

ゲームプログラミング

DirectDraw - 第7回 サーフェイスの転送

サーフェイスが保持している画像の転送は、BltFastメソッドかBltメソッドで行います。これらのメソッドは、APIのBitBlt関数やStretchBlt関数よりも高速に転送することができます。

BltFastメソッド

サーフェイスオブジェクトのBltFastメソッドは、サーフェイスが保持している画像を転送します。異なるサーフェイス間でも同一のサーフェイス間でも転送できます。このメソッドでは単純な転送しか行えませんが、ドライバのサポートがない場合、Bltメソッドより10%速く転送できます。なお、クリッパーが設定されたサーフェイスへの転送はできません。

BltFastメソッド

- 説明 -

サーフェイスに保持されている画像をビットブロック転送(Bit Block Transfer)します。

- パラメータ -

1つ目の引数は、転送先のx座標です。

2つ目の引数は、転送先のy座標です。

3つ目の引数は、転送元サーフェイスオブジェクトのインタフェースを指定します。

4つ目の引数は、転送元領域を格納したRECT構造体のアドレスです。NULLだと全領域になります。

5つ目の引数は、転送フラグです。おもに以下のフラグを使用します。

| | |
|------------------------|-----------------------------------|
| DDBLTFAST_WAIT | サーフェイスが転送中の場合、転送終了か別のエラーが発生するまで待つ |
| DDBLTFAST_SRCCOLORKEY | 転送元カラーキーを使用する |
| DDBLTFAST_DESTCOLORKEY | 転送先カラーキーを使用する |

- 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。

```
lpDestSurface->BltFast(160, 50, lpSrcSurface, &rcSrc, DBDLTFAST_WAIT);
```



Bltメソッド

サーフェイスオブジェクトのBltメソッドは、サーフェイスが保持している画像を転送します。異なるサーフェイス間でも同一のサーフェイス間でも転送できます。拡大縮小、反転、回転、塗りつぶしといった追加効果を用いて転送することもできます。このメソッドのみクリッパーを設定したサーフェイスへの転送が行えます。

Bltメソッド

- 説明 -

サーフェイスに保持されている画像をビットブロック転送(Bit Block Transfer)します。転送先と転送元で領域の大きさが異なる場合、伸縮されて転送されます。

- パラメータ -

1つ目の引数は、転送先領域を格納したRECT構造体のアドレスです。NULLだと全領域になります。

2つ目の引数は、転送元サーフェイスオブジェクトのインタフェースを指定します。

3つ目の引数は、転送元領域を格納したRECT構造体のアドレスです。NULLだと全領域になります。

4つ目の引数は、転送フラグです。おもに以下のフラグを使用します。

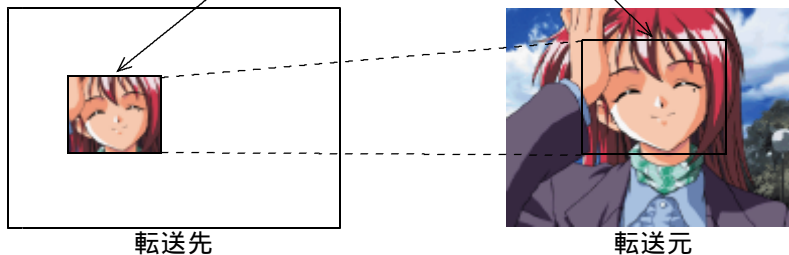
| | |
|-----------------|---------------------------------------|
| DDBLT_WAIT | サーフェイスが転送中の場合、転送終了か別のエラーが発生するまで待つ |
| DDBLT_KEYSRC | 転送元カラーキーを使用する |
| DDBLT_KEYDEST | 転送先カラーキーを使用する |
| DDBLT_COLORFILL | DDBLTFX構造体のdwFillColorメンバの色で転送先を塗りつぶす |
| DDBLT_DDFX | DDBLTFX構造体のdwDDFXメンバを使用して回転、反転を行う |
| DDBLT_ROP | DDBLTFX構造体のdwROPメンバを使用してROPを用いた転送を行う |

5つ目の引数は、エフェクトを設定したDDBLTFX構造体変数のアドレスを指定します。使用しない場合はNULL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。

```
IpDestSurface->Blit(&rcDest, IpSrcSurface, &rcSrc, DDBLT_WAIT, NULL);
```



Blitメソッドのエフェクト

Blitメソッドは、エフェクトを用いた転送ができます。エフェクトには、回転、反転、塗りつぶし、ROPなどがあります。

塗りつぶし

サーフェイスを指定した色で塗りつぶすには、DDBLTFX構造体を使用します。dwSizeメンバにDDBLTFX構造体のサイズを設定し、dwFillColorメンバに塗りつぶす色値を指定します。色値はRGBマクロの色値ではなく、サーフェイス上での色値です。

Blitメソッドの4つ目の引数でDDBLT_COLORFILLフラグを指定し、5つ目の引数にDDBLTFX構造体変数のアドレスを渡すと、サーフェイスを塗りつぶすことができます。

```
DDBLTFX ddbfx;  
ZeroMemory(&ddbfx, sizeof(ddbfx));  
ddbfx.dwSize = sizeof(ddbfx);  
ddbfx.dwFillColor = 0xffffffff;  
IpDDSOffscreen->Blit(NULL, NULL, NULL, DDBLT_COLORFILL | DDBLT_WAIT, &ddbfx)
```

回転・反転

回転や反転を行うには、DDBLTFX構造体を使用します。dwSizeメンバにDDBLTFX構造体のサイズを設定し、dwDDFXメンバに以下の効果フラグを指定します。

Blitメソッド実行時に4つ目の引数でDDBLT_DDFXフラグを指定し、5つ目の引数にDDBLTFX構造体変数のアドレスを渡すと、回転や反転を加えて転送することができます。

- 効果フラグ -

| | |
|-------------------------|---------|
| DDBLTFX_MIRRORLEFTRIGHT | 左右反転します |
| DDBLTFX_MIRRORUPDOWN | 上下反転します |

(2)以下のプログラムは、Blitメソッドで転送を行うDDBlt関数です。この関数を適切な場所に追加しましょう。

DDBlt関数

- 説明 -

DDBlt関数は、サーフェイスの画像をBlitメソッドで転送します。

- パラメータ -

const DDSRFC ddsDest...転送先サーフェイス。DDSRFC列挙体型を指定

LPRECT lprcDest...転送先領域を格納したRECT構造体のアドレス。NULLは全領域

const DDSRFC ddsSrc...転送元サーフェイス。DDSRFC列挙体型を指定

LPRECT lprcSrc...転送元領域を格納したRECT構造体のアドレス。NULLは全領域

const DWORD dwFlags...転送フラグ。おもに以下のフラグを使用

 DDBLT_WAIT サーフェイスが転送中の場合、転送終了か別のエラーが発生するまで待つ

 DDBLT_KEYSRC 転送元カラーキーを使用する

- 戻り値 -

なし

```
/*
 *                               サーフェイス転送                               *
 */
void DDBlt(const DDSRFC ddsDest, LPRECT lprcDest, const DDSRFC ddsSrc, LPRECT lprcSrc, const DWORD dwFlags)
{
#ifdef _DEBUG
    if(NULL == g_lpDDSurface7[ddsDest] || NULL == g_lpDDSurface7[ddsSrc]) {
        OutputDebugString("*** Error - 対象サーフェイス未生成(DDBlt)¥n");
        return;
    }
#endif
    g_lpDDSurface7[ddsDest]->Blit(lprcDest, g_lpDDSurface7[ddsSrc], lprcSrc, dwFlags, NULL);
}
```

(3)DDBltFast関数とDDBlt関数が正しく動作するか検証してみましょう。Test関数に以下のプログラムを追加します。

```
DDLoadFromFile(DDS_OFFSCREEN1, "BG.bmp"); // 背景読み込み
DDLoadFromFile(DDS_OFFSCREEN2, "Chara.bmp"); // キャラクター読み込み
```

TestProc関数は以下のようにします。

```
void TestProc()
{
    static int nIdx = 0; // キャラクター転送元インデックス

    // 背景転送
    DDBltFast(DDS_PRIMARY, 0, 0, DDS_OFFSCREEN1, NULL, DDBLTFast_WAIT);

    // キャラクター転送
    RECT rcChara;
    rcChara.left = 120;
    rcChara.top = 0;
    rcChara.right = rcChara.left + 120;
    rcChara.bottom = rcChara.top + 120;
    DDBltFast(DDS_PRIMARY, 0, 0, DDS_OFFSCREEN2, &rcChara, DDBLTFast_WAIT);

    Sleep(15); // フレーム制御。適当なものに書き換えてください
}
```

(4)TestProc関数を変更してキャラクターをアニメーションさせてみましょう。キャラクターは、ビットマップの位置(0, 0)から120×120のサイズで、横に5パターン用意されています。

(5)以下のプログラムは、サーフェイスの塗りつぶしを行うDDColorFill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。

DDColorFill関数

- 説明 -

DDColorFill関数は、サーフェイスの指定された領域を指定された色で塗りつぶします。

- パラメータ -

const DDSRFC ddsDest...塗りつぶしを行うサーフェイス。DDSRFC列挙体型を指定
LRECT lprcDest...塗りつぶし領域を格納したRECT構造体のアドレス。NULLは全領域
const COLORREF rgbFillColor...RGBマクロで合成された塗りつぶし色値

- 戻り値 -
なし

```
/*-----*/
/*                      塗りつぶし                      */
/*-----*/
void DDColorFill(const DDSRFC ddsDest, LRECT lprcDest, const COLORREF rgbFillColor)
{
#ifdef _DEBUG
    if(NULL == g_lpDDSurface7[ddsDest]) {
        OutputDebugString("対象サーフェイス未生成(DDColorFill)¥n");
        return;
    }
#endif

    DDBLTFX ddbfx;
    ZeroMemory(&ddbfx, sizeof(ddbfx));
    ddbfx.dwSize = sizeof(ddbfx);
    ddbfx.dwFillColor =   ここは各自考えましょう(ヒント:rgbFillColorは単純に代入しないこと)
    g_lpDDSurface7[ddsDest]->Blit(???????, ????, ????, ???????????????? | DDBLT_WAIT, ??????);
}
```