# MK Glow

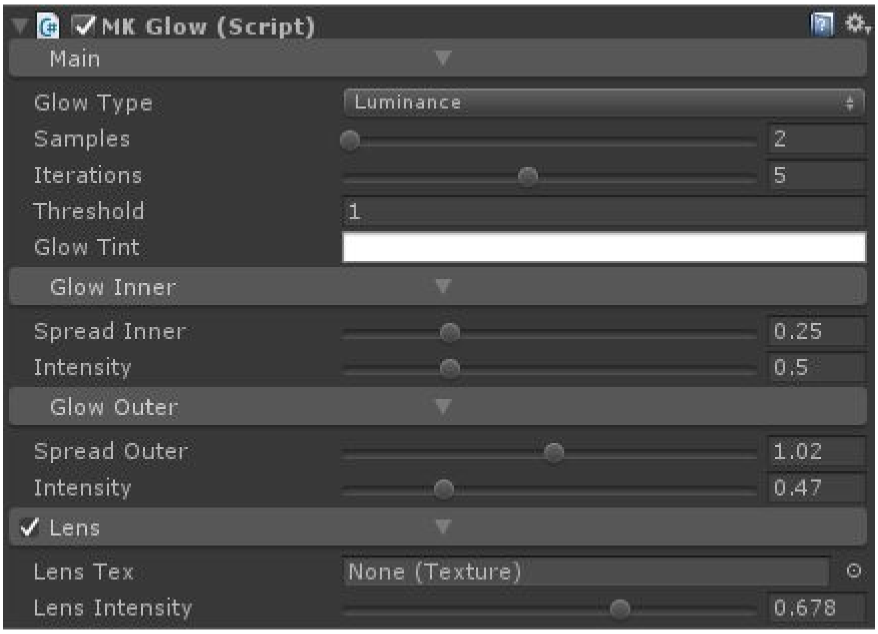
Reference

Created by Michael Kremmel
www.michaelkremmel.de

# 1.0 Setup

Before activating the MK Glow, a Camera Object has to be selected. Following this, the entry can be found here: *“Window/MKGlow/Add MK Glow To Selection”*. That’s simply it.

# 2.0 Global configuration



☑ MK Glow (Script)

Main

Glow Type: Luminance

Samples: 2

Iterations: 5

Threshold: 1

Glow Tint

Glow Inner

Spread Inner: 0.25

Intensity: 0.5

Glow Outer

Spread Outer: 1.02

Intensity: 0.47

☑ Lens

Lens Tex: None (Texture)

Lens Intensity: 0.678

| **Adjustment** | **Description** |
|---|---|
| Glow Type | Luminance: All things beyond a given threshold glow.<br>Fullscreen: The whole screen glows.<br>Selective: Only specific objects glow (needs to assign a MK/Glow/Selective shader). |
| Samples | This influences the downsampling for the blur. A higher sampling results in better performance, but also gains more flickering. |
| Iterations | Iterations for blur rendering. A Lower value results in better performance, but reduce the quality (5 should be fine in most cases). |
| Threshold | Threshold value for luminance glow (glow map extraction can be customized in MKGlowExtractLuminance shader) |
| Tint | Color tint for the glow |

| | |
|---|---|
| Spread Inner | Inner width of the glow effect |
| Spread Outer | Outer width of the glow effect |
| Inner Intensity | Intensity of the inner glow effect |
| Outer Intensity | Intensity of the outer glow effect |
| Lens Intensity | Intensity of the lens effect |
| Lens tex | Texture for the lens effect (Usage of outer glow recommend) |

# 3.0 Shader configuration (selective mode only)

The MK Glow already brings a multitude of standard shaders. These shaders can be found here: MK/Glow/Selective.

These shaders are only needed in Selective Mode! Simply assign the respective shader to the objects which shall receive the glow effect.

The shaders have some basic parameters:

| **Adjustment** | **Description** |
|---|---|
| Glow Color | The color of the glow effect on the particular object |
| Glow Power | The object’s luminous intensity |
| Glow Texture | The glow texture / the areas that should glow |
| Glow Texture Color | The color of the glow texture |
| Glow Texture Strength | The texture’s luminous intensity |

# 4.0 Make your own shaders glow (selective mode only)

As an example, we are going to fit a new created shader with a glow effect.

## 4.1 Expanding the properties box

```
Properties
{
   _MKGlowColor ("Glow Color", Color) = (1,1,1,1)
   _MKGlowPower ("Glow Power", Range(0.0,2.5)) = 1.0
   _MKGlowTex ("Glow Texture", 2D) = "black" {}
   _MKGlowTexColor ("Glow Texture Color", Color) = (1,1,1,1)
   _MKGlowTexStrength ("Glow Texture Strength ", Range(0.0,10.0)) = 1.0
}
```

The content of the properties box can simply be copy-pasted in your own shader.

## 4.2 Setting the RenderType

You also need to modify the RenderType like this:

```
SubShader
{
   Tags { "RenderType"="MKGlow"}
}
```

## 4.3 Expanding the uniform variables

Add these uniform variables to your shader

```
CGPROGRAM
   sampler2D _MKGlowTex;
   half _MKGlowTexStrength;
   fixed4 _MKGlowTexColor;
ENDCG
```

## 4.4 Expanding the Fragment function

1. Create the glow texture with the MainTexture’s texture-coordinates.
2. Now multiply the glow texture with the glow texture color.
3. Combine the created glow texture with the MainTexture

```
void surf (Input IN, inout SurfaceOutput o)
{
    fixed4 c = tex2D(_MainTex, IN.uv_MainTex) * _Color;
    fixed4 d = tex2D(_MKGlowTex, IN.uv_MainTex) * _MKGlowTexColor;
    c += (d * _MKGlowTexStrength);
    o.Albedo = c.rgb;
    o.Alpha = c.a;
}
```

## 4.5 the complete shader

```
Properties
{
    _Color ("Main Color", Color) = (1,1,1,1)
    _MainTex ("Base (RGB)", 2D) = "white" {}
    _MKGlowColor ("Glow Color", Color) = (1,1,1,1)
    _MKGlowPower ("Glow Power", Range(0.0,2.5)) = 1.0
    _MKGlowTex ("Glow Texture", 2D) = "black" {}
    _MKGlowTexColor ("Glow Texture Color", Color) = (1,1,1,1)
    _MKGlowTexStrength ("Glow Texture Strength ", Range(0.0,10.0)) = 1.0
}
SubShader
{
    Tags { "RenderType"="MKGlow"}
    LOD 200
    CGPROGRAM
        #pragma surface surf Lambert
        sampler2D _MainTex;
        fixed4 _Color;

        sampler2D _MKGlowTex;
        half _MKGlowTexStrength;
        fixed4 _MKGlowTexColor;

        struct Input
        {
            float2 uv_MainTex;
        };

        void surf (Input IN, inout SurfaceOutput o)
        {
           fixed4 c = tex2D(_MainTex, IN.uv_MainTex) * _Color;
            fixed4 d = tex2D(_MKGlowTex, IN.uv_MainTex) * _MKGlowTexColor;
            c += (d * _MKGlowTexStrength);
           o.Albedo = c.rgb;
           o.Alpha = c.a;
        }
    ENDCG
}
Fallback "Diffuse"
```

# 5.0 Scripting

All settings can be changed and adjusted during the runtime. To do so, include the library *“using MK.Glow;”* and initialize it with the class MKGlow.

The following variables are available:

- UseLens
- LensTex
- LensIntensity
- GlowLayer (selective mode only)
- GlowType
- GlowTint
- Samples
- BlurIterations
- GlowIntensityInner
- GlowIntensityOuter (not available in fullscreen)
- BlurSpreadInner
- BlurSpreadOuter (not available in fullscreen)
- Threshold (luminance mode only)

# 6.0 Bug reporting / questions / feature requests

Should there be any questions regarding the MK Toon shader or you discovered a bug, you can contact me at any time. Just send me an e-mail: support@michaelkremmel.de and I will reply as soon as possible.

Are you missing a feature or do you have great ideas to improve the shader? Feel free to contact me.