Sharp Shadows Toolkit

version 1.1.1

The Sharp Shadows Toolkit is a code package that brings sharp, pixel-perfect shadows to your Lightweight/Universal Render Pipeline project. Many games and experiences target a non-photo-realistic art style for which sharp shadows are better suited than the built-in shadows that tend to be either too soft or too jagged, especially on low quality settings.

The toolkit can be useful in a wide variety of scenarios, from a single accurate drop-shadow for the main character in a mobile game to ubiquitous high quality shadows for an architecture visualization project.

Contents

1 What's new

v1.1.1

- Fix compile error when `SHARP_SHADOWS_DEBUG` was not defined
- Do not throw errors on missing `SharpShadowManager` unless `SHARP_SHADOWS_DEBUG` is defined

v1.1.0

- Add support for Unity 2021 and URP 11
- Add SharpShadowManager to centralize rendering (results in less Update() calls)
    - Make sure to add a game object with this script to existing scenes (see section 4)
- Reduce shadow mesh asset file size by omitting mesh components that are not relevant for shadows
- Do not render sharp shadows if target platform does not have stencil buffer support

v1.0.4

- Add VR-support to all shaders (Performance is quite bad though, due to the high fill-rate)

v1.0.3

- Update example scene to Unity 2019.3 (Prefab Variants would not load correctly in newer versions of Unity for some reason)

v1.0.2

- Add a `Legacy Prefab Converter` tool to facilitate adding sharp shadows to prefabs or game objects that are not prefab variants of the source model prefab
- Fix bug where bounds pad factor would not influence skinned mesh renderers
- Enable instancing for shadow volumes (noticeable when SRP batcher is turned off)

v1.0.1

- Support for Universal Render Pipeline
- Minor changes to README

v1.0.0

- Initial release

2 Overview

Features

- Dynamic pixel-perfect shadows for the Lightweight/Universal Render Pipeline
- No jagged shadow edges or visible pixels
- Seamless editor integration using automatic asset post processing
- Two rendering modes
    - Inject into screen space shadow texture for optimal image quality
    - Draw after opaque geometry for optimal performance
- Handles both skinned and non-skinned meshes
- No restriction on the mesh geometry of the shadow caster
- Multi-object editing
- Fog support

Current Limitations

- For the Lightweight/Universal Render Pipeline only
- Only supports the main directional light of the scene
- No support for custom vertex displacement
- Alpha blended or clipped geometry will produce solid shadows (the shape of the triangles)
- Skinned mesh shadows may appear cracked if they are subject to large deformations
- Shadows are not anti-aliased when injected into screen space shadow texture

3 Supported Unity builds

Unity version	Build tested	LWRP/URP version
Unity 2019.1	2019.1.10f1	5.7.2
Unity 2019.2	2019.2.3f1	6.9.1
Unity 2019.3	2019.3.0b2, 2019.3.5f1	7.0.1, 7.1.8
Unity 2021.1	2021.1.11f1	11.0.0

Note that because of the large number of possible Unity/URP version combinations, only the above combinations are supported.

4 Instructions

4.1 After importing the toolkit into a project for the first time

1. Choose `Assets` → `Reimport all` to reimport all assets
2. Set up the Scriptable Render Pipeline using the following steps:
   - Go to `Edit` → `Project Settings...` → `Graphics` and choose the *SharpShadowRP* as `Scriptable Render Pipeline Settings`
   - Or if you already have a `LightweightRenderPipelineAsset` / `UniversalRenderPipelineAsset` in your project: Select it → `Renderer Type: Custom` → Choose *SharpShadowRenderer* as `ForwardRendererData`
   - Or if you already have a custom `ForwardRendererData`: Select it → Add a new `Renderer Feature` using the (+)-sign → Add the *ShowSharpShadows* renderer feature

Make sure that the `LightweightRenderPipelineAsset` / `UniversalRenderPipelineAsset` has "Lighting → Main Light → Cast Shadows" enabled and that the main directional light of the scene has its "Shadow Type" set to "No Shadows" (this disables Unity's built-in shadows).

At least in Unity 2021, sharp shadows might not show up if there are multiple `UniversalRenderPipelineAssets` in the project. If the shadows do not show up in the example scene at this stage, remove all other `UniversalRenderPipelineAssets` in the project.

3. Add a game object with a `SharpShadowManager` script to any scene that should display shadows

4. The shadows will appear the next time you open the scene or start Play-mode

4.2 Workflow

Simply drag & drop a model asset into a new scene or open up an existing scene after all models have been reimported. Unity's built-in shadows should now be replaced by sharp shadows.

If you are using prefabs in your project, make sure that prefabs for models are what is known as *Prefab Variants* in order to keep them updated when the source model (and its shadow) changes. To create a *Prefab Variant* from a model, right-click on the model in the Project panel and choose `Create` → `Prefab Variant`.

4.3 Under the hood

The toolkit automatically updates all 3d models at import time using a custom asset post-processor. In this post-processing step, `Sharp Shadow` components are attached to all game objects with a `Mesh Renderer` or `Skinned Mesh Renderer`. A `Sharp Shadow` component is responsible for rendering a sharp shadow for a single game object. Each sharp shadow component references a `Shadow Asset` that stores a special mesh and a set of configurable options for the shadow in question. This shadow asset is automatically created, also at import time, and saved to a per-model `Shadow Asset Collection` that is placed next to the model in the project. The new shadow asset collection will have the same name as the model but with a "_shadows.asset" postfix.

4.4 Prefabs not connected to model assets

Legacy Prefab Converter

The `Legacy Prefab Converter` tool automatically adds sharp shadows to a prefab or game object that is not a prefab variant of a model prefab or a plain model prefab instance. This can be useful when retrofitting sharp shadows into existing projects where model connections have been broken. Note that the tool is currently in "preview" and might not work correctly in all situations.

To use the tool, open it by clicking `Window` → `Sharp Shadows Toolkit` → `Legacy Prefab Converter (Preview)` and follow the on-screen instructions.

Manually

To manually add sharp shadows to a game object, follow these steps:

1. Add a `Sharp Shadow` component to each game object with a `Mesh Filter` component
2. For each skinned mesh renderer, create an empty child game object and attach a new skinned mesh renderer component and a `Sharp Shadow` component.
3. For each sharp shadow component, choose the `Shadow Asset` that corresponds to the mesh being rendered by that game object. For example, if a mesh filter component specifies a mesh called `Monkey`, choose the shadow asset called `Monkey_shadow_asset`.

4.5 Built-in meshes

The built-in Unity meshes (*Cube*, *Capsule*, *Cylinder*, *Plane*, *Sphere* and *Quad*) do not go through the usual model asset pipeline and will thus not get any sharp shadows by default. To work around this, manually add a `Sharp Shadow` component to the game object in question and enable the `Create Runtime Shadow Asset` property.

4.6 Procedurally generated meshes

To show shadows for meshes generated at runtime, attach a `Sharp Shadow` component to the game object in question after the generated mesh has been set to the `Mesh Filter`. Then, enable the `Create Runtime Shadow Asset` property.

```
public void Start()
{
    // Generate custom mesh ...
    // ...
    gameObject.GetComponent<MeshFilter>.sharedMesh = customMesh;

    // Add shadow
    var sharpShadow = gameObject.AddComponent<SharpShadow>();
    sharpShadow.createRuntimeShadowAsset = true;
}
```

If the mesh is re-generated multiple times after the game object has been instantiated, call `gameObject.GetComponent<SharpShadow>().CreateRuntimeShadowAsset(true)` to update the shadow.

4.7 Opting out of asset post-processing

Some model assets are not intended to have shadows ever. For these assets, it can be useful to disable asset post-processing altogether. To do so, make sure that the model asset path contains the string “NoShadow”, case insensitive.

Here are a few examples of model assets that do not generate sharp shadows:

```
Assets/Models/NoShadow/A.fbx
Assets/Models/NoShadow/B.fbx
Assets/Models/NoShadow/C.fbx
Assets/Models/Monkey_NoShadow.fbx
Assets/Models/Airplane_noShadow.fbx
Assets/Models/noshadow_Tree.fbx
```

5 Settings

5.1 Project Settings

The global settings for the toolkit can be configured in `Edit` → `Project Settings...` → `Sharp Shadows Toolkit`. These settings are saved in `Assets/SharpShadowsToolkit/Scripts/Editor/ProjectSettings.asset` and are thus shared by all team members of a project.

It is not recommended to omit this file from version control because if two team members have different settings, they will overwrite each others shadow assets every time one of them reimports a model and commits the changes.

5.1.1 Model shadow asset generation defaults

These defaults are used when creating a new shadow asset in the model post-processor. Note that it is always possibe to change the initial values manually on a per-shadow-asset basis by selecting the asset in the Project window.

Allow Camera In Shadow

Enable if a camera will be in the shadow of this game object. If enabled, the shadow volume must be rendered twice degrading pixel fillrate performance in particular. For better performance, disable this property whenever your project allows for it. Under the hood, the implementation uses a 2-pass workaround for the 1-pass depth-fail algorithm.

Render Layer

The render layer used to render the shadow volumes. Make sure that this layer is not rendered by the currently active forward renderer.

Bounds Pad Factor

The factor to expand the shadow mesh bounds by in order to prevent the shadow from disappearing when the game object is outside the camera view frustum. The bounds will be expanded by this value times the magnitude of the size vector of the original

bounds.

5.2 Sharp Shadow component

Shadow Asset

The shadow asset to use when rendering this shadow. This property is automatically set when a model is post-processed but may be empty if the sharp shadow component was added manually.

Create Runtime Shadow Asset

Enable this property if there is no build-time mesh to create a shadow mesh asset from. This is usually the case for procedurally generated meshes. A runtime shadow asset will take precedence over the shadow asset property, if specified.

Runtime Creation Settings

The settings to use when creating the runtime shadow asset

5.3 Show Sharp Shadows render feature

Enabled

Quick toggle for switching sharp shadows on or off

Shadow Volume Render Layer

The shadow asset render layer to visualize

Near Extrusion Distance

The amount to extrude the part of the shadow volume facing towards the light

Far Extrusion Distance

The amount to extrude the part of the shadow volume facing away from the light”;

ShadowIntensity

The shadow intensity

Shade Mode

The shade mode to use when drawing the sharp shadows. 'Inject Into Screen Space Shadow Resolve Texture' renders a scene depth pass and the shadow volumes into the screen space shadow resolve texture of the Lightweight/Universal Render Pipeline. In this mode, the sharp shadows are integrated into the render pipeline the same way the built-in shadows are, which makes them play nice with other graphics features such as light-mapping and non-main lights. The downside is that an extra render target the

size of the screen is used and the scene must be rendered twice. 'Multiply Scene After Opaque' draws the shadow volumes after all opaque objects have been rendered by simply multiplying the shadow intensity with the current color. This does not play nice other graphics features but works well for high-contrast art styles. The upside of this mode is that it performs really well, especially on low-end devices.

Mitigate Self Shadow Artifacts

Reduce self-shadowing artifacts using a clever trick. Can look odd on complex concave meshes when using the 'Multiply Scene After Opaque' shade mode but improves image quality for most kinds of meshes. Does not degrade image quality when using the 'Inject Into Screen Space Shadow Resolve Texture' shade mode.

6 Performance

6.1 Shade Mode

As described in section 5.3, the shade mode used to render the shadows impacts performance. The following table shows how a given factor impacts performance when using a given shade mode:

	Inject	Multiply
Shadow complexity	fillrate[1]	fillrate[1]
Scene complexity	scene is drawn twice	scene is drawn once
Screen resolution	fillrate[2] + gpu memory[3]	fillrate[2]

[1] The shadow technique used to render the sharp shadows requires a high fillrate as many shadow volumes may be drawn on top of each other. The best performance is achieved when the shadow volumes are cast a limited distance before being blocked by an occluder, for example a shadow casting object lying on the ground when viewed from a top-down perspective, and when the `Allow Camera In Shadow` setting is disabled, see section 6.2. See the wikipedia entry for fillrate for more information.

[2] The screen resolution matters because it affects how much fillrate is available. See the wikipedia entry for fillrate for more information.

[3] The screen space shadow resolve render texture uses the R8 color format if available or, if it is not available, falls back to the ARGB32 color format.

6.2 Allow Camera In Shadow setting

As described in section 5.1, the `Allow Camera In Shadow` setting impacts performance. Not only must shadows with this setting enabled be rendered twice, they are rendered with

depth tests disabled, impacting fillrate even if the shadow is not seen.

For optimal performance, only enable this setting where it is absolutely necessary and remember that it is a per-shadow-asset setting and consequently a per-model-setting (i.e. you can choose which types of models to enable it for, such as “only for buildings that are large enough to cast shadows onto the player's camera”).

6.3 Mesh quality

Even though the toolkit has no restrictions on the mesh geometry of the shadow caster, the quality of the shadow caster mesh influences the runtime performance of the shadows.

For best performance, keep shadow casting meshes closed and two-manifold. That is, make sure *that each edge in the mesh is part of exactly two triangles*.

For example, if the mesh for a vase is open at the bottom, the triangles that make up the very bottom of the vase will each contain an edge that is part of only one triangle. This makes the mesh non-manifold and forces the toolkit to render more complex geometry that degrades performance.

7 Platform support

Note that the end-user target device must have stencil buffer support for the shadows to work.

Platform	Supported	Note
Desktop	Yes	
WebGL	Yes	
iOS	Yes	Should work fine but not tested
Android	Yes	Tested on Fairphone 2 (Android 7), OnePlus A6003 (Android 9)
VR	Yes	Performance is bad due to high fill-rate
Others	-	Please let me know :)

8 Deployment

The toolkit requires the following set of shaders to always be included with your builds for shadows to show up:

```
SharpShadowsToolkit/DrawDepthFromTexture
SharpShadowsToolkit/VisualizeShadowsFullscreen
SharpShadowsToolkit/VolumeUpdateStencilAlways
SharpShadowsToolkit/VolumeUpdateStencilOnDepthPass
```

Go to `Edit → Project Settings... → Graphics → Built-in Shader Settings` and add these shaders to the list of shaders if not already present.

9 Implementation details

The toolkit uses the shadow volumes technique to draw pixel-perfect shadows. The toolkit is *not* using the patented depth-fail technique. Instead, it uses the normal depth-pass shadow volume algorithm for most shadow casters and a workaround for the depth-fail technique for shadow casters where the camera can be located inside the shadow volume. The workaround renders the shadow volume one extra time with depth tests turned off alltogether in addition to the normal depth-pass shadow volume pass. The workaround is enabled for a specific shadow caster when the 'Allow Camera In Shadow' property is set on the shadow caster's shadow asset.

10 Contact

Please let me know if you run into any problems when using the toolkit or if you have feedback on how I can improve it in the future. I am also interested in seeing projects that use any of my toolkits in practice!

Website: https://gustavolsson.com/

Contact: https://gustavolsson.com/contact/