MK Toon

Quick Start



Contents

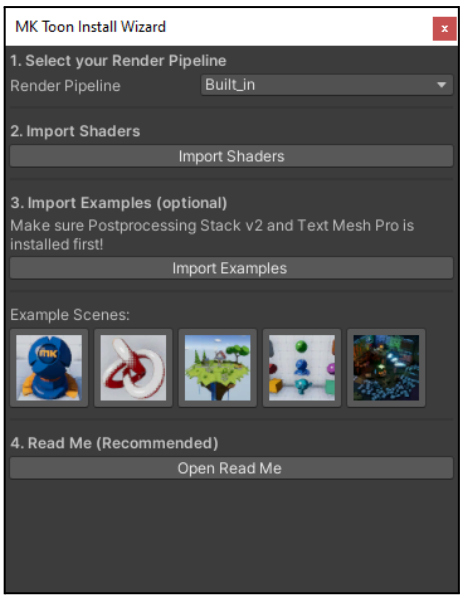
1.0 Setup / Install
- 1.1 Shaders
- 1.2 Per Object Outlines on the URP
- 1.3 How to import examples

2.0 Overview
- 2.1 Shaders
- 2.2 What are 2D ramps?
- 2.3 Built-in textures
- 2.4 Performance & optimization tips

3.0 Features
- 3.1 Options
  - 3.1.1 Workflow
  - 3.1.2 Surface
  - 3.1.3 Blending
  - 3.1.4 Alpha Clipping
- 3.2 Input
  - 3.2.1 Albedo
  - 3.2.2 Metallic
  - 3.2.3 Specular
  - 3.2.4 Smoothness
  - 3.2.5 Roughness
  - 3.2.6 Normals
  - 3.2.7 Height
  - 3.2.8 Thickness
  - 3.2.9 Occlusion
  - 3.2.10 Emission & Global Illumination
  - 3.2.11 Detail
  - 3.2.12 Detail Normals
  - 3.2.13 Tiling & Offset
- 3.3 Stylize
  - 3.3.1 Lighting
  - 3.3.2 Threshold Offset
  - 3.3.3 Gooch
  - 3.3.4 Rim
  - 3.3.5 Iridescence
  - 3.3.6 Artistic
  - 3.3.7 Color Grading
  - 3.3.8 Dissolve

3.3.9 Vertex Animation
3.4 Advanced
3.4.1 Receive Shadows
3.4.2 Wrapped Lighting
3.4.3 Diffuse
3.4.4 Specular
3.4.5 Light Transmission
3.4.6 Environment Reflection
3.4.7 Fresnel Highlights
3.4.8 Pipeline
3.4.9 Stencil
3.5 Particles
3.6 Outline
3.7 Refraction
4.0 Additional Tools
4.1 Ramp Creator
4.2 Texture Channel Packer
4.3 Mesh Utility
4.4 Material Upgrader
5.0 Scripting
6.0 Feedback / Get in touch

1.0 Setup / Install



1.1 Shaders

Right after the import the Install Wizard should open ("*Window/MK/Toon/Install Wizard*").

1. Choose your Render Pipeline (Built-in, URP).
2. Click the Import Package button.

Simply apply any shader you like to your material via "*MK/Toon/RenderPipeline/...*". That's it!

1.2 Per Object Outlines on the URP

On the Universal Render Pipeline an additional setup for the outlined shader variants is needed.

1. Select your renderer asset.
   All of your used renderer assets can be found in the renderers list on your Universal Render Pipeline Assets.
2. Click the “Add Renderer Features” button and add the “MK Toon Per Object Outlines” component.

Every renderer, which should be able to render the per object outlines of MK Toon should have the MK Toon Per Object Outlines Renderer Feature assigned.

Why is this setup required?
To utilize all of the possible optimizations of the Universal Render Pipeline additional render passes should be injected properly. Using the renderer features it’s possible to render the outlines optimized for draw calls and depth priming.
In older versions of MK Toon you had not to do this, but nowadays this is simply the most optimized way to render outlines in the URP

How to find my URP Renderer Assets?
If you create a new project using the Universal Render Pipeline the Renderer Assets are located under: “Assets/Settings/”. Based on your project setup the location of your used Renderer Assets may vary.

You can use the following two helper functions of MK Toon to find your renderer assets:

Function	Outcome
“Window/MK/Toon/Find All URP Renderer Assets in Project Window"	This shows all your renderer assets in the project window.
"Window/MK/Toon/Select Default URP Renderer Asset"	This selects your active default renderer asset.

Warning if the default renderer does not have the MK Toon Per Object Outlines Renderer Feature
By default MK Toon shows a warning on the install wizard and on the materials if your default renderer does not have the required renderer feature.

If your project is not using outlines on the default renderer and you want to disable the warning you can disable it on the MK Toon Install Wizard (Window/MK/Toon/Install Wizard) by enabling the “Disable Per Object Outlines Warning” toggle.

1.3 How to import examples

Built-in:

1. Select Render Pipeline as Built-in/URP.
2. Install Text Mesh Pro via the Package Manager.
3. Install Post-Processing Stack v2 via the Package Manager.
4. Click the Import Examples button.

URP:

1. Select Render Pipeline as URP.
2. Install Text Mesh Pro via the Package Manager.
3. Click the Import Examples button.

2.0 Overview

2.1 Shaders

MK Toon comes with 2 different basic shader setups. Standard and Particles. Each of them are offering an Unlit, Simple and Physically Based setup.

Standard → *Unlit*
→ *Simple*
→ *Physically Based*

Particles → *Unlit*
→ *Simple*
→ *Physically Based*

They inherit the following: **Unlit → Simple → Physically Based.** Every inheritance is able to use the child's features.

Unlit →	**Simple →**	**Physically Based**
Blending Culling Alpha Clipping Albedo Vertex Colors Color Grading Dissolve Vertex Animation GPU Instancing Render Priority Stencil	Isotropic Specular Normal Mapping Emission Global Illumination Custom Lighting Threshold Offset Gooch Ramp / Colors Rim Iridescence Artistic	PBS Workflow Parallax Mapping Details Advanced Diffuse Lighting Anisotropic Specular Light Transmission Fresnel Highlights

Additionally each of them are offering an outline and/or refractive variants:

Particles	**Standard**
Default Refraction	Default Refraction Outline

The Particles Shaders have some additional features:

Particles Features
Color Blending Flipbook Soft Fade Camera Fade

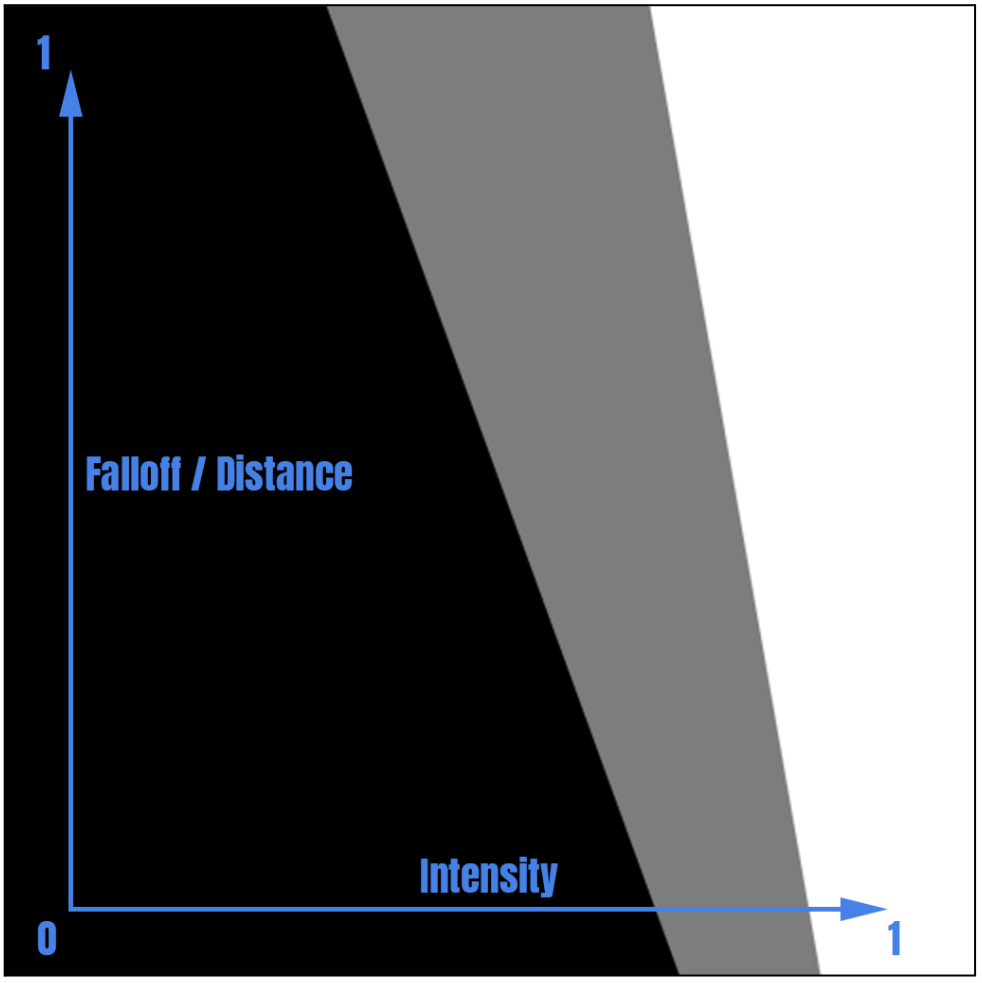
2.2 What are 2D ramps?

A common practice to modify lighting are ramp textures, which represent the **light intensity on the x-axis**. MK Toon extends this method to define the light visualization. In addition to the x-axis of the ramp the **y-axis** is used for the **light falloff of non-directional lights**. The light falloff is based on the distance to the light source.

Therefore applies the following:

1. Directional lights always have a distance of 1 (Top of the 2D ramp).
2. Non-Directional lights interpolating between 0 (close distance) and 1 (max distance).

The values of the ramps are distributed in the following way:



This allows you to modify the light intensity based on the distance to the light to create even more advanced and dynamic lighting.

If a shader property offers you to use a 2D ramp you can also simply apply a 1D ramp if you like.

2.3 Built-in textures

MK Toon comes with a bunch of built-in texture you can use for the shader features. You find the textures under: "*MKToon/Builtin/*".

Dissolve	Basic dissolve pattern.
Dissolve Border	Border color ramps for the dissolve effect.
Drawn	Light patterns for the Artistic feature in Drawn mode.
DuDv	Distortion patterns for the Refraction shaders.
Gooch Ramp 1D	Tint patterns for the Gooch Ramp feature.
Gooch Ramp 2D	Tint patterns for the Gooch Ramp feature.
Hatching	Light patterns for the Artistic feature in Hatching mode.
Iridescence	Color ramps for the Iridescence feature.
Light Ramp 1D	Light customization ramps for the Lighting in Ramp mode.
Light Ramp 2D	Light customization ramps for the Lighting in Ramp mode.
Sketch	Light patterns for the Artistic feature in Sketch mode.
Threshold	Threshold textures for the Threshold Offset feature.

2.4 Performance & optimization tips

MK Toon is engineered with performance in mind, striking a balanced trade-off between stunning visuals and efficient rendering. While exact performance can vary based on your project's specifics, the following guide will help you get the most out of MK Toon:

1. **Shader Complexity Hierarchy:**
    - **Unlit:** Fastest performance – ideal for non-dynamic lighting.
    - **Simple:** Balanced performance with enhanced visual appeal.
    - **Physically Based:** Richest visuals with a performance cost.

    Tip: Choose the simplest shader that meets your aesthetic needs to maximize performance. A good starting point in most cases is the Simple shader variant and if more precision or a physically based workflow is needed switch over to the Physically Based one.

2. **Outline Rendering:**
    - Outlines are rendered on a per-object basis, which introduces an extra draw call for each object. However, they are fully optimized and compatible with Unity's SRP Batcher.

3. **Feature Utilization:**
    - The more features you enable, the heavier the shader becomes. Evaluate your scene requirements carefully, and disable non-essential features to keep rendering efficient.

If performance is a top priority for your project and you encounter specific issues, feel free to reach out via support email.

3.0 Features

3.1 Options

3.1.1 Workflow

For physically based shaders only. Physically based workflow based on **metallic** / **smoothness** / **roughness**.



Rough to smooth physically based properties

Metallic: Material is based on metallic and smoothness properties.
Specular: Material is based on Specular and smoothness properties.
Roughness: Material is based on Metallic and roughness properties.

Each of them can produce the same result, just the input data differs.

3.1.2 Surface

The Surface setting defines how the object is rendered.

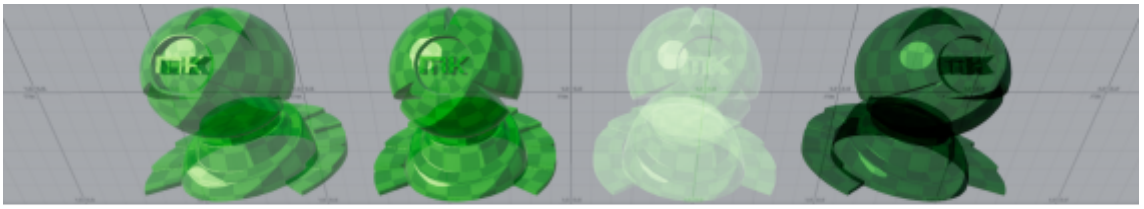


Transparent - Opaque - Transparent

Opaque: Object writes into the Depth Buffer and culls pixels behind.
Transparent: Object gets blended into the scene based on the Blending property.

3.1.3 Blending

For transparent surfaces only. Controls how the surface blends into the background.



Blending: Alpha - Premultiply - Additive - Multiply

Alpha/Default: Raw albedo alpha is used.
Premultiply: Similar to Alpha, but preserves reflections and highlights.
Additive: Blending is done additive as an extra layer.
Multiply: Surface color is multiplied into the background.
Custom: Overwrites the blending and lets you control it via two blend factors, a ZWrite and a ZTest.

Custom Blending:

Z-Write: Controls if the material writes into the depth buffer.
Z-Test: Controls how depth testing should be performed.
Src Factor: Generated color is multiplied by the Src Factor.
Dst Factor: Color that is already on the screen is multiplied by the Dst Factor.

3.1.4 Alpha Clipping

If enabled the material acts like a cutout shader.



Alpha Clipping using different blendings and settings

Alpha Cutoff: Pixels are discarded if the alpha value is lower than the threshold.

3.2 Input

3.2.1 Albedo

Albedo represents the base of your material (RGBA) using a texture and a color. If no texture is set, vertex colors will be used.

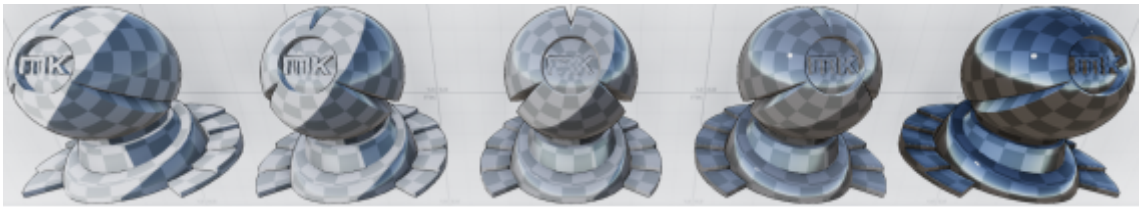


Different Albedo Maps and colors

Albedo Map: Base texture for the surface.
Albedo Color: Color gets multiplied onto the albedo map.

3.2.2 Metallic

Physically based metallic or roughness workflows only. Defines how metal-like the surface is.



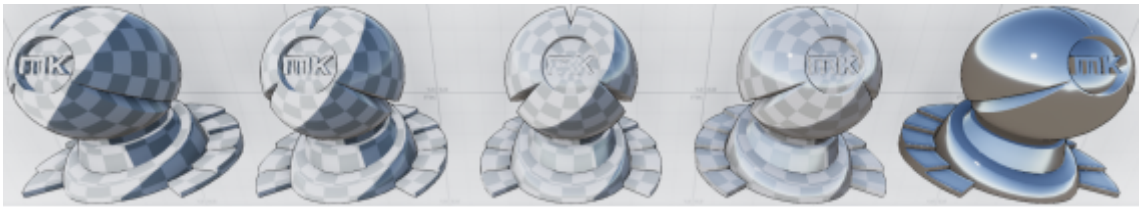
Metallic: 0 - 0.25 - 0.5 - 0.75 - 1.0

Metallic: (R) defines how metallic the surface is.
0: Fully dielectric
1: Fully metallic

On metallic workflow (A) is used to control the surface smoothness additionally.

3.2.3 Specular

Physically based specular workflow only. Controls the strength and color tint of the specular reflectivity.



Specular: 0 - 0.25 - 0.5 - 0.75 - 1.0

Specular: (RGB) defines the reflectivity of the surface.
Black: Fully dielectric.
White: Fully reflective.

(A) is used to control the surface smoothness additionally.

3.2.4 Smoothness

Physically based metallic or specular workflows only. Controls how the surface spreads the highlights.



Smoothness: 0 - 0.25 - 0.5 - 0.75 - 1.0

Smoothness: Value to control the surface smoothness.
0: Rough highlight, rough surface
1: Sharp highlight, smooth surface

3.2.5 Roughness

Physically based roughness workflow only. Controls how the surface spreads the highlights.



Roughness: 0 - 0.25 - 0.5 - 0.75 - 1.0

Roughness: (R) is used to control the surface roughness.
0: Sharp highlight, smooth surface.
1: Rough highlight, rough surface.

3.2.6 Normals

Normal mapping allows you to add more details to the surface.

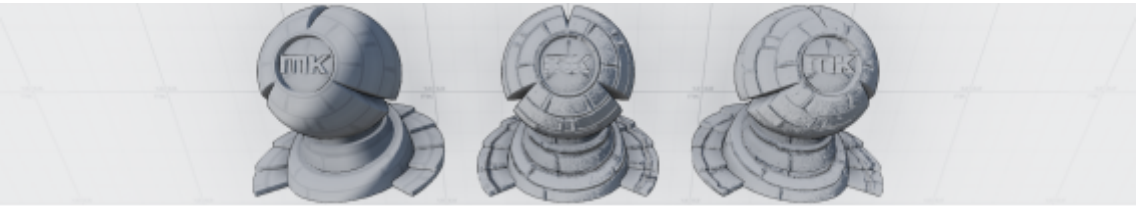


Different Normal Maps

Normal Map: Tangent Spaced Normal Map.
Intensity: Intensity of the Normal Map.

3.2.7 Height

For physically based shaders only. The Height Map is used for parallax mapping. Parallax mapping is an advanced technique of Normal Mapping, which adds the illusion of depth by modifying texture coords. To provide best rendering performance MK Toon uses only a view spaced approximation of Height Mapping.

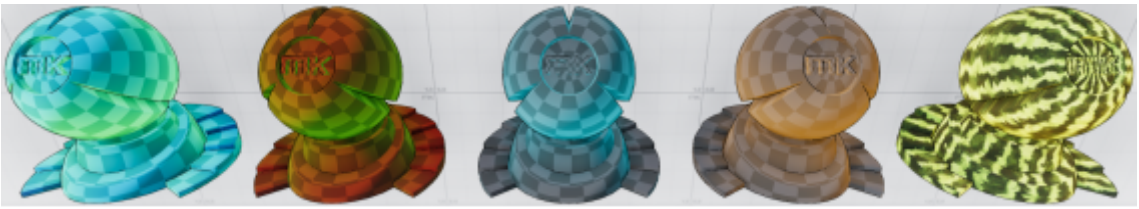


Default - Normal - Normal + Height

Height: (R) is used to control the height.
Parallax: Applies the parallax effect based on the Height Map.
0/black: No height applied.
1/white: Max height applied.

3.2.8 Thickness

For physically based shaders only. The Light Transmission is based on a thickness map, distortion and a transmitted color. It controls how much light passes through the surface. Requires the Light Transmission feature to be enabled on the advanced tab.



Different light transmission setups

Thickness: (R) controls the thickness of the surface. A higher value lets more light pass through.
0/black: thick, zero light is transmitted.
1/white: thin, light is fully transmitted.
Distortion: Distorts the light, which passes through the surface. A higher value scatters more light.
Transmission Color: Tint color for the transmitted light.

3.2.9 Occlusion

For physically based shaders only. Occlusion is used to control how much light is shown on specific areas on the surface. In MK Toon the Occlusion **differs a bit from a regular implementation**. The occlusion controls how much **indirect and direct** light is received.



Occlusion comparison

Occlusion: (R) indirect and (G) direct light received.
0/black: No lighting applied.
1/white: Default lighting applied.
Intensity: Controls how much occlusion is applied.

3.2.10 Emission & Global Illumination

Emissive materials appear as visible light sources. Emission makes the material emit light to the environment.



Emission variations

Emission: (RGB) controls which areas emit light.
0/black: No emission.
1/white: Emission is applied.
Color: (RGB) Applies a HDR color to the emission. Value is multiplied into the Emission Map if set.

Global Illumination: Controls how the emitted light affects the nearby area.
Baked: Only has effect where baked GI is enabled in the scene.
Realtime: Realtime GI if enabled in the scene.
None: Emission does not affect the surrounding area.

3.2.11 Detail

Physically based shader only. Details allowing you to apply an additional layer to the surface. This layer can be used for any kind of details. Details will be combined with the Albedo using a Blending. (A) is used for the visibility of the details.



Detail: Multiply - Mix - Add

Detail: Detail texture. (RGB) is used for the color and (A) is used for visibility.
Color: (RGB) controls the tint of the Detail Map.

Detail Blend: Controls how the detail is blend into the surface.
Mix: Albedo and detail is interpolated
Add: Details are additive added to the albedo.
Multiply: Detail is multiplied into albedo.

Detail Mix: Controls how strong the details are shown.
0: No details visible
1: Details fully visible

3.2.12 Detail Normals

Physically based shader only. Detail normals are an additional layer of normals, which are mixed into the default normals.



Different Detail Normal Maps

Normal Map: Tangent Spaced Normal Map.
Intensity: Intensity of the Detail Normal Map.

3.2.13 Tiling & Offset

The tiling and offset is split into two parts. Main and Detail. Every non-detail texture input is using the main tiling & offset as a base. Every detail texture is using the detail tiling & offset.

3.3 Stylize

3.3.1 Lighting

The appearance of light can be customized using four different styles.



Light Style: Built-in - Cel - Banded - Ramp - Ramp

Light Style:

Builtin: Lighting calculation has no custom style.

No additional properties required to set up.

Cel: Generic cel shading style. One light cut is applied to lighting.

Smoothness	Smoothness of the lighting. *0*: Sharp & local antialiased edge *1*: Smooth interpolation
Threshold	Controls the break point to split between lit and unlit areas on the surface. *0*: Material is fully lit, no shadows are visible *1*: Material is fully shadowed, no lights are visible.

Banded: A variation to the simple one. Multiple light cuts can be applied.

Bands	Controls the amount of calculated light bands.
Bands Scale	Interpolation of the light bands. A higher value will “push” the band more into shadowed areas.
Smoothness	Smoothness of the lighting. *0*: Sharp & local antialiased bands *1*: Smooth interpolation in between the bands
Threshold	Controls the break point to split between lit and unlit areas on the surface. Light bands get moved based on the threshold. *0*: Material is fully lit, less shadows are visible *1*: Material is fully shadowed, less lights are visible.

Ramp: Lighting is based on a ramp map.

Ramp Map	Lights ramps control the intensity and falloff of the shaded light. They can be either 1D or 2D. *X Axis*: Controls the intensity from left/dark to right/bright. *Y Axis*: Controls the falloff from bottom/far to top/near (non directional lights only).

3.3.2 Threshold Offset

The Threshold Offset is a special light modification. It is used to distort the light intensity to create special gradients between bright and dark areas of the surface.



Threshold Offset combined with ramp light style

Threshold: (R) controls applied light pattern. A low valued grayish color palette is recommended.
Scale: Controls the tiling of the pattern.
Offset: Controls the intensity of the effect. A higher value will distort the lighting more.

3.3.3 Gooch

The Gooch feature is used to tint the surface based on bright and dark colors. Your albedo map is the base, which gets interpolated to the bright or dark gooch map based on the light intensity. This way you can control which color tint the surface gets based on the received light. After that the resulting color is tinted using the Gooch Ramp.



Gooch variations using colors, ramp and textures

Ramp: (RGB) controls the color **tint** of the surface **based on** the **lighting condition**. This ramp does **not** affect the **lighting calculations** itself. The ramp can be either 1D or 2D.
X Axis: Controls the colors based on the intensity of the light from left/dark to right/bright.
Y Axis: Controls the colors based on the falloff of the light from bottom/far to

top/near (non directional lights only).
Bright Map: Defines the color tint for bright areas on the surface.
Bright Color: Tints the Bright Gooch Map additionally.
Dark Map: Defines the color tint for dark areas on the surface.
Dark Color: Tints the Dark Gooch Map additionally.

3.3.4 Rim

Rim Lighting is based on the fresnel effect. Outer areas of the Material will be highlighted.



Rim variations

Rim:
Off: Effect is disabled.
Default: Effect is applied to the whole surface.

Color	(RGB) defines the resulting color. Alpha is used for visibility.
Size	Controls the size of the effect.

Split: Effect is split between bright and dark areas of the surface.

Bright Color	(RGB) defines the resulting color for bright areas. Alpha is used for visibility.
Dark Color	(RGB) defines the resulting color for dark areas. Alpha is used for visibility.
Size	Controls the size of the effect.

3.3.5 Iridescence

Adds a Iridescence effect on top of the surface. Based on the viewing angle, areas will change color gradually.



Iridescence, different setups

Off: Disables the effect.
On: Enables the effect.

Ramp: Ramp (RGB) defines the iridescence color based on the viewing angle. A 1D ramp should be applied.
Color: (RGB) tints the effect additionally. (A) is used for visibility.
Size: Size of the iridescence effect.

3.3.6 Artistic

Artistic style gives your material a customization for the lighting based on input texture(s). For best control over the effect it is recommended to use a softer lighting as a base such as cel shading with some diffuse smoothness.

Off: Artistic style is disabled.

Drawn: Lighting is customized via a lookup texture (R) and two thresholds for lit and shadowed areas. Lighting will change based on the lookup texture as a repeatable pattern.



Artistic Drawn examples

Map	Lookup texture (R) to customize the lighting.
Clamp Min	Threshold to control the shadowed areas.
Clamp Max	Threshold to control the lit areas.
Map Scale	Controls the tiling of the lookup texture.

Hatching: Lighting is customized via two lookup textures (RGB). Six Map channels are used to control how the light interacts from dark to bright. The map channels are distributed the following from dark to bright: RGBRGB.



Artistic Hatching examples

Map Dark	Light pattern for dark areas. (R) => (G) => (B)
Map Bright	Light pattern for bright areas. (R) => (G) => (B)
Map Scale	Controls the tiling of the hatching maps.

Sketch: Lighting is customized via one lookup texture by interpolating smoothly into the sketch pattern.



Artistic Sketch examples

Map	Lookup texture ® to customize the lighting.
Map Scale	Controls the tiling of the lookup texture.

Artistic Projection:
The artistic customization can be projected via Tangent Space or Screen Space.
Tangent Space: Original UV coords are used.
Screen Space: Projected onto the surface based on the view angle. Gives a more flat look.

Artistic Frequency: Animates the artistic effect to give it a constantly redrawn look. A value of 1 disables the animation.

3.3.7 Color Grading

Controls how the contrast, saturation and brightness is applied.



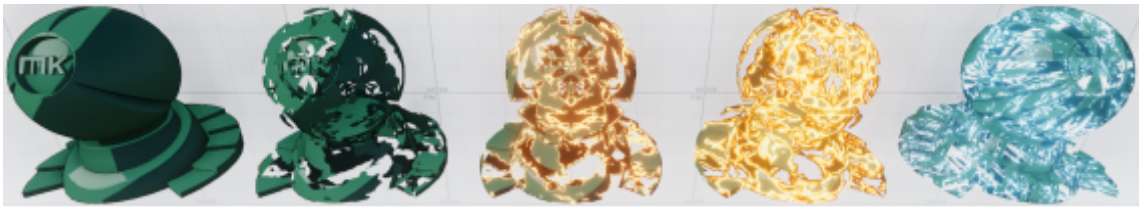
Color Grading: Off - Contrast - Saturation - Brightness - Combined

Color Grading:
Off: Effect disabled.
Albedo: Effect is applied to input albedo only.
After Everything: Effect is applied after lighting is done.

Contrast: Controls the contrast of the surface.
Saturation: Controls the saturation of the surface.
Brightness: Controls the brightness of the surface.

3.3.8 Dissolve

Dissolves the surface based on a given pattern.



Dissolve: Off - Default - Color - Ramp - Ramp

Dissolve	(R) is used for the layout of the dissolve effect.
Amount	Controls the amount of dissolving. *0*: Object is visible. *1*: Object is fully dissolved.

Dissolve:
Off: Dissolve is disabled
Default: Dissolving is done via a map and a dissolve value
Border Color: Dissolving is done via a map and a dissolve value. Additionally a border around the edges is created based on a given color.

Border Size	Size of the border around the dissolve effect.
Border Color	Tint color for the created border.

Border Ramp: Dissolving is done via a map and a dissolve value. Additionally a border around the edges is created based on a ramp texture and a tint color.

Border Size	Size of the border around the dissolve effect.
Border Color	Tint color for the created border.
Border Ramp	1D ramp (RGB) to tint the border based on its visibility.

3.3.9 Vertex Animation

The Vertex Animation offers three generic animations, which are computed on the shaders side (no additional setup required).



Animation: Sine - Sine - Pulse - Noise - Sine

Vertex Animation:
Off: Vertex Animation is disabled.
Sine: Sine based animation, applied in object space.
Pulse: Pulse styled animation.
Noise: Animated noise effect over the surface.

Intensity: Controls the intensity of the Vertex Animation. Map (R) is used for additional adjustments.

Frequency: Frequency of the animation for X, Y, Z axis.

Stutter: Applies a stuttering to the vertex animation.

3.4 Advanced

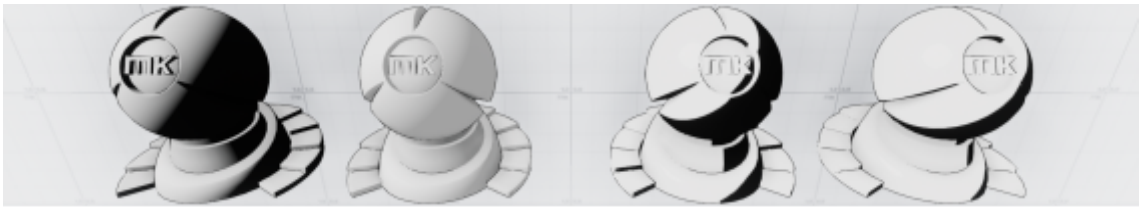
3.4.1 Receive Shadows

When enabled, other GameObjects can cast shadows onto this GameObject.

Note for Built-in Render Pipeline: You have to disable the Receive Shadows option on the Mesh Renderer. This option is disabled on the Built-in Render Pipeline.

3.4.2 Wrapped Lighting

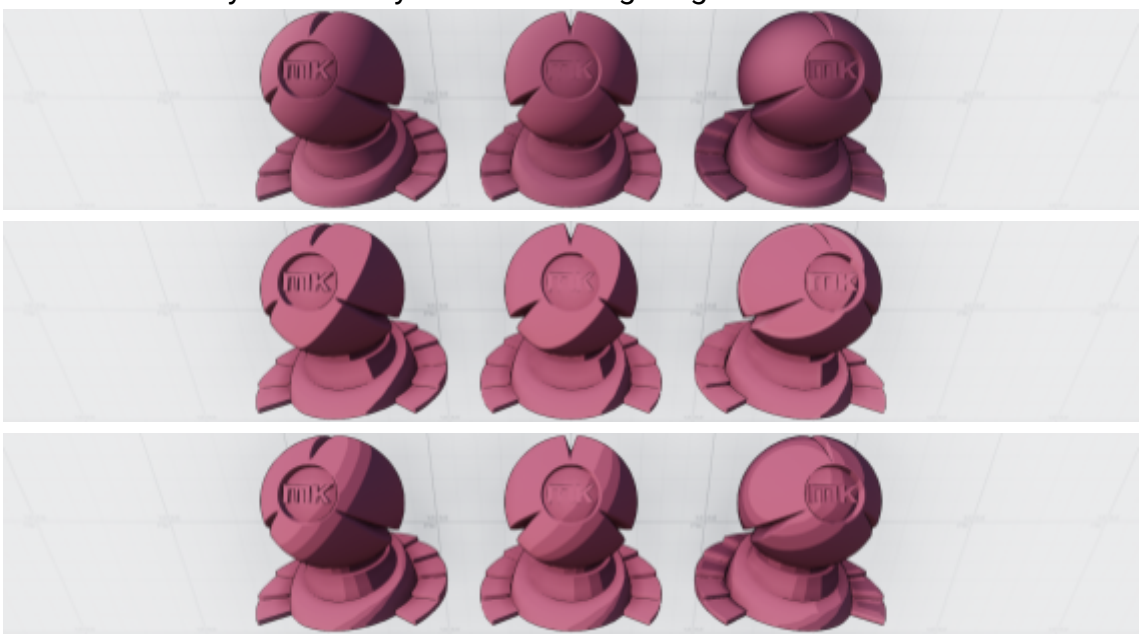
If enabled the diffuse lighting is more soft (areas get lit more easily). You may need to adjust your light threshold after enabling/disabling this setting, when using a stylized lighting.



Built-in default/wrapped - Cel default/wrapped

3.4.3 Diffuse

MK Toon offers you three styles of diffuse lighting.





Diffuse: Lambert - Oren Nayar - Minnaert

Lambert: The most simple lighting based lambertian reflectance.
Oren Nayar: A more precise lighting, which takes the roughness of the surface and the view angle into account.
Minnaert: Lighting adds some darkening limbs based on the view angle. Good for things like fabric.

Even without the matter of intended usage every diffuse style can give your model an unique style.

3.4.4 Specular

Controls how the specular affects the lighting.



Specular: Off - Iso - Iso - Aniso - Aniso

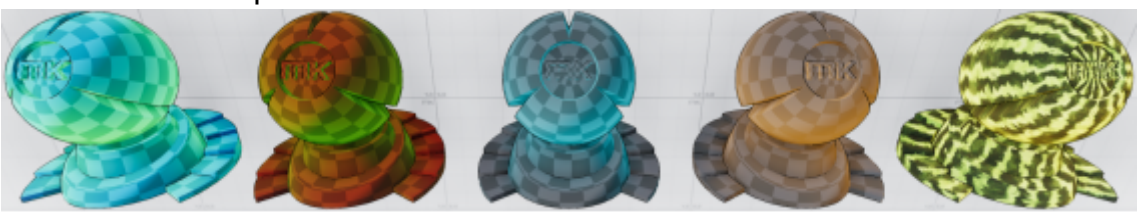
Off: Specular is disabled.
Isotropic: Round shaped specular highlight.
Anisotropic: Anisotropic stretched specular highlights. Good for Hair / brushed metal. For physically based shaders only.

Intensity: Intensity/brightness of the specular highlight. A physically correct value would be 1 (Built-in lighting style). However, in some lighting styles you want to expose your specular highlights.

Anisotropy: Anisotropic specular stretch based on the tangents at a Range between -1 and 1. Anisotropic specular only.

3.4.5 Light Transmission

For physically based shaders only. If enabled, light will pass through the object and defines how it passes through. After enabling the thickness area will be available on the input tab.



Different light transmission setups

Off: Disabled
Translucent: Light is fully transmitted and not scattered by the surface radiance.
Subsurface Scattering: Transmitted light is scattered by the radiance of the surface.

Intensity: Intensity/brightness of the transmitted light. A physically correct value would be 1 (Built-in lighting style). However, in some lighting styles you want to expose your transmitted light.

3.4.6 Environment Reflection

The Environment Reflection handles the indirect Lighting from reflections / Global Illumination / Sky.



Environment Reflection: Off - Ambient - Advanced

Off: Indirect Lighting is disabled. On physically based shaders, albedo color will be used for reflection.
Ambient: Indirect lighting only uses ambient and GI color.
Advanced: For physically based shaders only. Indirect lighting uses reflections / Sky / Ambient / GI.

3.4.7 Fresnel Highlights

For physically based shaders only. Let the light graze the surface on a shiny setup to reflect it to the camera based on a fresnel behavior.



Fresnel Highlights: Off - On - On - Off - On

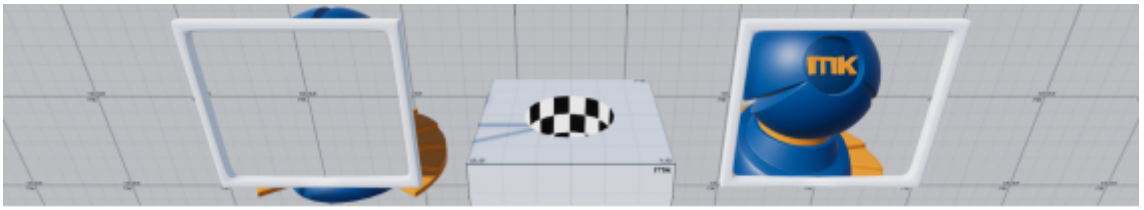
3.4.8 Pipeline

GPU Instancing: If enabled, the material uses GPU instancing to draw multiple copies of the same mesh at once. This can reduce the number of draw calls.

Double Sided GI: When enabled, the lightmapper accounts for both sides of the geometry when calculating Global Illumination.

Render Priority: Determines the chronological rendering order for a Material. High values are rendered first.

3.4.9 Stencil



Stencil: Hide - Hole - Show

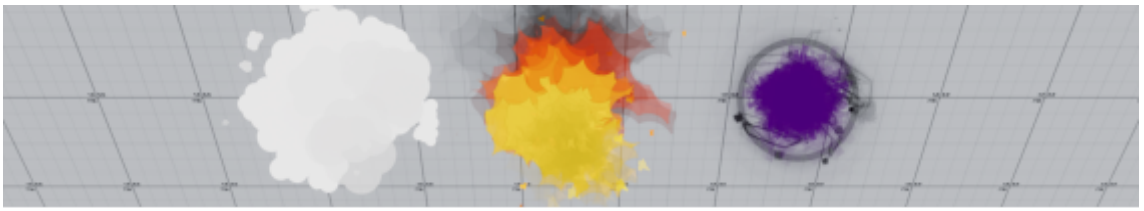
Use a custom stencil test based on the unity setup.

Builtin: Default Stencil will be used.
Custom: Define your own stencil test.

Stencil Properties:
Based on the unity documentation.

Ref	The value to be compared against (if Comp is anything else than always) and/or the value to be written to the buffer (if either Pass, Fail or ZFail is set to replace). 0–255 integer.
Read Mask	An 8 bit mask as an 0–255 integer, used when comparing the reference value with the contents of the buffer (referenceValue & readMask) comparisonFunction (stencilBufferValue & readMask). Default: 255.
Write Mask	An 8 bit mask as an 0–255 integer, used when writing to the buffer. Note that, like other write masks, it specifies which bits of stencil buffer will be affected by write (i.e. WriteMask 0 means that no bits are affected and not that 0 will be written). Default: 255.
Comp	The function used to compare the reference value to the current contents of the buffer. Default: always.
Pass	What to do with the contents of the buffer if the stencil test (and the depth test) passes. Default: keep.
Fail	Fail stencilOperation
ZFail	ZFail stencilOperation

3.5 Particles



Some particle variations

Color Blend: Defines how the particles color (RGBA) is combined with the Albedo (RGBA).

Multiply	Particles color is multiplied into the albedo.
Additive	Particles color is added to the albedo.
Subtractive	Particles color is subtracted from the albedo.
Overlay	Particles color is added to the gray values of the albedo. Similar to additive, but preserves the albedo color.
Difference	Subtracts the particles color from the albedo color, but preserves a positive value. Useful for effects with a dynamic color change.
Color	Particles color is generated using the albedo alpha and the particles input color (albedo color will be overwritten).

Flipbook: Enabled Flipbook Frame Blending. Settings have to be made on the Particle System. Blends the frames in a flip-book together in a smooth animation.

Soft Fade: Surface fades out if intersecting with other geometry in the depth buffer.

Near	The closest distance, the surface can get to the surrounding geometry before it fades from the camera's view.
Far	The farthest distance, the surface can get away from the surrounding geometry before it fades from the camera's view.

Camera Fade: Surface fades out when getting close to the camera.

Near	The closest distance, the surface can get to the camera before it fades from the camera's view.
Far	The farthest distance, the surface can get away from the

	camera before it fades from the camera's view.

3.6 Outline

Creates an outline based on set outline mode. Outlined objects are forced to render opaque due to pipeline limitations.



Outline: Off - On - Noise - On - Noise

Outline:
Hull Object: Outline is created as an inverted hull. It's the most common per object outline method.
Hull Origin: Outline is created based on the object space vertex position as an inverted hull. Your geometry needs to be centered.
Hull Clip: Similar to the Hull Object method, but it is tweaked towards pixel perfection and avoids foreshortening.

Data: Defines which data set should be used to create the hull. In most cases Normal should be fine.

Normal	Original normal of the mesh is used.
Baked	Outline is created using a different data set of normals. You can use the MK Mesh Utility to create new meshes with a modified UV7 channel or just average the mesh normals in your 3D program and save them to the UV7 channel. Baked normals can only work for static meshes.

Color: Color of the Outline.
Width: Pixel width of the outline based on a 4k resolution. Outline size is scaled into the user's used resolution.
Noise: Noise effect for the outline. Outline vertices are distorted to create an irregular look.

3.7 Refraction

Creates a refracted look for geometry behind the surface. The refraction is not physically based at all, rather approximated to maintain rendering speed.



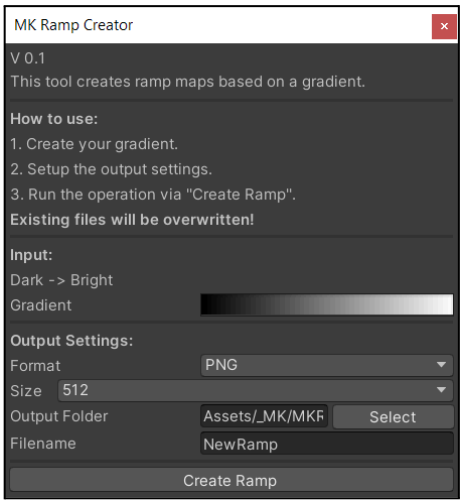
Some refraction variations

Index Of Refraction: Let the refraction bend around curved surfaces.
DuDv Map: Defines the refraction pattern. Texture has to be imported as a Normal Map. Either a Normal Map or a DuDv Map can be used.
Fade: Controls the visibility of the refraction.
0: No refraction visible.
1: Refraction fully visible.

4.0 Additional Tools

4.1 Ramp Creator

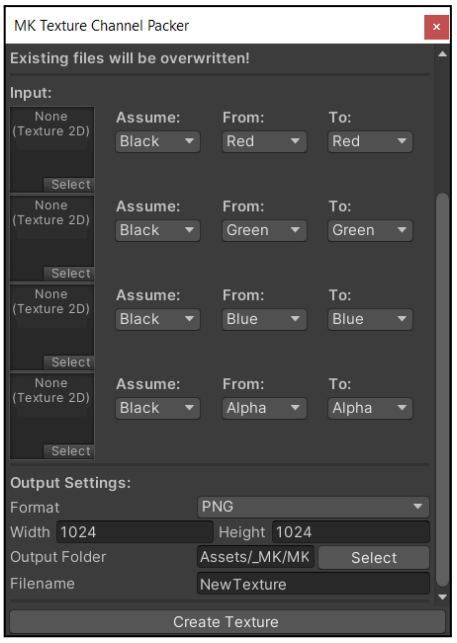
The Ramp Creator Tool ("*Window/MK/Ramp Creator*") allows you to create 1D ramp textures based on a set gradient.



Gradient: Represents the final ramp gradient. The gradient is created using the built-in Unity Gradient Tool.
Format: PNG, JPG and TGA are available. In most cases you want to avoid JPG, because it's recommended to use a lossless texture format in Unity.
Size: Ramp width in pixels.
Output Folder: Folder for your output.
Filename: Filename of the created file. Already existing files in the folder are overwritten!

4.2 Texture Channel Packer

The Texture Channel Packer (“Window/MK/Texture Channel Packer”) lets you create a new texture based on four inputs. This way you can combine different color channels of different textures into a new one.



This tool is very useful to create textures for MK Toon inputs. Every texture which is using multiple color channels can be created using this tool if you have multiple textures.

The textures should match the size, otherwise the output channels will be scaled.

Input:

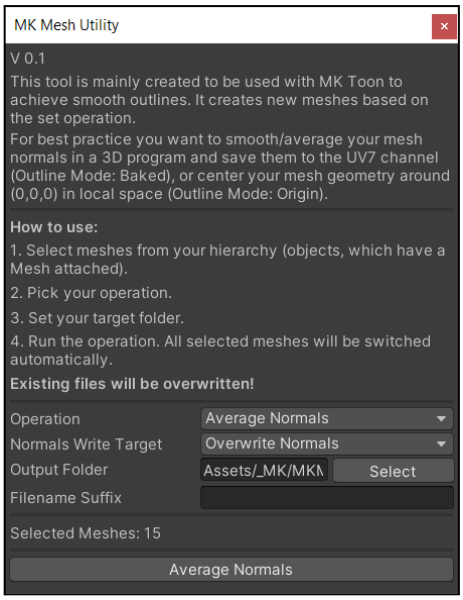
Texture: Source Texture
Assume: If no texture is set, a blank color will be used as input.
From: The color channel you want read from the texture.
To: The target color channel, which you want to pack the “From” color channel.

Output:

Format: PNG and TGA are available.
Width: Width of the output.
Height: Height of the output.
Output Folder: Folder for your output.
Filename: Filename of the created file. Already existing files in the folder are overwritten!

4.3 Mesh Utility

The Mesh Utility (“*Window/MK/Mesh Utility*”) lets you create new meshes based on existing ones of gameobjects you selected in your hierarchy.



Operation:

Average Normals: Normals get smoothed based on an average value of the vertices, which are using the same position.
Center Geometry: The geometry gets centered around (0,0,0) in local space.

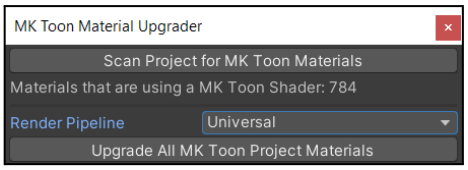
Normals Write Target:

Overwrite Normals: Overwrite the original normals of the mesh.
UV7: Normals are written onto the UV7 vertex input. This does not work with animated meshes.

Output Folder: Folder for your output.
Filename Suffix: Adds a suffix to the original mesh file name.

4.4 Material Upgrader

The Material Upgrader ("*Window/MK/Toon/Material Upgrader*") lets you upgrade materials, which are using a "*MK/Toon/Built-in*" shader from the built-in render pipeline to the LWRP or URP.



5.0 Scripting

By default MK toon offers a way to simply change variables via scripting.
Namespace: MK.Toon
Class: Properties
This static class lists every changeable property of MK Toon. Every property has the following two methods to get and set values.

<T> GetValue(material)	Get the value of a material property.
void SetValue(material, <T> value)	Set the value of a material property.

6.0 Feedback / Get in touch

Do you have some cool stuff to show?
I would love to see your results (High resolution screenshots / videos) using the shader!

Questions, bug reports, feature requests, feedback:
Feel free to get in touch via support@michaelkremmel.de.