Client-side function

GetColorFilter

This function is used to get the values of color filtering.

> **Tip:** Normally the game is adding these colors to a screen to simulate weather effects. Sometimes it can be important to disable these effects. You can get rid of the effects by calling setColorFilter with zero values.

Syntax

```
int, int, int, int, int, int, int, int getColorFilter ( bool isOriginal )
```

Required Arguments

- **isOriginal:** A bool indicates if the return values of color filter are GTASA original or changed by setColorFilter. If this is set to *false,* the return values would be the color filter that is currently being used.

Returns

Returns 8 *integers,* of which the first 4 indicate the color (R,G,B,A) of color filter A, and the last 4 indicate the color (R,G,B,A) of color filter B.

Example 1

This example corrects color of dxDrawMaterialLine3D. But this method has some limit.

```
local testRT = dxCreateRenderTarget(32,32,true)

x,y,z = 0, 0, 4
size = 4
addEventHandler("onClientRender", root, function()
        dxSetRenderTarget(testRT,true)
        dxDrawRectangle(0,0,32,32,tocolor(255,255,255,255))
        dxSetRenderTarget()

        local aR,aG,aB,aA,bR,bG,bB,bA = getColorFilter(false)                                                    --Get current
 color filter
        local cR,cG,cB = 127/255+(aR*aA+bR*bA)/65535*0.5, 127/255+(aG*aA+bG*bA)/65535*0.5, 127/255+(aB*aA+bB*bA)/65535*0.5
        --Calculate the result color of color filter
        dxDrawMaterialLine3D(x+size, y+size, z-0.95, x-size, y-size, z-0.95,false, testRT, size*2,tocolor(127, 127, 127, 255)
)
        dxDrawMaterialLine3D(x+size+20, y+size, z-0.95, x-size+20, y-size, z-0.95,false, testRT, size*2,tocolor(127/cR, 127/c
G, 127/cB, 255))
end)
```

Example 2

This example corrects color of dxDrawMaterialLine3D using shader

```
local shader = [[
float3 colorFilter = float3(1,1,1);
texture sourceTexture;

sampler2D SamplerTex = sampler_state{
    Texture = sourceTexture;
    MipFilter = Linear;
    MinFilter = Linear;
    MagFilter = Linear;
    AddressU = Mirror;
    AddressV = Mirror;
};

float4 colorFilterRemover(float4 color:COLOR0, float2 UV:TEXCOORD0) : COLOR0{
        color *= tex2D(SamplerTex, UV);
        color.rgb /= colorFilter;
        return color;
}

technique cFilterRemover{
        pass P0{
                PixelShader = compile ps_2_0 colorFilterRemover();
        }
```

```
}
]]

local cFilterRemover = dxCreateShader(shader)
local testRT = dxCreateRenderTarget(32,32,true)
dxSetShaderValue(cFilterRemover,"sourceTexture",testRT)

x,y,z = 0, 0, 4
size = 4
addEventHandler("onClientRender", root, function()
        dxSetRenderTarget(testRT,true)
        dxDrawRectangle(0,0,32,32,tocolor(255,255,255,255))
        dxSetRenderTarget()

        local aR,aG,aB,aA,bR,bG,bB,bA = getColorFilter(false)                                                   --Get current
 color filter
        local cR,cG,cB = 127+(aR*aA+bR*bA)/255*0.5, 127+(aG*aA+bG*bA)/255*0.5, 127+(aB*aA+bB*bA)/255*0.5        --Calculate t
he result color of color filter
        dxSetShaderValue(cFilterRemover,"colorFilter",cR/255,cG/255,cB/255)                                     --Apply to th
e color filter remover shader
        dxDrawMaterialLine3D(x+size, y+size, z-0.95, x-size, y-size, z-0.95,false, cFilterRemover, size*2,tocolor(127, 127, 1
27, 255))
        dxDrawMaterialLine3D(x+size+20, y+size, z-0.95, x-size+20, y-size, z-0.95,false, testRT, size*2,tocolor(127, 127, 127
, 255))
end)
```

Requirements

Minimum server version	**n/a**
Minimum client version	**1.6.0-9.22188**

Note: Using this feature requires the resource to have the above minimum version declared in the meta.xml **<min_nrp_version>** section. *e.g. <min_nrp_version client="1.6.0-9.22188" />*