



The Toon Shader

Documentation

Version 1.0.0
January 2025

Website: <https://www.shadercrew.com>
Email: support@shadercrew.com
Discord: <https://discord.gg/Ndx3PED7Zq>

Contents

Contents	2
Introduction	3
Overview	3
Color-Based Shading	3
Light-Based Shading	3
Toon Styling	4
Hatching	4
Halftones	4
Randomization for Both Styles	5
Outlines	5
Optimization	5
Material Replacement Tool	5
What is a Shader?	6
How to Import the Samples	6
How does it work	7
How to apply the Shader	7
Method 1 - Manually apply the shader	8
Method 2 - Automatically apply the shader	8
How to control the effect of the shader	9
Section 1 - Performance	10
Section 2 - Surface Options And Inputs	10
Section 3 - Toon Shading	11
Section 4 - Styling	11
Section 5 - Outline	13

Introduction

Overview

The Toon Shader is your ultimate toolkit for crafting awesome cartoon-style materials! Whether you're going for a bold, vibrant comic book vibe with cross-hatching, a smooth, soft animation look, or even a retro pop art style with halftones, this shader has got you covered. The best part? It's designed for *total artistic control*, so you can craft any toon style you can dream up with just the “**The Toon Shader**” shader.

Color-Based Shading

In the **Color-based** method, you get two ways to control the shading style:

- **Cell-based Shading:** This gives you a sharp, comic-book-style look with flat blocks of color. It adapts to the light direction, giving you that classic cartoon appearance. Plus, you have complete control over the core shadow color, the terminator width and smoothness, and the form shadow color, allowing you to fine-tune the shading to get exactly the look you want.
- **Gradient-based Shading:** Want more color versatility? The gradient-based approach allows you to use rich, multi-color gradients to create smooth, dynamic transitions between light and shadow. You can design gradients with a wide range of colors, adding depth and complexity while responding to light directions for that true cartoon feel.

If you prefer a simple, flat look with no shading effects, just turn off **Toon Shading**. This will give you a solid color appearance without any shading alterations.

Light-Based Shading

The **Light-based** method adds a layer of realism by interacting with scene lighting, offering two lighting models:

- **Blinn-Phong(currently URP only):** A fast and efficient lighting model that delivers simple yet effective highlights and shadows. It's perfect for real-time applications where performance is crucial.
- **Physically-Based Rendering (PBR):** Offers accurate, advanced, and dynamic lighting by simulating real-world light interactions for detailed, lifelike effects.

Now, here's where it gets fun! You can apply **Toon Shading** on top of your **Light-based** setup by enabling **cell-based light partitioning**. This adds that signature "toon" effect by dividing your lighting into distinct chunks, transforming realistic lighting into a playful, exaggerated cartoon look.

With **Color-based Shading**, you can add **specular highlights** and **rim lighting** for extra gloss and dramatic outlines. **Light-based Shading** lets you include **rim lighting** to emphasize the edges of your objects, enhancing their cartoonish appeal.

Toon Styling

Styling is all about giving you full control over how your toon materials look. Whether you go for bold **(Cross-)Hatching** or fun **Halftones (Polka Dots)**, you can adjust everything from pattern density to stroke or dot size, edge smoothness, opacity, and more. Both styles also allow for randomization, letting you add natural variation, just like in real life, where lines and patterns aren't always perfectly uniform.

Hatching

For **Hatching**, you have control over:

- **Density** of the strokes
- **Stroke size** and **falloff**
- **Edge smoothness**
- **Opacity** and **falloff**
- **Color**
- **Draw space**: UV space or screen space
- **Coordinate system**: Cartesian or polar
- **Blending modes**: Lerp, additive, subtractive, multiply

Hatching-Specific Options:

- **Stroke direction**: Control the angle of your hatches
- **Number of hatching cells**: Add more layers for a cross-hatched effect

Halftones

For **Halftones**, you can adjust:

- **Density** of the dots
- **Halftone size** and **falloff**
- **Halftone hardness**
- **Opacity** and **falloff**
- **Color**

- **Draw space:** UV space or screen space
- **Coordinate system:** Cartesian or polar
- **Blending modes:** Lerp, additive, subtractive, multiply

Halftone-Specific Options:

- **Dot roundness:** Adjust the shape of your halftones
- **Line offset:** Fine-tune the spacing between halftone lines for precise control

Randomization for Both Styles

Both **Hatching** and **Halftones** allow you to introduce **randomization** to any of these settings—whether it's stroke/dot size, opacity, or density. This adds a natural, real-world feel, making your lines or dots vary in thickness, smoothness, or placement, just like a hand-drawn effect. It's perfect for adding that organic touch to your toon materials!

Outlines

Naturally, you can add outlines with ease! Take full control over the **color**, and **width**, giving your toon assets that perfect comic book look.

Optimization

No need to worry about performance! Our one-click optimization solution automatically generates the most efficient version of the shader based on your material settings, keeping everything fast and smooth.

Material Replacement Tool

Last but not least, let's talk about the **Material Replacement Tool**! This powerful feature allows you to **bulk apply** the shader to all your materials and keep them **in sync** with a reference material. But here's where it gets even better: you have **full control** over which properties get synced! Want to update only the styling but leave the toon shading untouched? No problem—just disable toon shading and enable only the styling. Every single property can be synced or left as is, giving you complete flexibility.

You can also choose whether to make the changes permanent in **Edit Mode** (before Play Mode) or just temporarily during Play Mode. Save time and effortlessly iterate on your entire scene without all the clicking!

What is a Shader?

To understand what a shader is, we highly recommend you to read the introduction on <https://en.wikipedia.org/wiki/Shader> and <https://docs.unity3d.com/Manual/shader-introduction.html>

We support the BiRP and URP Render Pipelines and every stable version from 2020 on!

That means we support:

Built-in
URP 2020, URP 2021, URP 2022 and URP 2023/6.

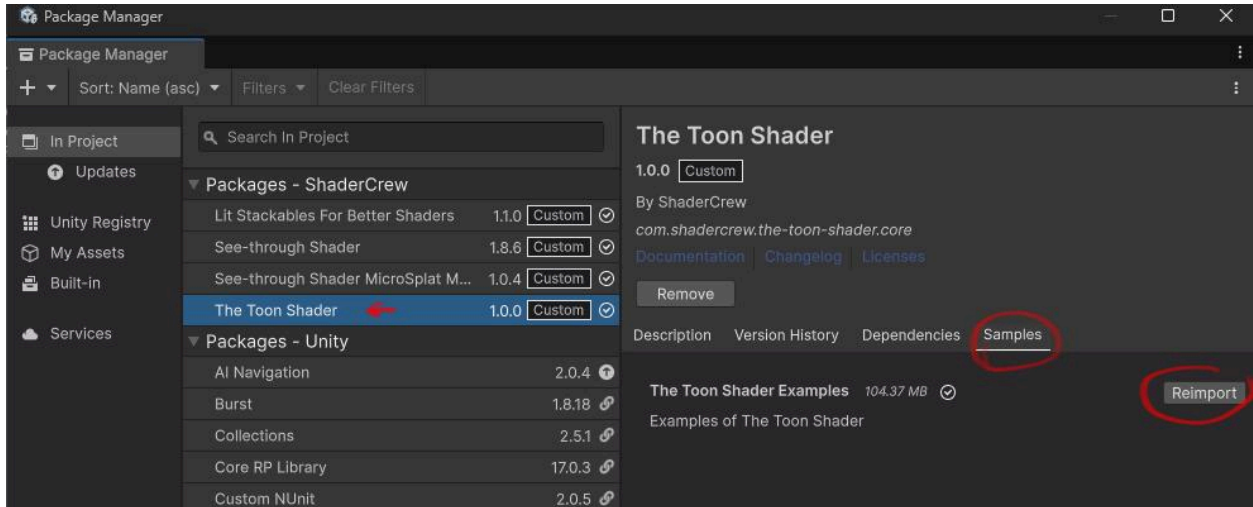
Note: Please know that you do NOT need to understand what a shader is to use our asset. Plenty of scripts are included so that you can entirely focus on the result without worrying about the inner workings.

How to Import the Samples

Unity Package Manager

The Demo Samples are shipped with the asset but have to be imported via the **Package Manager**.

After Importing the Asset follow these steps to import the samples:
Click Window > Package Manager and select “The Toon Shader”.
Now select the Samples Tab and press Import as shown here:



How does it work

The Toon Shader is a shader that you can apply to your materials.

There are **two ways of applying the shader** to your materials:

Either by setting it **manually** as the material shader or **automatically** by using the Material Replacement Tool, [ReplaceOrSynchronizeToonSettings.cs](#).

For more details on this, see [How to apply the shader](#).

The next step is to set the “The Toon Shader” shader settings (see [How to control the effect of the shader](#)).

How to apply the Shader

The first thing you have to do to use this asset is to apply the shader to your materials. You can either manually do this (see [Manually apply the shader](#)), or use one of our replacement scripts that automatically applies “The Toon Shader” to all materials of a specific group (see [Automatically apply the shader](#)).

Method 1 - Manually apply the shader

To add the shader to your material, you first need to add a material to your GameObject by clicking **Assets->Create->Material** in the project view context menu or from the main menu.

After that, assign the shader to your material using the Shader drop-down menu in the inspector window under TheToonShader.

You could also drag and drop the `.shader` onto the shader drop-down menu.

For more information regarding seeing the official Unity documentation on this topic: <https://docs.unity3d.com/Manual/materials-introduction.html>

Method 2 - Automatically apply the shader

Add the **TTS-Replace or Synchronize Toon Settings** component (`ReplaceOrSynchronizeToonSettings.cs`) and assign a **reference material**—which has **The Toon Shader**—under **Reference Material**.

Group Types

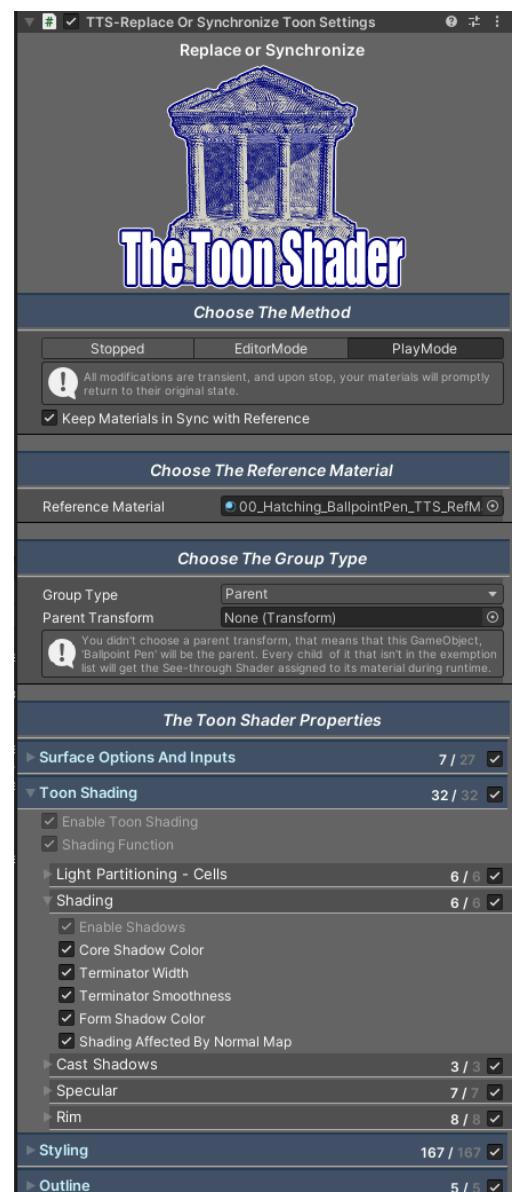
Choose **Parent** or **List of Materials**:

- **Parent:** Select a parent transform, and all child objects automatically use the property values from the reference material with The Toon Shader.
- **List of Materials:** Specify a list of materials, and each one is updated with the reference material property values.

Toon Shader Properties:

Decide which properties will sync with the reference material and which remain untouched.

- **Checked:** The property is replaced by the value from the reference material.
- **Unchecked:** The property stays as is (or defaults if it isn't originally a Toon Shader).



Modes:

- **Stopped:** Does nothing.
- **PlayMode:** Updates materials only in Play Mode and restores them to their original settings once you stop the scene.
- **EditorMode:** Permanently changes the materials, even outside Play Mode—use this only if you don't need to revert later.

Note: If you don't continuously change settings in the reference material, consider disabling **Keep Materials in Sync with Reference** to save on performance.

How to control the effect of the shader

The **The Toon Shader** uses a custom GUI, which is divided into five sections.

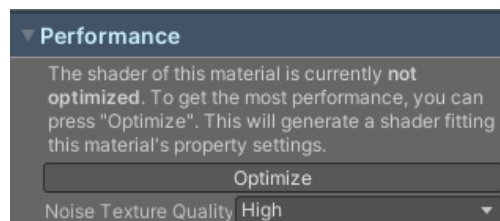
Those are:

1. Performance
2. Surface Options And Inputs
3. Toon Shading
4. Styling
5. Outline



Section 1 - Performance

The Performance Area lets you optimize your shader with one click. The optimized shader is automatically used in the current material and you can always switch back to the original one. The save location of the optimized shaders is:



“/Packages/com.shadercrew.the-toon-shader.core/Scripts/Resources/OptimizedShaders/”

Section 2 - Surface Options And Inputs

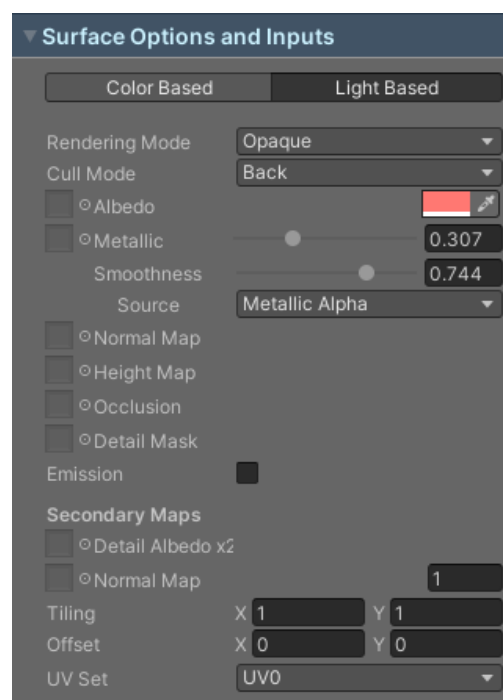
This section lets you choose between **color-based** or **light-based** shading. For light-based shading, you can select either **Blinn-Phong** or **PBR** lighting (though BiRP currently only supports PBR; let me know if you'd like Blinn-Phong added to BiRP).

Light-Based Shading

When you enable light-based shading, you'll have access to the same settings found in Unity's **Standard** (BiRP) and **SimpleLit/Lit** (URP) shaders.

Color-Based Shading

In color-based mode, you can only define an albedo texture and color, a normal map, and an emission texture and color—essentially an adapted **Unlit** shader.



By default, if **Toon Shading** is disabled, the shader behaves exactly like its Unity counterparts. For instance, **light-based PBR** aligns with Unity's **Standard/Lit** shader.

Note: Only the light-based mode supports multiple light sources!

To enable **posterization** or **cell-based** shading, simply turn on **Toon Shading**. For more details, see the next section, **Toon Shading**.

Section 3 - Toon Shading

Depending on which shading mode you select in **Surface Options and Inputs**, you'll see different controls here.

Light-Based Shading

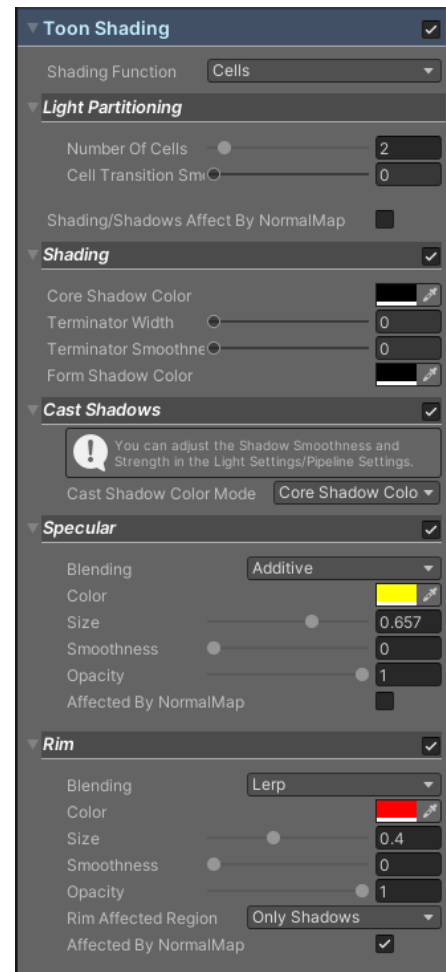
If you choose light-based shading, you can posterize it by dividing the shading into a set number of cells (1–15) and adjusting how smoothly each cell transitions. You can also decide whether the normal map affects shading and/or specular highlights, and you can enable Rim lighting.

Color-Based Shading

When using the color-based mode—essentially an unlit shader by default—you can still add shading through one of two functions: **Cells** or **Gradient**.

- **Cells:** Set shadow colors (core and form), define terminator width and smoothness, and choose how many cells and how smooth the transitions are, much like in the light-based mode.
- **Gradient:** Create a custom gradient to control how the shading is applied.

Finally, you can include **Specular** and **Rim** lighting in this section as well.



Section 4 - Styling

You can choose between various **styling** options, including procedural **hatching** and **halftones**. Each component—**Shading**, **Cast Shadows**, **Specular**, and **Rim**—can be styled independently. For convenience, you can also sync certain styles (for example, matching Specular to Rim). Syncing is enabled by default, so if you want to style Rim separately, set **Sync with** to **None**.

Both **hatching** and **halftones** give you control over blending, draw space, density, and more. Here's a quick overview:



Partitioning (Hatching only)

For **Hatching**, you can set the **Number of Cells**, which is similar to the posterization cells in “Toon Shading,” but they can overlap, letting you replicate cross-hatching when using more than one cell.

Blending

Determine how to blend the style with the rest of the shader and whether it’s inverted.

Draw Space

Choose **Object UV Space** or **Screen Space** and select either a **Cartesian** or **Polar** coordinate system.

Density & Direction / Density, Rotation and Offset

Density controls how dense the pattern is. It’s best to start from 0 and gradually increase it to avoid odd optical effects.

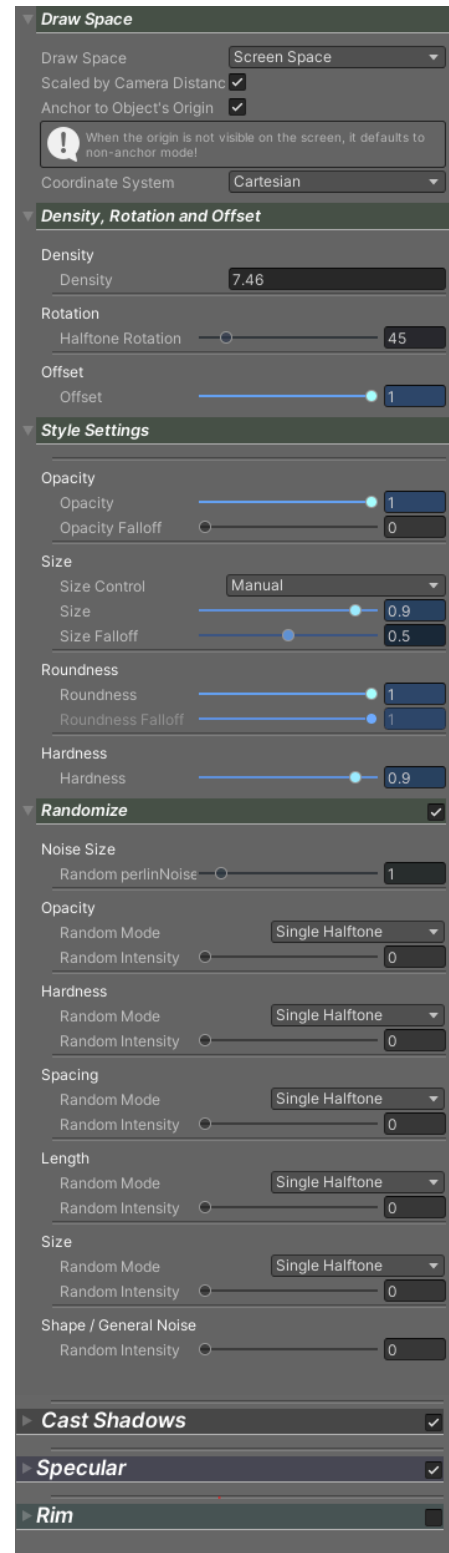
Adjust the **Initial Direction** and rotation between cells (for hatching) or the overall rotation (for halftones).

In halftones, you can also set an **Offset** between halftone rows.

Style Settings

Adjust **Opacity** and **Falloff**, **Thickness/Size** and **Falloff**, and **Hardness** for both Hatching and Halftones.

In Halftones mode, you can additionally fine-tune the **Roundness** and **Falloff** of the patterns.



Randomization

Add a natural feel by randomizing **Opacity**, **Hardness**, **Spacing**, **Length**, **Thickness**, and general **Shape**.

In Hatching mode, randomization can apply per line, to all lines at once, or a mix of both.

In Halftones mode, it can apply per halftone, to all halftones simultaneously, or a combination of both.

Section 5 - Outline

Finally, you can add an outline using a simple **inverted hull** method. More advanced techniques may be introduced in future versions since outlining wasn't the main focus for version 1.0. Keep in mind that this feature can be quite performance-heavy, so it's best not to use it on every object.

