

ゲームプログラミング

DirectDraw - 第3回 協調レベルとディスプレイモード

DirectDrawの協調レベルによっては、ディスプレイを占有することができます。ディスプレイを占有すると、ディスプレイモードを自由に変更することができるようになります。状況によって解像度と色数を多くしてきれいな画像にしたり、解像度と色数を少なくして描画速度を上げるといったことが行えます。

協調レベル

協調レベルは、ディスプレイをほかのアプリケーションとどのように協調して使用するかを示すものです。協調レベルには、フルスクリーンモード(排他的占有モード)とウィンドウモード(ノーマルモード)の2つがあり、必ずどちらかに設定する必要があります。

フルスクリーンモードは、画面全体を占有するアプリケーションに最適なモードです。このモードでは、DirectDrawのほとんどの機能が使用でき、ディスプレイを完全に占有することができます。アプリケーションがアクティブ状態にある限り、ほかのアプリケーションはディスプレイに描画することはできません。また、このモードではディスプレイモードを変更することができます。

ノーマルモードは、ウィンドウで動作するアプリケーションに最適なモードです。このモードは、一般的なWindowsアプリケーションのように動作します。ディスプレイをほかのアプリケーションと共有するので、ディスプレイモードの変更やフリップができないなど、DirectDraw(およびDirect3D)の一部の機能が使用できなくなります。また、一部の機能にHELが使われる場合もあり、フルスクリーンモードよりも実行速度が劣る場合もあります。

協調レベルの設定は、DirectDrawオブジェクトのSetCooperativeLevelメソッドで行います。

SetCooperativeLevelメソッド

- 説明 -

SetCooperativeLevelメソッドは、協調レベルの設定を行います。

- パラメータ -

1つ目の引数は、メインウィンドウのハンドルです。ノーマルモードに設定する場合はNULLにします。

2つ目の引数は、協調レベルを指定するフラグです。最低でも以下のどれかを指定します。

DDSCFL_EXCLUSIVE | DDSCFL_FULLSCREEN フルスクリーンモード
DDSCFL_NORMAL ノーマルモード

複数のフラグを指定するときは、ビット論理和演算子"|"でつなげます。

- 戻り値 -

成功した場合はDD_OK、それ以外はエラーの原因をエラーコードで返します。

```
// 協調レベルの設定(フルスクリーンモード)
```

```
lpDDDraw7->SetCooperativeLevel(hWnd, DDSCFL_EXCLUSIVE | DDSCFL_FULLSCREEN);
```

ディスプレイモード

解像度と色数の組み合わせを、ディスプレイモードと呼びます。ディスプレイモードを表現するときは、x解像度、y解像度、色数の3つの値で記述します。たとえば、640×480×256(または640×480×8)は、画面の幅が640ピクセル、画面の高さが480ピクセル、色数が256色であることを示します。

サポートされるディスプレイモードは、グラフィックカードによってあらかじめ選択肢が決まっています。特殊なディスプレイモードを使用すると、特定の製品でしか動作しなくなってしまいます。

ディスプレイモードに関する一般的な用語を以下に示します。

ピクセル

画面に表示される画像は、ピクセルの集合によって構成されます。画面上で個別に色を指定できる単位をピクセルと呼びます。ピクセルに設定された色が組み合わさることで、画面全体のイメージが形成されます。

ピクセルが使用しているメモリには、画面に表示する色を示す値が格納されています。画面に表示する画像は、グラフィックカード上のメモリ(VRAM: ビデオラム)に配置されます。

解像度

解像度は通常、画面の幅(x方向の解像度)と高さ(y方向の解像度)を使って表現します。解像度は、 $x \times y$ という形式で表現します。たとえば、 640×480 という解像度は、画面の幅が640ピクセル、高さが480ピクセルで、全部で307,200ピクセルがあることを示します。使用できる解像度はいくつかありますが、解像度が大きくなれば転送量が多くなり速度が低下します。できるだけ多くのピクセルを使って鮮明な画像を表示するのが理想的です。しかし実際には、見栄えをある程度保ちつつ速度も遅すぎないように、速度と解像度のバランスを考える必要があります。

色数

色数とは、画面に同時に表示できる色の数です。画面の解像度の場合と同じように、選択できる色数は限られています。一般的な色数は256色、65,535色、16,777,216色です。色数を指定するときは、ピクセルあたりのビット数(bpp:bits per pixel)を使用します。たとえば、ピクセルあたりのビット数が8ならば、使用できる色数は256色になります($2^8 = 256$)。使用する色の数が多いほど、画質は向上します。しかし、色数が増えるほど転送量も多くなるので、速度と色数のバランスを考える必要があります。

パレット

パレットには、使用する色の情報が格納されています。1つ1つの色の情報は数値として、パレットエントリに格納されます。パレットエントリに含まれている色の情報は、赤、緑、青の3つの値で構成され、画面に表示されるときにこれらの値を組み合わせた色が表示されます。パレットは、通常は256色以下の色数のときに使用します。ピクセルはパレットエントリの番号で表現されます。パレットを操作することで、さまざまな画面効果を作り出すことができます。たとえば、画像を変更することなく、パレットエントリの内容だけを変更して色を変化させることができます。

ハイカラー

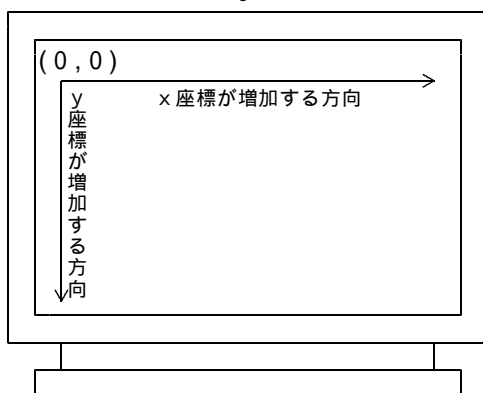
ピクセルあたりのビット数が16の色数をハイカラー(High Color)と呼びます。ハイカラーには2種類あります。ピクセルを赤5ビット、緑5ビット、青5ビットの32,768色で表現する場合と、赤5ビット、緑6ビット、青5ビットの65,535色で表現する場合があります。どちらが使われるかはグラフィックカードにより異なります。

フルカラー

色数が16,777,216色をフルカラー(Full Colorまたはトゥルーカラー:True Color)と呼びます。ピクセルは赤8ビット、緑8ビット、青8ビットで表現されます。1ピクセルは24ビットで表現することができますが、24ビットちょうどうの変数の型がないため、ダミーの8ビットを追加して32ビットになる場合があります。どちらになるかはグラフィックカードにより異なります。古い製品では24ビットが多く、最新のものはほとんどが32ビットです。32ビットの方が転送効率がよいため速度が向上しますが、メモリを多く消費してしまいます。

スクリーン座標

画面上の位置を指定するときは、スクリーン座標で指定します。スクリーン座標は、2つの値(x, y)を順番に記述して表現します。x座標は水平方向の位置を、y座標は垂直方向の位置を指定します。画面の左上角はx座標とy座標がともに0です。座標の値は右または下へ向かって増加します。x座標は画面の左から右へ向かって増加し、y座標は画面の上から下へ向かって増加します。



ディスプレイモードの設定

協調レベルがフルスクリーンモードのときはディスプレイモードを設定することができます。変更できるディスプレイモードは、グラフィックカードにより異なります。640×480や800×600といった解像度ならほとんどのグラフィックカードでサポートされています。

ディスプレイモードの設定は、DirectDrawオブジェクトのSetDisplayModeメソッドで行います。SetDisplayModeメソッド

- 説明 -

SetDisplayModeメソッドは、ディスプレイモードの設定を行います。

- パラメータ -

1つ目の引数は、x 解像度です。

2つ目の引数は、y 解像度です。

3つ目の引数は、色数でピクセルあたりのビット数で指定します。256色のときは8，ハイカラーのときは16，フルカラーのときは32または24となります。

4つ目の引数は、リフレッシュレート(1秒あたりの画面を更新する回数)です。ほとんどのグラフィックカードでは変更できないので、0にします。

5つ目の引数は、追加フラグです。現在利用できるフラグは1つしかなく、通常は0にします。

- 戻り値 -

成功した場合はDD_OK、それ以外はエラーの原因をエラーコードで返します。

```
IpDDraw7->SetDisplayMode(640, 480, 8, 0, 0);
```

SetDisplayModeメソッドで変更したディスプレイモードは、RestoreDisplayModeメソッドで元に戻すことができます。ディスプレイモードを変更した場合は、DirectDrawオブジェクトを解放する前に必ず元のモードに戻しましょう。

```
IpDDraw7->RestoreDisplayMode();
```

課題

(1)以下のプログラムを完成させて適切な場所に追加し、協調レベルの設定とディスプレイモードを設定する処理を追加しましょう。ディスプレイモードは、640×480ピクセル、フルカラーまたはハイカラーに設定しましょう。フルカラーに設定する場合は、グラフィックカードによってビット数が異なる点に注意し、32ビットでも24ビットでも正しく設定できるようにプログラムを作成しましょう。

```
// 協調レベル設定
if( ここは各自考えましょう) {
    OutputDebugString("*** Error - 協調レベル設定失敗(DDInit)%n");
    DDRelease();
    return false;
}

// ディスプレイモード設定
if( ここは各自考えましょう) {
    OutputDebugString("*** Error - ディスプレイモード設定失敗(DDInit)%n");
    DDRelease();
    return false;
}
```

(2)DDRelease関数にディスプレイモードを元に戻す処理を追加しましょう(DirectDrawオブジェクトがNULLの場合もあることを考慮しましょう)。