

ゲームプログラミング

DirectX Audio - 第14回 リスナー

DirectSound3Dでは、出力されるサウンドは、3Dサウンドバッファとリスナーによって決定されます。リスナーとは、音を聞くオブジェクトのことです。

リスナーオブジェクトの取得

リスナーは、DirectSound3DListenerオブジェクトで管理します。このオブジェクトは、プライマリバッファから取得します。プライマリバッファの生成時にDSBCAPS_CTRL3Dフラグを指定すると、生成後にQueryInterfaceメソッドで取得できるようになります。

```
// プライマリバッファ生成 (lpDSound8は、初期化済みのDirectSound8オブジェクト)
LPDIRECTSOUNDBUFFER lpDSBPrimary = NULL;

DSBUFFERDESC dsbd;
ZeroMemory(&dsbd, sizeof(dsbd));
dsbd.dwSize = sizeof(dsbd);
dsbd.dwFlags = DSBCAPS_PRIMARYBUFFER | DSBCAPS_CTRLVOLUME | DSBCAPS_CTRL3D;
dsbd.dwBufferBytes = 0;
dsbd.lpwfxFormat = NULL;
lpDSound8->CreateSoundBuffer(&dsbd, &lpDSBPrimary, NULL);

// リスナー取得
LPDIRECTSOUND3DLISTENER8 lpDS3DListener8 = NULL; // リスナーオブジェクト
lpDSBPrimary->QueryInterface(IID_IDirectSound3DListener8, (LPVOID*)&lpDS3DListener8);
```

リスナーオブジェクトの設定

リスナーオブジェクトは、方向、位置、速度、ロールオフ係数、ドップラー乗数を設定することができます。これらの設定値と3Dサウンドバッファの設定値をもとに、スピーカから出力される最終的なサウンドが計算されます。なお、3Dサウンドでないサウンドバッファは、これらの影響を受けないので、通常どおり出力されます。

方向の設定

3Dサウンドの効果を正しく計算するには、リスナーが向いている方向を設定する必要があります。リスナーの方向の設定は、DirectSound3DListenerオブジェクトのSetOrientationメソッドで行います。このとき、Direct3Dのカメラと同じように「リスナーの前方ベクトル」と「リスナーの上方ベクトル」を設定します。この2つのベクトルは直角でなければならず、DirectSound3Dは前方ベクトルを設定した後、必要に応じて上方ベクトルを調整します。

```
// 方向設定
lpDS3DListener8->SetOrientation(0.0f, 0.0f, 1.0f, // リスナーの前方を示す方向ベクトル
                                0.0f, 1.0f, 0.0f, // リスナーの上方を示す方向ベクトル
                                DS3D_IMMEDIATE);
```

現在の方向は、GetOrientationメソッドで取得することができます。

位置の設定

リスナーの位置の設定は、DirectSound3DListenerオブジェクトのSetPositionメソッドで行います。また、GetPositionメソッドで現在の位置を取得することができます。

```
// 位置設定
lpDS3DListener8->SetPosition(0.0f, 0.0f, 0.0f, DS3D_IMMEDIATE);
```

速度の設定

リスナーの速度の設定は、DirectSound3DListenerオブジェクトのSetVelocityメソッドで行います。また、GetVelocityメソッドで現在の速度を取得することができます。

```
// 速度設定
lpDS3DListener8->SetVelocity(0.0f, 0.0f, 1.0f, DS3D_IMMEDIATE);
```

この速度は、DirectSound3Dがドップラー効果を計算するための値であり、実際にリスナーが移動するわけではありません。

ロールオフ

ロールオフとは、リスナーとサウンドバッファの距離による音量の減衰の度合いのことです。ロールオフの設定は、DirectSound3DListenerオブジェクトのSetRolloffFactorメソッドで行います。値の範囲はDS3D_MINROLLOFFFACTOR(ロールオフなし)からDS3D_MAXROLLOFFFACTOR(実世界の10倍の効果)です。デフォルト値はDS3D_DEFAULTROLLOFFFACTORで、実世界と同じ減衰量になっています。

```
// ロールオフ係数設定
lpDS3DListener8->SetRolloffFactor(2.0f, DS3D_IMMEDIATE); // 2倍
```

ドップラー効果

ドップラー効果は、リスナーと3Dサウンドバッファの相対的な速度により決定されますが、DirectSound3DListenerオブジェクトのSetDopplerFactorメソッドで効果を調整することができます。設定できる範囲はDS3D_MINDOPPLERFACTOR(ドップラー効果なし)からDS3D_MAXDOPPLERFACTOR(実世界の10倍の効果)です。デフォルト値はDS3D_DEFAULTDOPPLERFACTORで、実世界と同じドップラー効果になっています。

```
// ドップラー乗数設定
lpDS3DListener8->SetDopplerFactor(1.5f, DS3D_IMMEDIATE); // 1.5倍
```

すべてのパラメータの設定

DirectSound3DListenerオブジェクトのSetAllParametersメソッドは、リスナーに関するすべてのパラメータの設定を行うことができ、GetAllParametersメソッドですべてのパラメータを取得することができます。なお、これらのメソッドを呼び出す前に、引数で渡すDS3DLISTENER構造体のdwSizeメンバに構造体のサイズを設定しておく必要があります。

```
// すべてのパラメータの設定
DS3DLISTENER ds3dListener;
ZeroMemory(&ds3dListener, sizeof(ds3dListener));
ds3dListener.dwSize = sizeof(ds3dListener);
ds3dListener.vPosition = D3DXVECTOR3(0.0f, 0.0f, 0.0f); // 位置
ds3dListener.vVelocity = D3DXVECTOR3(0.0f, 0.0f, 0.0f); // 速度ベクトル
ds3dListener.vOrientFront = D3DXVECTOR3(0.0f, 0.0f, 1.0f); // 前方ベクトル
ds3dListener.vOrientTop = D3DXVECTOR3(0.0f, 1.0f, 0.0f); // 上方ベクトル
ds3dListener.flDistanceFactor = 1.0f; // 距離係数
ds3dListener.flRolloffFactor = DS3D_DEFAULTROLLOFFFACTOR; // ロールオフ係数
ds3dListener.flDopplerFactor = DS3D_DEFAULTDOPPLERFACTOR; // ドップラー乗数
lpDS3DListener8->SetAllParameters(&ds3dListener, DS3D_IMMEDIATE);
```

課題

リスナーの取得と設定を行う処理を追加しましょう。

(1)以下のプログラムを適切な場所に追加しましょう。

```
static LPDIRECTSOUND3DLISTENER8 g_lpDS3DListener8 = NULL; // リスナーオブジェクト
```

(2)以下のプログラムをDXAInit関数の適切な場所に追加し、リスナーオブジェクトを取得ように変更しましょう。

```
// プライマリバッファ生成
DSBUFFERDESC dsbd;
ZeroMemory(&dsbd, sizeof(dsbd));
dsbd.dwSize = sizeof(dsbd);
dsbd.dwFlags = DSBCAPS_PRIMARYBUFFER | DSBCAPS_CTRLVOLUME | ??????????????;
dsbd.dwBufferBytes = 0;
dsbd.lpwfxFormat = NULL;
if(DS_OK != g_lpDSound8->CreateSoundBuffer(&dsbd, &g_lpDSBPrimary, NULL)) {
    OutputDebugString("*** Error - プライマリバッファ生成失敗(DXAInit)¥n");
    DXARelease();
    return false;
}

// リスナー取得
if(S_OK != ここは各自考えましょう)
    OutputDebugString("*** Error - リスナー取得失敗(DXAInit)¥n");
```

(3)以下のプログラムは、リスナーの方向を設定するDXASetListenerOrientation関数です。関数を完成させ、追加しましょう。なお、引数は左からリスナーの前方を示すベクトル、リスナーの上方を示すベクトルです。

```
/*
 * リスナー方向設定
 */
void DXASetListenerOrientation(const D3DVECTOR& dvFront, const D3DVECTOR& dvTop)
{
    if(NULL == g_lpDS3DListener8) {
        OutputDebugString("*** Error - リスナー未初期化(DXASetListenerOrientation)¥n");
        return;
    }
    g_lpDS3DListener8-> ここは各自考えましょう
}
```

(4)以下のプログラムは、リスナーの方向を設定するDXASetListenerPosition関数です。関数を完成させ、追加しましょう。なお、引数はリスナーの位置です。

```
/*
 * リスナー位置設定
 */
void DXASetListenerPosition(const D3DVECTOR& dvPos)
{
    if(NULL == g_lpDS3DListener8) {
        OutputDebugString("*** Error - リスナー未初期化(DXASetListenerPosiotion)¥n");
        return;
    }
    g_lpDS3DListener8-> ここは各自考えましょう
}
```

(5)以下のプログラムは、リスナーの速度を設定するDXASetListenerVelocity関数です。関数を完成させ、追加しましょう。なお、引数はリスナーの速度ベクトルです。

```
/*
 * リスナー速度設定
 */
void DXASetListenerVelocity(const D3DVECTOR& dvVelocity)
{
    if(NULL == g_lpDS3DListener8) {
        OutputDebugString("*** Error - リスナー未初期化(DXASetListenerVelocity)¥n");
        return;
    }
    g_lpDS3DListener8-> ここは各自考えましょう
}
```

(6)以下のプログラムは、ロールオフ係数を設定するDXASetRollOffFactor関数です。関数を完成させ、追加しましょう。なお、引数はロールオフ係数です。

```

/*****
/*                                     ロールオフ係数設定                                     */
/*****
void DXASetRollOffFactor(const D3DVALUE dvRollOff)
{
    if(NULL == g_lpDS3DListener8) {
        OutputDebugString("**** Error - リスナー未初期化(DXASetRollOffFactor)¥n");
        return;
    }
    g_lpDS3DListener8-> ここは各自考えましょう
}

```

(7)以下のプログラムは、ドップラー乗数を設定するDXASetDopplerFactor関数です。関数を完成させ、追加しましょう。なお、引数はドップラー乗数です。

```

/*****
/*                                     ドップラー乗数設定                                     */
/*****
void DXASetDopplerFactor(const D3DVALUE dvRollOff)
{
    if(NULL == g_lpDS3DListener8) {
        OutputDebugString("**** Error - リスナー未初期化(DXASetDopplerFactor)¥n");
        return;
    }
    g_lpDS3DListener8-> ここは各自考えましょう
}

```

(8)以下のプログラムは、リスナーのすべてのパラメータを設定するDXASetListenerAllParameters関数です。関数を完成させ、追加しましょう。なお、引数はリスナーの情報が格納されたDS3DBUFFER構造体の変数です。

```

/*****
/*                                     リスナー全パラメータ設定                                     */
/*****
void DXASetListenerAllParameters(const DS3DLISTENER& ds3DListener)
{
    if(NULL == g_lpDS3DListener8) {
        OutputDebugString("**** Error - リスナー未初期化(DXASetListenerAllParameters)¥n");
        return;
    }
    g_lpDS3DListener8-> ここは各自考えましょう
}

```