

# ゲームプログラミング

## DirectDraw - 第16回 サーフェイスのロストと復元

サーフェイスはロスト(消失)してしまふことがあります。ロストしたサーフェイスへの操作はほとんどがエラーになります。

### サーフェイスのロスト

アプリケーションが非アクティブ状態になったり、ディスプレイモードが変更されたりすると、サーフェイスが保有するグラフィックデータを保持するためのメモリ領域(サーフェイスメモリ)が解放されてしまうことがあります。この状態をサーフェイスのロスト(消失)といいます。特に、サーフェイスがVRAMに作成されている場合はほとんどロストしてしまいます。ロストした状態では、多くのメソッドがDDERR\_SURFACELOSTエラーを返し、ほとんどの機能が実行できなくなります。

サーフェイスがロストしているかどうかは、サーフェイスオブジェクトのIsLostメソッドで調べることができます。このメソッドの戻り値がDDERR\_SURFACELOSTの場合はロストしていることを表し、DD\_OKの場合はロストしていないことを表します。

```
if(DDERR_SURFACELOST == lpDDSurface->IsLost())  
    // サーフェイスはロストしている
```

### サーフェイスの復元

ロストしたサーフェイスは、サーフェイスオブジェクトのRestoreメソッドで復元することができます。復元するといってもサーフェイスのためのメモリ領域を再確保するだけで、保持されていたビットマップデータは復元しません。また、プライマリサーフェイスとバックバッファのような複合サーフェイスの場合は、プライマリサーフェイスを復元するとバックバッファも復元されます。バックバッファを復元しようするとエラーになります。

```
lpDDSurface->Restore();    // サーフェイスの復元
```

### 課題

(1)以下のプログラムは、サーフェイスの復元を行うDDRestoreSurface関数です。関数の仕様をよく読んで完成させ、適切な場所に追加しましょう。

#### DDRestoreSurface関数

- 説明 -

DDRestoreSurface関数は、サーフェイスがロストしているかを調べ、ロストしている場合はサーフェイスメモリの復元を行います。

- パラメータ -

const DDSRFC dds...復元を行うサーフェイス。DDSRFC列挙型を指定

- 戻り値 -

サーフェイスを復元した場合はtrue, それ以外はfalseを返す。

```
/*  
*                                     サーフェイス復元                                     */  
/*****  
bool DDRestoreSurface(const DDSRFC dds)  
{  
    if(NULL == g_lpDDSurface7[dds])  
        return false;    // 復元不可  
    // サーフェイスがロストしているかを調べる  
    if(DDERR_SURFACELOST !=  ここは各自考えましょう)  
        return false;    // ロストしていない  
    // サーフェイス復元  
    if(DD_OK !=  ここは各自考えましょう) {
```

```

        OutputDebugString("*** Error - サーフェイス復元失敗(DDRestoreSurface)%n");
        return false;
    }
    return true;
}

```

(2) アプリケーションがアクティブ状態から非アクティブ状態に変更されたときに呼び出すDDSuspend関数を作成しましょう。この関数には、DirectDrawを休止するための処理を記述します。

```

/*****
/*                               DirectDraw休止                               */
*****/
void DDSuspend()
{
}

```

(3) アプリケーションが非アクティブ状態からアクティブ状態に変更されたときに呼び出すDDResume関数を作成しましょう。この関数には、休止状態のDirectDrawを復帰するための処理(サーフェイスの復元など)を記述します。

```

/*****
/*                               DirectDraw復帰                               */
*****/
void DDResume()
{
    // プライマリサーフェイス・バックバッファ復元
    ??????????????????(DDS_PRIMARY);
    // オフスクリーンサーフェイス復元
    for(int dds = DDS_OFFSCRN1; dds < DDS_MAX; dds++)
        ??????????????????(DDSRFC)dds);
}

```

(4) SuspendGame関数にDirectDrawを休止する処理(DirectDrawを休止する関数を呼び出す処理)、ResumeGame関数にDirectDrawを復帰する処理(DirectDrawを復帰する関数を呼び出す処理)を追加しましょう。