

# aardio 范例: GDI+调色矩阵演示

```
//调色矩阵
import win.ui;
/*DSG{ */
var winform = win.form(text="GDI+调色矩阵演示";right=767;bottom=351;bgcolor=16579828)
winform.add(
btnGray={cls="button";text="去色";left=608;top=240;right=744;bottom=280;db=1;dr=1;z=5};
btnSetAlpha={cls="button";text="修改透明度";left=608;top=144;right=744;bottom=184;db=1;dr=1;z=1};
button={cls="button";text="反相";left=608;top=192;right=744;bottom=232;db=1;dr=1;z=4};
plus={cls="plus";left=184;top=56;right=528;bottom=232;background="\.gdip.jpg";clipBk=false;db=1;dl=1;dr=1;dt=1;z=3};
plus2={cls="plus";left=16;top=152;right=432;bottom=312;background="\.gdip.jpg";z=2}
)
/*} */

import gdip.imageAttributes;
winform.btnSetAlpha.oncommand = function(id,event){
    winform.plus.onDrawBackground = function(graphics,rc,backgroundColor,color){

        var attr = gdip.imageAttributes();

        /*
        把图像的所有像素看作二维数组,
        每个像素为一行, 包含R,G,B,A,W 5列, 其中W为虚拟列 (值总是255)

        下面的调色矩阵必须为5行5列, 格式为:
        {
            rr;gr;br;ar;wr;
            rg;gg;bg;ag;wg;
            rb;gb;bb;ab;wb;
            ra;ga;ba;aa;wa;
            rw;gw;bw;aw;ww;
        }

        原图原图矩阵里0为最小值, 255为最大值,
        调色矩阵里0为最小值, 1为最大值, 小数可理解为百分比.

        绘图时使用原图矩阵与调色矩阵行列相乘,
        原图中的i行乘以调色矩阵中的j列, 也就等于结果图像矩阵矩阵的i行j列,

        设有图像数据如下
        {
            R1;G1;B1;A1; //第一个像素的R,G,B,A分量
            R2;G2;B2;A2; //第二个像素的R,G,B,A分量
        }

        那么经过矩阵转换以后得到的结果就是
        R1 = R1*rr+G1*rg+B1*rb+A1*ra+W*rw (原图R,G,B,A,W逐个与调色矩阵第1列相乘)
        G1 = R1*gr+G1*gg+B1*gb+A1*ga+W*gw (原图R,G,B,A,W逐个与调色矩阵第2列相乘)
        B1 = R1*br+G1*bg+B1*bb+A1*ba+W*bw (原图R,G,B,A,W逐个与调色矩阵第3列相乘)
        A1 = R1*ar+G1*ag+B1*ab+A1*aa+W*aw (原图R,G,B,A,W逐个与调色矩阵第4列相乘)

        R2 = R2*rr+G2*rg+B2*rb+A2*ra+W*rw (原图R,G,B,A,W逐个与调色矩阵第1列相乘)
        G2 = R2*gr+G2*gg+B2*gb+A2*ga+W*gw (原图R,G,B,A,W逐个与调色矩阵第2列相乘)
        B2 = R2*br+G2*bg+B2*bb+A2*ba+W*bw (原图R,G,B,A,W逐个与调色矩阵第3列相乘)
        A2 = R2*ar+G2*ag+B2*ab+A2*aa+W*aw (原图R,G,B,A,W逐个与调色矩阵第4列相乘)
        */
        attr.setColorMatrix({
            1;0;0;0;0;
            0;1;0;0;0;
            0;0;1;0;0;
            0;0;0;0.6;0;
            0;0;0;0;1
        })

        /*

        上面的矩阵运算后R,G,B值不变, 但透明度A的分量变为了 A * 0.6, 也就是改变了透明度
        */

        graphics.drawImageStretch(winform.plus.background,rc,attr);
        return true;
    }
    winform.plus.predraw()
    winform.redraw()
}

winform.button.oncommand = function(id,event){
    winform.plus.onDrawBackground = null;

    var attr = gdip.imageAttributes();
```

```

attr.setColorMatrix({
    -1;0;0;0;0;
    0;-1;0;0;0;
    0;0;-1;0;0;
    0;0;0;1;0;
    1;1;1;0;1
})

winform.plus.background = winform.plus.background.copy(attr);

winform.plus.predraw()
winform.redraw()
}

winform.btnGray.oncommand = function(id,event){
    winform.plus.onDrawBackground = function(graphics,rc,backgroundColor,color){

        var attr = gdip.imageAttributes();
        attr.setColorMatrix({
            0.213;0.213;0.213;0;0;
            0.715;0.715;0.715;0;0;
            0.072;0.072;0.072;0;0;
            0;0;0;1;0;
            0;0;0;0;0
        })

        graphics.drawImageStretch(winform.plus.background,rc,attr);
        return true;
    }
    winform.plus.predraw()
    winform.redraw()
}

winform.show()
win.loopMessage();

```

[Markdown 格式](#)