

オブジェクト指向と ゲームプログラミング

DirectInput編 - 第8回 アクセス権

アクセス権の消失と再取得

DirectInputが管理する入力デバイスには、アクセス権があり、IDirectInputDevice8::Acquireメソッドで取得することができます。アクセス権を取得するとそのデバイスから情報を取得することができますようになります。

DirectInputDevice8オブジェクトの協調レベルがフォアグラウンドモードの場合、アプリケーションが非アクティブ状態になると、デバイスへのアクセス権を解放します。また、ほかのアプリケーションがより高い協調レベルを取得した場合も、デバイスへのアクセス権が失われます。

アクセス権が無くなると、デバイスから情報を取得することはできません。いくつかのメソッド(GetDeviceState, GetDeviceData, Poll)は、「DIERR_INPUTLOST」または「DIERR_NOTACQUIRED」のエラーを返し、アクセス権が失われていることを知らせます。アクセス権が失われた場合は、IDirectInputDevice8::Acquireメソッドで再取得します(ただし、ほかのアプリケーションが排他占有している場合は取得できません)。

アクセス権の解放

協調レベルがバックグラウンドモードの場合は、アプリケーションが非アクティブ状態になってもアクセス権は解放されず、デバイスから情報を取得することができます。しかし、排他占有モードの場合は、アクセス権を解放しないと、OSを含むほかのアプリケーションがそのデバイスに対するいっさいのアクセスができないので、操作不能などの影響を及ぼします。キーボードやマウスは必ず解放し、ほかのアプリケーションが使用できるようにします。

アクセス権の解放は、IDirectInputDevice8::Unacquireメソッドで行います。

```
// アクセス権解放(pDIDeviceは初期化済みのDirectInputDevice8オブジェクト)
pDIDevice->Unacquire();
```

課 題

CDInput8クラスに、アクセス権を管理する機能を追加しましょう。

(1) 入力デバイスクラスに、アクセス権を取得するAcquire関数とアクセス権を解放するUnacquire関数を追加します。以下のプログラムを適切な場所に追加しましょう。

```
virtual void Acquire() = 0;
virtual void Unacquire() = 0;
```

(2) CInputDeviceクラスにAcquire関数とUnacquire関数を実装します。以下のプログラムの足りない部分を補い、適切な場所に追加しましょう。

```
virtual void Acquire() { m_pDIDevice->???????(); }
virtual void Unacquire() { m_pDIDevice->?????????(); }
```

(3) CNullInputDeviceクラスに、「何もしない」Acquire関数とUnacquire関数を実装しましょう。

(4) CDInput8クラスに、すべてのデバイスのアクセス権を取得するAcquire関数を追加します。以下のプログラムの足りない部分を補い、適切な場所に追加しましょう。

```
/*
 *                               デバイスアクセス権取得
 */
void CDInput8::Acquire()
{
    if(m_pDIInput == NULL) {
```

```

        ::OutputDebugString("*** Error - DirectInput未初期化(CDInput8_Acquire)¥n");
        return;
    }

    m_pKeyboard->?????????();
    m_pMouse  ->?????????();

    for(int i = 0; i < m_GamePad.size(); i++)
        m_GamePad[i]->?????????();
}

```

(5) CDInput8クラスに、すべてのデバイスのアクセス権を解放するUnacquire関数を追加します。以下のプログラムの足りない部分を補い、適切な場所に追加しましょう。

```

/*****
/*                      デバイスアクセス権解放                      */
/*****
void CDInput8::Unacquire()
{
    if(m_pDInput == NULL) {
        ::OutputDebugString("*** Error - DirectInput未初期化(CDInput8_Unacquire)¥n");
        return;
    }

    m_pKeyboard->?????????();
    m_pMouse  ->?????????();

    for(int i = 0; i < m_GamePad.size(); i++)
        m_GamePad[i]->?????????();
}

```

(6) アプリケーションが非アクティブ状態になったときに、ほかのアプリケーションが入力デバイスを使用できるように、すべてのデバイスのアクセス権を解放します。以下のプログラムを適切な場所に追加しましょう(ヒント: CGameAppクラス)。

```

DInput().Unacquire();    // 入力デバイスアクセス権解放

```

(6) アプリケーションがアクティブ状態になったときに、入力デバイスを使用できるように、すべてのデバイスのアクセス権を取得します。以下のプログラムを適切な場所に追加しましょう。

```

DInput().Acquire();    // 入力デバイスアクセス権取得

```

(7) GetState関数に、「消失したアクセス権の復元処理」を追加します。以下のプログラムの足りない部分を補い、適切な場所に追加しましょう。

```

/*****
/*                      状態取得                      */
/*****
bool CInputDevice::GetState(const DWORD inSize, LPVOID pState)
{
    const HRESULT hr = m_pDIDevice->GetDeviceState(inSize, pState);
    if(hr == DI_OK)
        return true;

    ::OutputDebugString("*** Error - 状態取得失敗(CInputDevice_GetState)¥n");
    ::ZeroMemory(pState, inSize);

    // アクセス権再取得
    if(hr == ?????????????? || hr == ??????????????????)
        m_pDIDevice->Acquire();

    return false;
}

```

(8) (7)を参考に、GetBuffer関数とFlushBuffer関数にアクセス権の復元処理を追加しましょう。