MK Glow

Reference

Created by Michael Kremmel

www.michaelkremmel.de

Contents

1. Setup

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

2. Global configuration

Glow Type: Selective = to specifically bring objects to glow, Fullscreen = complete screen glows

Blur Spread: Width of the glow effect

Blur Iterations: Number of used blurs

Samples: Significantly influences the blurs quality

Glow Intensity: The global luminous intensity

Glow Tint : The glows coloration

3. Shader configuration

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

The MK Shader is only needed in Selective Mode! Simply assign the respective shader to the objects which shall receive the glow effect.

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 (Tip: To make the whole object glow simply use the MainTexture)

Glow Texture Color: The color of the glow texture

Glow Texture Strength: The texture's luminous intensity

4. Make your own shaders glow

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

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

4.3 Expanding the CGPROGRAMM variables

```
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

```
Shader "MK/MKGlow/Normal/Diffuse"
{
        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. Scripting

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

The following commands are available:

- BlurIterations
- BlurSpread
- GlowIntensity
- GlowType
- ShowCutoutGlow
- GlowTint

6. Bug reporting / Questions

Should there be any questions regarding the MK Glow or you discovered a bug, you can contact me at any time. Just send me an E-Mail: mkremmel@gmx.de and I will reply as soon as possible.