Shadow Softener

Index

Introduction

Shadow Softener gives you total control over how Unity's shadows are filtered. It allows you to...

- Use PCF filtering up to 64-taps to create ultra-smooth shadows.
- Soften Directional, Spot, and Point lights, in Forward and Deferred renderers.
- Define custom filters, to make your game's shadows unique.
- Even enable Unity's *Soft* shadows to further filter the result!

Deferred Setup

Using Shadow Softener for Deferred Rendering is incredibly easy. Just follow these steps:

- **Quit and restart** the Unity Editor. This forces Unity to load the replacement Deferred shaders.
- To configure the global Shadow Filter(s), simply edit *ShadowSoftenerConfig.cginc* located in *Shadow Softener/Shaders/Deferred/Resources/*.
- Remember to use *Hard Shadows* with Directional Lights, if you don't want them looking noisy!

Forward Setup

Using Shadow Softener for Forward Rendering requires a simple change to *Shaders* you want supported. Remember, <u>not everything in your game will require crazy-soft shadows</u>, consider your most prominent *Shadow Receivers*, like the Ground or Walls.

Shader Converter

The easiest, and most common way to add support to a *Shader* is to use the included *Shader Converter*.

- Select one or more *Shaders*
- Select the *Tools/Shadow Softener/Convert Shaders* menu.

- Converter Options
  - *Create Backup* - Backs up your original Shader to YourShaderName.shader.backup.
  - *Shadow Filter* - The filter to use in the converted shader(s). If you Select "Built-in", Shadow Softener support will be removed.
- Press the *Convert Shaders* button...
- That's it! Your shader should now support Shadow Softener.

Manual Editing

If you'd prefer to manually edit your shader, simply add the following line inside your *SubShader* code block:

```
// Valid filters are PCF2x2, up to PCF8X8
UsePass "ShadowSoftener/PCF4X4"
```

Forward Setup (Mobile/Advanced)

Some features will not work with the basic Forward Setup, these include:

- Mobile (OpenGL ES 3.0) support
- Vertex modification
- Forward-rendered *Point* or *Spot* shadows

To support any of these features, make the following changes to your *CGPROGRAM* section:

```
// Use addshadow when not using UsePass "ShadowSoftener/..."
// Use fullforwardshadows to enable Forward Spot and Point shadows.
#pragma surface surf Lambert addshadow fullforwardshadows

// Shadow Softener requires Shader Model 3.0
#pragma target 3.0

// Define the Shadow Filter
#define SOFTENER_FILTER PCF4x4

// Include Shadow Softener
#include "Assets/Shadow Softener/Shaders/ShadowSoftener.cginc"
```

Notes

- Adding Shadow Softener changes how a shader **Recieves Shadows**, not how it casts them.
- Unity provides the source to built-in shaders if you wish to modify them. [Download Page](#)
- There are example shaders based on the built-in *Diffuse* shader in *Shadow Softener/Shaders/Forward/Samples/*.
- Remember to use *Hard Shadows* with Directional Lights, if you don't want them looking noisy!

Configuration

Whether you edit the Deferred config or Forward Shaders, the options available to you are the same. All options must be defined *before* including ShadowSoftener.cginc.

Setting a Shadow Filter

```
// Set the Shadow Filter
#define SOFTENER_FILTER PCF4x4
```

Available Directional/Spot Filters

- **PCF2x2** - 2x2 Percentage-Closer Filter with Edge-Tap Smoothing. (Hand Optimized)
- **PCF3x3** - 3x3 Percentage-Closer Filter with Edge-Tap Smoothing. (Hand Optimized)
- **PCF4x4** - 4x4 Percentage-Closer Filter with Edge-Tap Smoothing. (Hand Optimized)
- **PCF5x5** - 5x5 Percentage-Closer Filter with Edge-Tap Smoothing. (Generic Implementation)
- **PCF6x6** - 6x6 Percentage-Closer Filter with Edge-Tap Smoothing. (Generic Implementation)
- **PCF7x7** - 7x7 Percentage-Closer Filter with Edge-Tap Smoothing. (Generic Implementation)
- **PCF8x8** - 8x8 Percentage-Closer Filter with Edge-Tap Smoothing. (Generic Implementation)
- **CUSTOM_FILTER** - Specifies that you'll be using a Custom Filter function.
  For the adventurous, a sample is included: "*Shadow Softener/Shaders/Forward/Samples/Normal-Diffuse (Custom Filter).shader*".

Available Point Filters

- **PCF4** - A 4-Tap Percentage-Closer Filter for Point Lights.
- **PCF**[...] - Any multiple of 4 Taps will work. (ie. *PCF20*, *PCF32*, *PCF44*, etc)
- **PCF64** - A 64-Tap Percentage-Closer Filter for Point Lights.

Note: If you specify a Directional/Spot filter such as *PCF4x4*, a compariable Point filter will be selected.

Setting Filters by Light Type

If you wish to specify the Filter by Light Type, use the "_DIRECTIONAL", "_SPOT", or "_POINT" suffix as shown in this example:

```
// Set the Directional Light Filter
#define SOFTENER_FILTER_DIRECTIONAL PCF5x5

// Set the Spot Light Filter
#define SOFTENER_FILTER_SPOT PCF3x3

// Set the Point Light Filter
#define SOFTENER_FILTER_POINT PCF4x4
```

Configuration (Continued)

Advanced Options

- **SOFTENER_SAMPLE_BIAS_OFF** - Disables per-sample depth biasing, which is intended to reduce *Shadow Acne*. Disabling this feature will improve performance at the cost of increasing *Shadow Acne*.

```
// Example Usage
#define SOFTENER_SAMPLE_BIAS_OFF
```

- **SOFTENER_SPOT_BIAS_SCALE** - Scales the Spot Light Sample Bias:

```
// Example Usage (Default: 0.01)
#define SOFTENER_SPOT_BIAS_SCALE 0.02
```

- **SOFTENER_POINT_BIAS_BASE** - Sets the baseline Point Light Bias:

```
// Example Usage (Default: 0.03)
#define SOFTENER_POINT_BIAS_BASE 0.025
```

- **SOFTENER_POINT_BIAS_SCALE** - Sets the amount to increase the Point Light Bias as the kernel increases:

```
// Example Usage (Default: 0.01)
#define SOFTENER_POINT_BIAS_SCALE 0.015
```

Writing a Custom Filter

If you're feeling a bit adventurous, or simply masochistic, you can actually write Custom Filters!
Take a look at the included sample: *Shadow Softener/Shaders/Forward/Samples/Normal-Diffuse (Custom Filter).shader*

Troubleshooting

The following are some common problems you might run into, and solutions.

- If your shadows look *Noisy*, set the light to *Hard Shadows* in the Inspector.
- If you run out of *Instructions*, or *Temporary Registers* in a shader, use a smaller filter.

Support

If you need help for whatever reason, just let me know in the [Official Support Thread](#).
I can also be reached [On Twitter](#).