

ゲームプログラミング

DirectX Audio - 第9回 サウンドバッファの再生と停止

DirectSoundでは、サウンドバッファの再生時にボイス管理を行うことができます。

ボイス管理

ボイス管理とは、ハードウェアリソース(サウンドカード)を最大限に利用するために、サウンドへのリソースの割り当てを再生時まで遅延したり、優先順位の低いサウンドを早めに停止してそのリソースを解放し、新しいサウンドにそのリソースを割り当てることをいいます。

通常、サウンドはハードウェアに割り当てられ、バッファが破棄されるまで解放されません。ハードウェアに空きがない場合はソフトウェア(メインメモリ)に割り当てられ、CPUでバッファの管理とミキシングを行います。ハードウェアに割り当てられる数は環境により異なりますが、あまり多くはないので、多くのバッファを作成する場合、一部のバッファはソフトウェアに存在する可能性があります。

ボイス管理を使うと、バッファの再生時にリソースの割り当てと解放が行えるので、ハードウェアの利用制限を克服することができます。

ボイス管理を行うには、セカンダリバッファの生成時にDSBCAPS_LOCDEFERフラグを指定する必要があります。

サウンドバッファの再生

サウンドバッファの再生は、DirectSoundBufferオブジェクトのPlayメソッドで行います。再生は、現在の再生カーソル位置から行われます。再生カーソルは、DirectSoundBufferのSetCurrentPositionメソッドで変更できます。0を指定すると先頭に戻ります。

Playメソッド

- 説明 -

Playメソッドは、サウンドバッファを再生カーソルの位置から再生します。

- パラメータ -

1つ目の引数は、予約済なので0でなければなりません。

2つ目の引数は、サウンドの優先度です。この優先度は、ボイス管理が使います。最も低い優先度は0、最も高い優先度は0xFFFFFFFFです。バッファがDSBCAPS_LOCDEFERフラグで作成されていない場合、この値は0でなければなりません。

3つ目の引数は、バッファの再生方法を指定するフラグです。以下のフラグを組み合わせます。

・ループフラグ

DSBPLAY_LOOPING

ループ再生を行います

・ボイス割り当てフラグ

バッファがDSBCAPS_LOCDEFERフラグで作成されている場合に有効です。

DSBPLAY_LOCHARDWARE

ハードウェアで再生します

DSBPLAY_LOCSOFTWARE

ソフトウェアで再生します

・ボイス管理フラグ

バッファがDSBCAPS_LOCDEFERフラグで作成されている場合に有効です。ハードウェアに空きがない場合に新しいバッファ用の空きを確保する方法を指定します。

DSBPLAY_TERMINATEBY_PRIORITY

優先度が最も低いバッファを停止します

DSBPLAY_TERMINATEBY_TIME

再生の残り時間が少ないバッファを停止します

DSBPLAY_TERMINATEBY_DISTANCE

DSBCAPS_MUTE3DATMAXDISTANCEフラグが設定され、最大距離を超えているバッファを停止します。このようなバッファがない場合は失敗します。

- 戻り値 -

成功した場合はDS_OK、それ以外はエラーの原因をエラーコードで返します。

```
// 再生(IpDSBuffer8は初期化済みのDirectSoundBuffer8オブジェクト)
IpDSBuffer8->Play(0, 0, 0); // ループなし
IpDSBuffer8->Play(0, 0, DSBPLAY_LOOPING); // ループあり
```

サウンドバッファの停止

サウンドバッファの停止は、DirectSoundBufferオブジェクトのStopメソッドで行います。停止しても、再生カーソルの位置は先頭に戻りません。続けてPlayメソッドを呼び出せば、停止した位置から再生を再開できます。

```
// 停止  
lpDSBuffer8->Stop();
```

課題

セカンダリバッファの再生と停止を行う関数を作成しましょう。

(1)以下のプログラムは、セカンダリバッファを再生するDXAPlayBuffer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。

DXAPlayBuffer関数

- 説明 -

DXAPlayBuffer関数は、指定されたセカンダリバッファを先頭から再生します。

- パラメータ -

const DXA_SB dxaBuf...再生するセカンダリバッファの指定。DXA_SB列挙体を指定

const DWORD dwPriority...再生の優先度

const DWORD dwFlags...再生フラグ。ループフラグ、ボイス割り当てフラグ、ボイス管理フラグを組み合わせて指定

- 戻り値 -

再生に成功した場合はtrue、それ以外はfalseを返します。

```
/*  
*                               セカンダリバッファ再生                               *  
*/  
bool DXAPlayBuffer(const DXA_SB dxaBuf, const DWORD dwPriority, const DWORD dwFlags)  
{  
    if(NULL == g_lpDSBuf8[dxaBuf]) {  
        OutputDebugString("**** Error - バッファ未生成(DXAPlayBuffer)¥n");  
        return false;  
    }  
  
    // 再生  
    g_lpDSBuf8[dxaBuf]-> ここは各自考えましょう; // 先頭に戻す  
    if( ここは各自考えましょう) {  
        OutputDebugString("**** Error - バッファ再生失敗(DXAPlayBuffer)¥n");  
        return false;  
    }  
  
    return true;  
}
```

(2)以下のプログラムは、セカンダリバッファを停止するDXAStopBuffer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。

DXAStopBuffer関数

- 説明 -

DXAStopBuffer関数は、指定されたセカンダリバッファを停止します。

- パラメータ -

const DXA_SB dxaBuf...停止するセカンダリバッファの指定。DXA_SB列挙体を指定

- 戻り値 -

なし

```
/******  
/*                               セカンダリバッファ停止                               */  
/******  
void DXAStopBuffer(const DXA_SB dxaBuf)  
{  
    if(NULL == g_lpDSBuf8[dxaBuf])  
        return;  
    ここは各自考えましょう(バッファを停止します)  
}
```