

# ソフトゼミ 第 1 回 VisualC++・DX ライブラリの導入

## ■ なにこれ

ソフトゼミ (裏 B、もしくは横 B)では、ゼミに引き続き、本編が終わった人向けの情報を提供します。

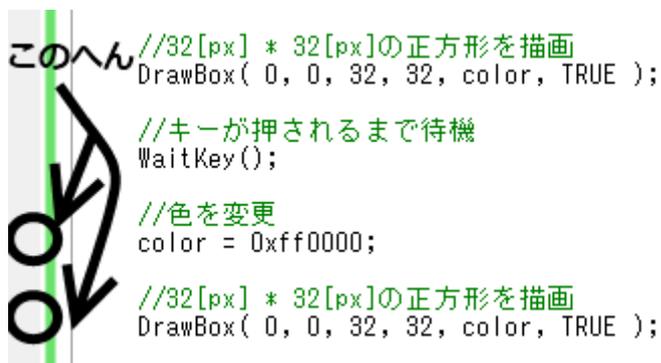
## ■ ブレークポイント

本編で書いたプログラムを改造して、以下のようにしてください。

```
#include "Dxlib.h"
int WINAPI WinMain( /*中略*/){
    //色を覚える変数 0x から始まるのは 16 進数表記
    int color = 0xffffffff;
    // ウィンドウモードにする
    ChangeWindowMode( TRUE );
    // 解像度とカラービット数を設定
    SetGraphMode( 640, 480, 32 );
    // DX ライブラリの初期化に失敗すると、即終了
    if( DxLib_Init() == -1 ){ return -1; }
    //32[px] * 32[px]の正方形を描画
    DrawBox( 0, 0, 32, 32, color, TRUE );
    //キーが押されるまで待機
    WaitKey();
    //色を変更
    color = 0xff0000;
    //32[px] * 32[px]の正方形を描画
    DrawBox( 0, 0, 32, 32, color, TRUE );
    //キーが押されるまで待機
    WaitKey();
    //DX ライブラリを終了する
    DxLib_End();
    return 0;
}
```

改造 or 新規に記述する部分は、灰色で塗った部分です。まずは普通に実行してみましよう。白い正方形が 32\*32 で表示された後、何らかのキーを押すもしくは画面をクリックすると赤くなります。理由は言うまでもなく、色を 0xff0000(赤) にして DrawBox にしたからですが、ここからが本題です。

VC++の以下の部分(color への代入文の行と、その下の DrawBox の行)をクリックしてみてください。



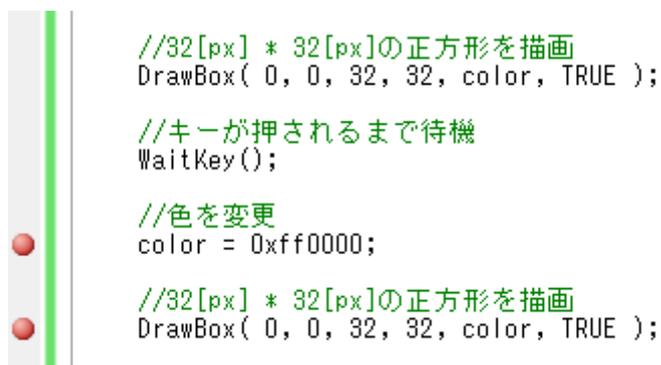
```
このへん //32[px] * 32[px]の正方形を描画
DrawBox( 0, 0, 32, 32, color, TRUE );

//キーが押されるまで待機
WaitKey();

//色を変更
color = 0xff0000;

//32[px] * 32[px]の正方形を描画
DrawBox( 0, 0, 32, 32, color, TRUE );
```

すると、



```
//32[px] * 32[px]の正方形を描画
DrawBox( 0, 0, 32, 32, color, TRUE );

//キーが押されるまで待機
WaitKey();

//色を変更
color = 0xff0000;

//32[px] * 32[px]の正方形を描画
DrawBox( 0, 0, 32, 32, color, TRUE );
```

何やら赤い出っ張りが出てきました。この状態で実行してみましよう。すると、赤い正方形が出てこようとする前に、画面が引っ込んでしまいます。さっきの赤い出っ張りには、黄色い矢印がありますね。実は、ここでプログラムが一時的に中断しているのです。

ここで注目すべきは、画面下の「自動変数」の欄です。名前: color, 値: 16777215(==0xfffff), 型 int となっています。このように、プログラムを止めている間に、その時点での変数の中身を見ることができます。再び動かすには、上の方にある緑の三角形ボタン(2つあるので、下にある方)を押すと再開します。(ちなみに、すぐ隣の隣にある青い四角のボタンを押すとプログラムを終了できます。)すると、また止まると思いますが。すると、変数の値が書き換わっているのが見えますね?そして、また動かすと赤い正方形が出てくると思いますが。

このように、この赤い出っ張りには、その行を実行する直前でプログラムを一時的に止める効果があり、止めている間、今変数がどうなっているかを見ることができます。この赤い出っ張りを「ブレークポイント」と言います。原因不明のバグがあった時には、要所

要所でこのブレークポイントを使って、プログラムを止めてみると何かわかるかもしれませんね。

なお、ブレークポイントを削除する時には、ブレークポイントの上でクリックします。