

オブジェクト指向と ゲームプログラミング

DirectX Audio編 - 第10回 ボリューム、パン、再生周波数の設定

サウンドパラメータの設定

DirectSoundでは、サウンドバッファごとにボリューム、パン、再生周波数といったパラメータの設定を行うことができます。これらのパラメータは、再生中でも停止中でも行うことができ、リアルタイムに反映されます。

ボリュームの設定

ボリュームの設定は、IDirectSoundBuffer8::SetVolumeメソッドで行います。引数は音量で1/100dB単位で設定します。範囲は「DSBVOLUME_MAX」から「DSBVOLUME_MIN」です。これらの値は、それぞれ0および10,000と定義されています。DSBVOLUME_MAXは、本来の音量(減衰なし)です。DSBVOLUME_MINは、音量が100dB減衰していることを表し、実質的に無音となります。DirectSoundは、増幅をサポートしていないので、0より大きな値を指定することはできません。

ボリュームの設定を行うには、バッファの生成時にDSBCAPS_CTRLVOLUMEフラグを指定しておく必要があります。

```
// ボリューム設定(pDSBufferは、初期化済みのDirectSoundBuffer8オブジェクト)
pDSBuffer->SetVolume(-1000); // 10dB減衰
```

ボリュームの設定はプライマリバッファに対しても行うことができます。プライマリバッファのボリュームを変更すると、DirectSound全体のボリュームが変化します。

パンの設定

パンとは、左右のチャンネルの相対ボリュームのことです。パンの設定は、IDirectSoundBuffer8::SetPanメソッドで行います。引数は相対ボリュームで1/100dB単位で設定します。範囲は「DSBPAN_LEFT」から「DSBPAN_RIGHT」です。これらの値は、それぞれ10,000および-10,000と定義されています。DSBPAN_LEFTは、右チャンネルが100dB減衰し、実質的に右チャンネルが無音となります。DSBPAN_RIGHTは、左チャンネルが100dB減衰し、実質的に左チャンネルが無音となります。DSBPAN_CENTERは中央値で0と定義されており、両方のチャンネルが同じ音量となります。一方のチャンネルが減衰しても、もう一方は最大音量のままです。また、パンの効果は、ボリュームの影響を受けます。

パンの設定を行うには、バッファの生成時にDSBCAPS_CTRLPANフラグを指定する必要があります。

```
// パン設定
pDSBuffer->SetPan(DSBPAN_LEFT); // サウンドをもっとも左に寄せる
```

パンの設定はプライマリバッファに対しても行うことができます。プライマリバッファのパンを変更すると、DirectSound全体のパンが変化します。

再生周波数の設定

再生周波数を変更すると再生ピッチ(再生速度)が変化しますが、バッファのフォーマットには影響を与えません。再生周波数の設定は、IDirectSoundBuffer8::SetFrequencyメソッドで行います。引数は周波数です。100,000以上は、環境によってはサポートされません。「DSBFREQUENCY_ORIGINAL」を指定すると、元の状態に戻すことができます。

再生周波数の設定を行うには、バッファの生成時にDSBCAPS_CTRLFREQUENCYフラグを指定する必要があります。

```
// 再生周波数設定
pDSBuffer->SetFrequency(22050); // 再生周波数を22.05kHzに設定
```

再生周波数の設定はプライマリバッファに対して行うことはできません。

課 題

CSoundBufferクラスに、ボリューム、パン、再生周波数を設定する機能を追加しましょう。

(1) ボリュームの設定はSetVolume関数、パンの設定はSetPan関数、再生周波数の設定はSetFrequency関数として作成します。それぞれの関数には、新たな設定値を渡します。これをふまえると、各関数のプロトタイプは以下のようになります。ISoundBufferクラスの適切な場所に追加しましょう。

```
virtual void SetVolume(LONG inVolume) = 0;
virtual void SetPan(LONG inPan) = 0;
virtual void SetFrequency(DWORD inFrequency) = 0;
```

(2) CSoundBufferクラスに、それぞれの関数を実装します。以下のプログラムを適切な場所に追加しましょう。

```
virtual void SetVolume(LONG inVolume)      {   ここは各自考えましょう(inVolume); }
virtual void SetPan(LONG inPan)            {   ここは各自考えましょう; }
virtual void SetFrequency(DWORD inFrequency) {   ここは各自考えましょう; }
```

(3) CNullSoundBufferクラスに、それぞれの関数を「何もしない」関数として実装しましょう。

(4) CDXAUDIO8クラスの適切な場所に、プライマリバッファのボリュームを設定することによりDirect Sound全体の音量を変更する以下のSetMasterVolume関数の足りない部分を補い、追加しましょう。

```
/*
 *          マスターボリューム設定
 */
void CDXAUDIO8::SetMasterVolume(LONG inVolume)
{
    if(m_pDSBPrimary == NULL) {
        ::OutputDebugString("*** Error - プライマリバッファ未生成(CDXAUDIO8_SetMasterVolume)¥n");
        return;
    }
    ここは各自考えましょう(ヒント: プライマリバッファの音量の設定を行います)
}
```

(5) ボリューム、パン、再生周波数、マスターボリュームの設定が正しく行われるかどうかを確認しましょう。