

ゲームプログラミング

DirectDraw - 第1回 DirectDraw

DirectDrawの機能呼び出すための関数を定義するDDUtils.cppと、そのヘッダDDUtils.hを作成します。

DirectDraw

DirectDrawは、グラフィックカードの制御や管理、グラフィックを描画するためのインタフェースを提供します。主な機能として以下のようなものがあります。

画面の占有とディスプレイモードの制御

ディスプレイを占有し、ほかのアプリケーションからの描画をできないようにすることができます。占有した場合は描画が高速になり、ディスプレイモード(解像度、色数、リフレッシュレート)の変更を行えます。

画像の高速転送とデバイスコンテキストのサポート

APIのBitBlt関数やStretchBlt関数よりも高速に画像を転送できます。このとき、透明色を指定して透かしたり、拡大縮小や反転などの効果を加えることもできます。また、デバイスコンテキストもサポートしているので、従来のAPIを用いた描画も行えます。

フリップ

ちらつきのない描画をするために必要となるダブルバッファリング(現在の画像を表示するための領域と、次の画像を準備するための作業領域の2つの領域を切り替えて使用すること)による描画を、ディスプレイに同期して簡単に実現できます。

パレットの管理

ディスプレイモードが8ビットモード(256色)のときは、パレットの管理を行えます。パレットを直接操作して、フェードイン、フェードアウト、フラッシュ効果などのパレットアニメーションを実現できます。

画像領域へのポインタの取得

画像を保持している領域のアドレス(ポインタ)を取得することができます。画像を直接書き換えることにより、さまざまな画像効果を行うことができます。

クリッパー

描画できる領域を指定しておき、領域外への転送をできないようにすることができます。

ガンマコントロール

ディスプレイの色を構成する赤、緑、青それぞれについて明るさの設定を行えます。この機能を使えば、赤のみを強くしたり、画面全体を暗くしたりすることが簡単に行えます。

オーバーレイ表示

つねに画面のいちばん上に表示されるようなオーバーレイを表示できます。

DirectDrawを使ったソースファイルをビルドする場合は、ヘッダファイル<ddraw.h>とライブラリ"ddraw.lib", "dxguid.lib"が必要になります。

DDUtils(DirectDraw Utilities)

DirectDrawの機能を簡単に呼び出せるように、それぞれの機能について関数化を行います。関数本体をDDUtils.cppに、DDUtils.cppの関数を呼び出すときに必要となるプロトタイプ宣言などはDDUtils.hに記述します。

また、DDUtilsで定義する関数の名前は、先頭に"DD"をつけることにします。このようにしておくと、DDUtils所属の関数であることがひと目でわかります。また、プロジェクト全体の関数をソートしたとき、DDUtilsの関数が1か所に集まるという利点もあります。

課題

以下のファイルDDUtils.cppとDDUtils.hのたりない部分を補い、プロジェクトに追加しましょう。

- DDUtils.cpp -

```
/*
=====
                        ゲームプログラミング
    Programmed by Hibikino software. Copyright (c) 2003 Hibikino software. All rights reserved.
=====
【対象OS】
    Microsoft Windows98/2000以降
【コンパイラ】
    Microsoft VisualC++ 6.0J ServicePack5
【プログラム】
    DDUtils.cpp
        DirectDrawユーティリティ
【履歴】
    * Version    0.00    2003/04/dd hh:mm:ss 初版
=====
*/

/*****
/*
                        インクルードファイル
*****/
#include "DDUtils.h"

/*****
/*
                        スタティックライブラリ
*****/

/*****
/*
                        定 数
*****/

/*****
/*
                        グローバル変数
*****/

/*****
/*
                        プロトタイプ(プライベート)
*****/

/*****
/*
                        DirectDraw初期化
*****/
bool DDInit(const HWND hWnd)
{
    DDRelease();

    return true;
}

/*****
/*
                        DirectDraw解放
*****/
void DDRelease()
{
}
```

- DDUtils.h -

```
/*
=====
                        ゲームプログラミング
Programmed by Hibikino software. Copyright (c) 2003 Hibikino software. All rights reserved.
=====
【対象OS】
Microsoft Windows98/2000以降

【コンパイラ】
Microsoft VisualC++ 6.0J ServicePack5

【プログラム】
DDUtils.h
DirectDrawユーティリティヘッダ

【履歴】
* Version    0.00    2003/04/dd hh:mm:ss 初版
=====
*/

#pragma once

/*****
/*
                        インクルードファイル
*****/

/*****
/*
                        列挙体定義
*****/

/*****
/*
                        プロトタイプ
*****/
```

DDUtils関数解説

DDInit関数

- 説明 -

DDInit関数は、DirectDrawの初期化処理を行い、DirectDrawを使用できるようにします。

- パラメータ -

const HWND hWnd...ウィンドウのハンドル。DirectDrawの初期化処理で必要になります。

- 戻り値 -

すべての初期化処理に成功した場合はtrue、それ以外はfalseを返します。

DDRelease関数

- 説明 -

DDRelease関数は、DirectDrawが使用した資源をすべて解放します。

- パラメータ -

なし

- 戻り値 -

なし

DDUtils.cppとDDUtils.hが完成したら、「DirectDrawの初期化」と「DirectDrawの解放」を行うように(これらの関数が呼び出されるように)、あるソースファイルに変更を加えます。以下の - 追加1 - と

- 追加2 - を適切な場所に追加しましょう。

- 追加1 -

```
if(false == DDInit(hWnd))
    return false;
```

- 追加2 -

```
DDRelease();
```