

2Dでのグレア処理

参考

- [元ネタ](#)
- [DirectXゲームグラフィックスプログラミング](#)

グレア処理とは？

グレアとは「きらきらする光」で、リアルタイムに物体の明るい部分を取り出して、それを、ぴかー、と強調する処理のことです。もともと3Dの技術ですが、それを擬似的に2Dで実装してしまおう、というのがこのページの狙いです。

グレア処理の手順

手順としてはこんな感じ。

1. 明るい部分（輝度）を抽出
2. 抽出した輝度を、ぼかしたか十字型のグレアにそれっぽく変形
3. 物体にグレアをレンダリング

2Dでのグレア処理の手順

輝度の抽出

3Dだとスペキュラー（強い光）をMAXにして抽出するらしいのですが、2Dだとそんなことはできないので、白塗り画像を減算合成して取り出します。

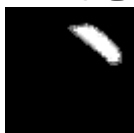
例えば、こんな物体があったとして、



この白塗り画像を減算合成してやると、



こんな感じで抽出できます。



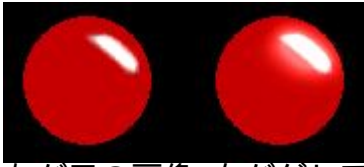
グレアに変形

今回は簡単なぼかしで、先ほど抽出した輝度を、ランダムな座標に加算合成してやります。



レンダリング

もとのボール画像と、グレアを加算合成してやれば、
ほら！ぴかー、という感じに！



左が元の画像、右がグレア処理したもの。

(、、まあ、元画像がいまいちなので、あんまりカッコよくないですが(・`´) ;

HSPでのソース

```
#const BALL_SIZE 64 // ボール画像のサイズ

// ボール画像バッファ
buffer 2
picload "ball.bmp"

// 輝度フィルタ用バッファ (HSPはデフォルトで白塗り)
buffer 3, BALL_SIZE, BALL_SIZE

// 輝度フィルタ書き出し用バッファ
buffer 4, BALL_SIZE, BALL_SIZE
gcopy 2, 0, 0, BALL_SIZE, BALL_SIZE

gmode 6, BALL_SIZE, BALL_SIZE, 238 // 白塗り画像を減算合成して、
gcopy 3, 0, 0, BALL_SIZE, BALL_SIZE // 輝度抽出

gmode 5, BALL_SIZE, BALL_SIZE, 64
// ぼかし
repeat 20
    x = rnd(10)-5
    y = rnd(10)-5
    pos x, y
    gcopy 4, 0, 0, BALL_SIZE, BALL_SIZE
loop

gsel 0
i = 0
repeat
    redraw 0
    color 0, 0, 0
    boxf 0, 0, 640, 480
    // 元画像 (比較のために)
    pos 32, 128
    gmode 2
    gcopy 2, 0, 0, BALL_SIZE, BALL_SIZE
    // グレアする画像
    pos 128, 128
```

```
    gmode 2
    gcopy 2, 0, 0, BALL_SIZE, BALL_SIZE
    i += 2
    if i > 255 {
        i -= 255
    }
    gmode 5, BALL_SIZE, BALL_SIZE, abs(-128+i) // グレア加算合成
    gcopy 4, 0, 0, BALL_SIZE, BALL_SIZE
    redraw 1
    wait 1
loop
```

課題

- 元画像がいまいちだときれいなグレアになってくれない
- グレア変形で、何度もイメージ転送するのが重い

感想

- 2Dでも、加算・減算合成ができれば、グレア処理ができるぞ。シゴ-
- 輝度を抽出するには、「黒塗りの画像 加算合成」しても同じことだということに気がつきました。