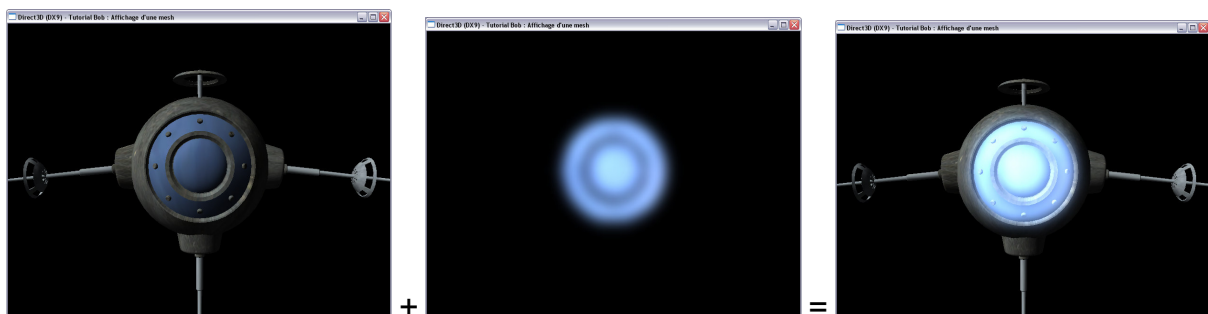


Tutorial 2 : Glow

Description : L'effet de **GLOW** est pour rendre le halo de lumière entourant une source.

Nous allons voir comment réaliser cet effet en shader avec l'aide du langage Cg. Tout d'abord, il vous faut avoir des connaissances en DirectX et quelques notions sur les shader pour comprendre ce tutorial.

Explication : Pour rendre cet effet il nous faut comprendre la technique... Afin de rendre un halo sur un objet de la scène, il faut tout d'abord le rendre dans une texture ensuite créer un flou sur cette texture et enfin l'appliquer sur un polygone qui va prendre tout l'écran.



Voici le VertexShader à appliquer :

```
struct vertexin
{
    float4 Position      : POSITION;
    float2 TexCoord0     : TEXCOORD0;
};

struct vertexout
{
    float4 HPosition     : POSITION;
    float2 TexCoord0     : TEXCOORD0;
    float2 TexCoord1     : TEXCOORD1;
    float2 TexCoord2     : TEXCOORD2;
    float2 TexCoord3     : TEXCOORD3;
};

vertexout main( vertexin IN,
                uniform float4x4 WorldViewProj,
                uniform float2 T0_Offset,
                uniform float2 T1_Offset,
                uniform float2 T2_Offset,
                uniform float2 T3_Offset )
{
    vertexout OUT;

    OUT.HPosition = mul( IN.Position, WorldViewProj );
    OUT.HPosition.w = 1.0;

    OUT.TexCoord0 = IN.TexCoord0 + T0_Offset;
```

```
    OUT.TexCoord1 = IN.TexCoord0 + T1_Offset;
    OUT.TexCoord2 = IN.TexCoord0 + T2_Offset;
    OUT.TexCoord3 = IN.TexCoord0 + T3_Offset;

    return( OUT );
}
```

Et le PixelSader :

```
struct v2f
{
    float2 TexCoord0 : TEXCOORD0;
    float2 TexCoord1 : TEXCOORD1;
    float2 TexCoord2 : TEXCOORD2;
    float2 TexCoord3 : TEXCOORD3;
};

float4 mad(float f_mult1, float4 f_mult2, float4 f_add )
{
    return (f_mult1*f_mult2)+f_add;
}

fragout main(    v2f IN,
                uniform sampler2D Texture0,
                uniform float4 Weight )
{
    fragout OUT;

    float4 t0 = tex2D( Texture0, IN.TexCoord0 );
    float4 t1 = tex2D( Texture0, IN.TexCoord1 );
    float4 t2 = tex2D( Texture0, IN.TexCoord2 );
    float4 t3 = tex2D( Texture0, IN.TexCoord3 );

    float4 r0;
    r0 =      Weight[0] * t0;
    r0 = mad( Weight[1],  t1, r0 );
    r0 = mad( Weight[2],  t2, r0 );
    r0 = mad( Weight[3],  t3, r0 );

    OUT.col = r0;
    OUT.col.a = 1.0;
    return( OUT );
}
```

Et voici le rendu final :

