょう。

```
    // エフェクト設定(コーラスとエコー)
    DXASetBufferFX(DXA_SB1, DXAFX_CHORUS | DXAFX_ECHO);
```