

色の合成

DxLibには、色の合成を簡単に処理してくれる、
SetDrawBlendMode という関数があります。

宣言 `int SetDrawBlendMode(int BlendMode , int Pal) ;`

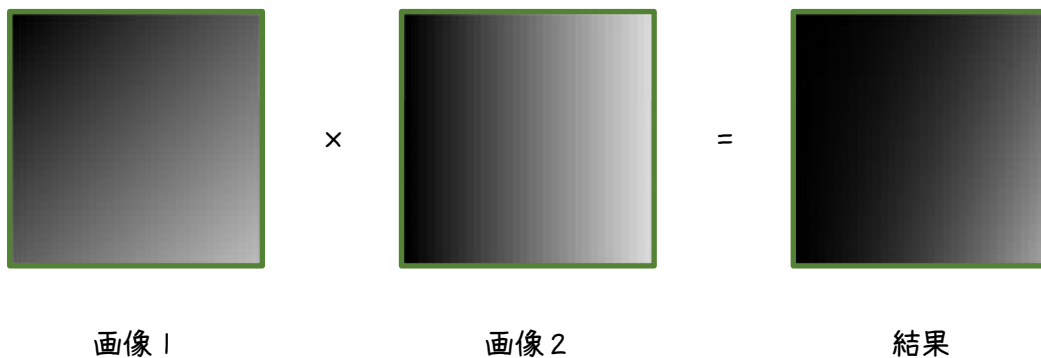
概略 描画の際のブレンドモードをセットする

| | |
|--------------------------------------|-------------------------------------|
| <code>DX_BLENDMODE_NOBLEND</code> | : ノーブレンド（デフォルト） |
| <code>DX_BLENDMODE_ALPHA</code> | : α ブレンド |
| <code>DX_BLENDMODE_ADD</code> | : 加算ブレンド |
| <code>DX_BLENDMODE_SUB</code> | : 減算ブレンド |
| <code>DX_BLENDMODE_MULA</code> | : 乗算ブレンド |
| <code>DX_BLENDMODE_INVSRC</code> | : 反転ブレンド |
| <code>DX_BLENDMODE_PMA_ALPHA</code> | : 乗算済み α 用の α ブレンド |
| <code>DX_BLENDMODE_PMA_ADD</code> | : 乗算済み α 用の加算ブレンド |
| <code>DX_BLENDMODE_PMA_SUB</code> | : 乗算済み α 用の減算ブレンド |
| <code>DX_BLENDMODE_PMA_INVSRC</code> | : 乗算済み α 用の反転ブレンド |

関数の詳細は、DxLibのホームページを、
ゲームへの応用方法には、https://dixq.net/g/02_04.html
こちらのサイトなどが参考になると思います。

今回は、荷物が荷物置き場に設置されたことが、
ユーザーにキチンと伝わるような演出の1つとして、
簡単に使用していきますので、
合成の理屈と触りだけ、解説していきます。

色の乗算



乗算なので、色のかけ算になります。
合成の結果は、画像1・2いずれの色よりも暗くなっています。

DxLibのGetColorという関数がありますが、

GetColor(0, 0, 0) ⇒ 真っ黒

GetColor(255, 255, 255) ⇒ 真っ白

RGBを引数に指定します。

0が黒、255が白、これをまず覚えてください。

色の真ん中あたりの128を合成する場合、

“かけ算”なので、 $128 * 128 = 16384$ ・・・超真っ白、ではないのです。

$((128 / 255) * (128 / 255)) * 255$

$= (0.5019 * 0.5019) * 255 = 64.3$

元の色、128よりも値が小さくなって、『黒』に近づいているのがわかります。

こういった風に、色を数値化して、諸々計算していった、

色の表現を豊かにしていています。

色の加算

色の加算は、光を表現するエフェクトに、よく使用されます。

