



Project

G.A.M.E.D.E.V.

Liquid Shader

v1.0
Documentation

Contents

1	Introduction	2
2	Documentation	3
2.1	Universal Render Pipeline (URP)	3
2.1.1	Universal Renderer	3
2.1.2	2D Renderer	13
3	Support	14

Section 1

Introduction

The Liquid shader turns any Sprite Shape into a living body of liquid – lava, water, blood, honey, acid, slime, mud, oil, molten metal, poison, radioactive waste, and anything in between. The liquid surface flows, rolls in waves, breaks into ripples, and grows floating patches of crust, foam, or scum, all of which are fully customizable.

WARNING

This shader supports only Universal Render Pipeline (URP) on Unity 6.0+. Built-in Render Pipeline (BIR) and High-Definition Render Pipeline (HDRP) are not supported.

Two helper scripts are included. The **Liquid Sprite Shape Geometry Creator** replaces Unity’s default Sprite Shape geometry with a dense, evenly tessellated mesh, so that the wave and ripple displacement has enough vertices to work with. The **Quick Liquid Setup** component condenses the many shader properties into five intuitive controls, letting you dial in a convincing liquid in seconds.

Thirteen ready-made liquids ship with the asset – each with its own textures, material, and Sprite Shape Profile – along with a demo scene for every one of them. Feel free to check them out for inspiration, or simply use them as they are.

Section 2

Documentation

2.1 Universal Render Pipeline (URP)

2.1.1 Universal Renderer

The Universal Renderer is the default URP Renderer when creating a 3D project. Please use the **/Universal Renderer** folder exclusively for this Renderer.

Setup

The following paragraph presents the steps necessary to make the Liquid shader work in your project.

Visit <https://docs.project-gamedev.com/liquid-shader/setup.html> for image-aided instructions.

The Liquid shader is, in principle, the same as the regular lit shader you would use in your Unity project, only reconstructed and expanded. It is meant to be applied to a **Sprite Shape**, since the liquid surface is animated by displacing the mesh vertices of the shape:

- Create a new **Sprite Shape** in your scene (**GameObject** → **2D Object** → **Sprite Shape**) and draw the outline of your body of liquid with the spline editor.
- Assign one of the included Sprite Shape Profiles (from the **/Sprite Shape Profiles** folder) to the **Profile** field of the Sprite Shape Controller.
- Assign one of the included Geometry Creators (from the **/Geometry Creators** folder) to the **Creator** field of the Sprite Shape Controller. Consult with 2.1.1 for more information.
- Create a new material for your liquid; then, select **Project G.A.M.E.D.E.V./Essential 2D Shaders/Liquid Shader** as the shader source for this material.
- Assign the new material to the first material slot of the **Sprite Shape Renderer**.

That's it! The shader is now active. Feel free to play around with the material properties to find a liquid that you are happy with, or attach the **QuickLiquidSetup.cs** component described in 2.1.1 to shape the liquid with a handful of controls instead.

Keep in mind that certain post-processing settings (most notably **Bloom** and **Tonemapping**) will have a profound effect on the emissive parts of your liquid. The demo scenes ship with the Post-Processing Profiles used to author them, in the **/Post-Processing Profiles** folder.

i NOTE

The waves and the ripples are produced by displacing the mesh vertices of the Sprite Shape. Unity's default Sprite Shape geometry has far too few vertices for that, which is exactly why the included Geometry Creator exists. If the surface looks blocky or barely moves at all, raise the **Mesh Detail** of the Geometry Creator asset you assigned.

Shader Options

The material inspector groups the shader options into five categories – **Main Surface**, **Emission**, **Liquid**, **Waves**, **Ripples**, and **Patches** – described below.

Main Texture The main (albedo/color) texture used by the shader. This is the texture that is scrolled across the surface by the **Liquid** options.

Albedo An additional color multiplied with the one already present in the main texture. Use this to further customize the look of your liquid.

Alpha Cutoff This is a threshold value – all parts of the texture whose alpha value is less than the alpha cutoff will not be rendered.

Ambient Occlusion (Universal Renderer only) The regular ambient occlusion from the standard URP Lit shader.

Metallic Map (Universal Renderer only) The standard metallic map option in the default URP Lit shader. The metallic map is a black-and-white texture that determines the surface type of the object. It is used to make certain parts of the liquid reflect light like a metal would (metallic), and others like sand would (smooth).

Metallic (Universal Renderer only) This value determines how “metallic” the surface should be, i.e., whether it should reflect light like a metal or a wet surface, or should absorb it like sand or another rough, non-metallic surface.

Normal Map The standard normal map option in the default URP Lit shader. The normal map is used to simulate a 3D-surface by containing information on how the light should be reflected by the surface. Every included liquid ships with a matching normal map.

Normal Strength This value is multiplied with the normal map and is used to make the effect stronger or weaker. A negative value will mirror the 3D-effect (i.e., protruding surfaces will now be concave, and vice versa).

Vertical Offset Shifts the whole liquid surface up or down along the vertices of the shape. Use this to sink the liquid deeper into its basin or to make it swell above the rim, without editing the Sprite Shape spline itself.

Emission Color The emission color as an HDR color. Keep in mind that **different post-processing effects (most notably bloom)** will have a **profound** effect on the look of your emission. This is the property that makes lava, magic liquid, and radioactive waste glow.

Emission Map The standard emission map option in the default URP Lit shader. A black-and-white map that determines which parts of the liquid will emit light. The white pixels of the texture are areas where the emission will have an effect, and the black pixels are the areas with no emission.

Emission Extra Intensity Since the default HDR color has a limited intensity (-10 to 10), this shader offers an additional intensity option that allows you to make your emission even stronger.

i NOTE

The emission is additionally faded out towards the outline of the Sprite Shape, using the edge mask written by the Geometry Creator. This is what gives lava its cooled, darkened rim where it meets the rock. The width and the falloff of that rim are controlled by the **Edge Mask Width** and **Edge Mask Softness** settings described in 2.1.1.

Liquid Movement Speed How quickly the liquid textures flow across the surface. Set to 0 to bring the flow to a complete standstill. Larger values produce a faster current.

Liquid Movement Direction The direction of the flow, in degrees (0 – 359). 0 flows along the +X axis, 90 flows along the +Y axis.

Wave Speed How quickly the waves travel across the surface. Set to 0 to freeze the waves in a single, static deformation.

Wave Direction The direction in which the waves travel, in degrees (0 – 359). Off-setting this slightly from the **Liquid Movement Direction** makes the surface look considerably more natural, since the waves then no longer run exactly along the current.

Wave Frequency How many wave crests fit across the surface (0 – 50). Low values produce long, rolling swells; high values produce short, choppy waves.

Wave Amplitude How far the vertices are displaced by the waves, i.e., how tall the waves are. Set to 0 to flatten the surface completely.

Wave Distortion Strength The amount of noise mixed into the wave motion (0 – 10). At 0 the waves are perfectly regular sine waves. Raising this value breaks that regularity up, so that no two crests look quite the same.

Ripple Speed How quickly the ripples – the second, finer layer of displacement on top of the waves – move across the surface.

Ripple Size The size of an individual ripple (0 – 1). Small values produce a dense pattern of tiny ripples; large values produce a few broad ones.

Ripple Amplitude How far the vertices are displaced by the ripples. Set to 0 to disable the ripple layer entirely and leave only the waves.

Ripple Frequency How rapidly the ripples oscillate. Larger values make the fine surface detail flicker and boil rather than gently undulate.

i NOTE

The waves and the ripples are two independent displacement layers, and they are meant to be used together. A thin liquid such as water usually wants fast, small, plentiful ripples over shallow waves, whereas a thick liquid such as lava or honey wants tall, slow waves and barely any ripple at all.

Patch Speed How quickly the patches – the darker crust, foam, scum, or mud floating on the surface – drift across the liquid.

Patch Density How many patches are spread across the surface (0 – 10). Low values produce a few large islands; high values produce many small ones.

Patch Movement Direction The direction in which the patches drift, in degrees (0 – 359). It is usually worth setting this a little apart from the **Liquid Movement Direction**, so that the patches appear to be carried by the current rather than glued to it.

Patch Threshold The cut-off that decides how much of the surface is covered by patches (0 – 2). Low values leave only the densest cores of the patches visible; high values let the patches spread out and merge into a closed crust.

Patch Softness How sharply the patches end (0 – 2). At 0 the patches have hard, clearly defined borders; larger values let them fade gradually into the surrounding liquid.

Patch Color The color of the patches. Dark, desaturated colors read as cooled crust or scum, while lighter ones read as foam.

Patch Distortion Strength The amount of noise warping the outline of the patches. At 0 the patches keep the smooth, cell-like shape they are generated with. Larger values eat into their borders and make them look torn and irregular.

Patch Transparency How much of the liquid underneath shows through the patches (0 – 1). At 0 the patches fully replace both the color and the emission of the liquid below them; at 1 they disappear entirely.

i NOTE

The quickest way to turn the patches off completely is to raise **Patch Transparency** to 1. The **Quick Liquid Setup** component described in 2.1.1 offers a **None** patch preset that achieves the same result for you.

API

Visit <https://docs.project-gamedev.com/liquid-shader/api.html> for the Liquid Shader API documentation. This API allows you to modify all shader values in-code, from your own scripts, if necessary.

Liquid Sprite Shape Geometry Creator

Unity generates the fill of a Sprite Shape with as few triangles as it can get away with. That is perfectly reasonable for a static shape, but it leaves the Liquid shader with almost no vertices to displace, and the waves and the ripples with almost nothing to act on.

`LiquidSpriteShapeGeometryCreator.cs` solves this. It is a `SpriteShapeGeometryCreator` – a `ScriptableObject` that a Sprite Shape Controller uses in place of Unity’s built-in geometry generator. It fills the spline with an evenly tessellated grid of triangles, generates the UVs for it, keeps the collider working as usual, and writes an **edge mask** into the red channel of the vertex colors, which the shader uses to fade the emission out towards the outline of the shape.

Three ready-made Geometry Creator assets are included in the `/Geometry Creators` folder – a general-purpose one, a **Lake** variant, and an **Edge** variant with a much wider edge mask. To use one, drag it onto the **Creator** field of your Sprite Shape Controller.

To create your own, use **Assets** → **Create** → **Project GAMEDEV** → **Essential 2D Shaders** → **Liquid Sprite Shape Geometry Creator**. The asset exposes the following settings:

Mesh Detail The density of the generated mesh (1 – 128), i.e., both the number of vertex columns per local unit and the number of quad rows. This is the single most important setting of the asset: higher values produce smoother, more detailed vertex displacement, at the cost of a heavier mesh. Values around 12 – 32 are plenty for a calm liquid; violent lava benefits from far more.

Open Ended Width The total width of the generated strip when the Sprite Shape spline is **open ended** rather than closed. The spline stays centered inside the strip. This setting is ignored for closed shapes, which are simply filled.

Stretch UVs To Bounds Generates the UVs from the bounds of the generated mesh instead of tiling them from the local position. Enable this when you want a single copy of the liquid texture stretched across the whole shape; leave it disabled to have the texture tile at a consistent scale regardless of how large the shape is.

Edge Mask Width The distance from the outline of the Sprite Shape over which the edge mask fades out. The mask is stored in the red channel of the vertex colors and is used by the shader to darken the emission near the outline – a wide value produces a broad, cooled rim, while a very small value effectively disables the rim.

Edge Mask Softness The softness of that falloff (0 – 1). At 0 the mask ends abruptly, giving a hard border; at 1 it fades smoothly across the full edge mask width.

NOTE

A Geometry Creator asset is shared by every Sprite Shape that references it, so changing its settings updates all of them at once. If you want one liquid to be denser than another, give it its own Geometry Creator asset.

⚠ WARNING

Sprite Shape meshes use 16-bit index buffers, which caps them at 65,535 vertices and indices. The Geometry Creator respects that limit and reduces the effective detail of very large shapes rather than fail. If a huge shape looks coarser than a small one at the same **Mesh Detail**, this is why – split it into several smaller Sprite Shapes instead.

Quick Liquid Setup

The Liquid shader exposes a lot of properties, and finding a believable combination of all of them by hand takes time. `QuickLiquidSetup.cs` exists to skip that step: it maps five intuitive controls onto the whole set of shader properties, so that a plausible liquid is only a couple of sliders away.

Attach the component to the same `GameObject` as your Sprite Shape Renderer via **Add Component** → **Project GAMEDEV** → **Essential 2D Shaders** → **Quick Liquid Setup**. It updates immediately, both in the Editor and in Play Mode, and offers the following settings:

Movement Speed The base speed of the whole liquid – the flow, the waves, the ripples, and the patches. Each of those layers receives its own multiplier on top of this value, so that they never move in lockstep. Set to 0 to bring the liquid to a complete standstill.

Movement Direction The primary direction of movement, in degrees (0 – 360). 0 moves along the +X axis, 90 moves along the +Y axis. The waves and the patches are automatically offset by a few degrees from this direction to keep the surface from looking mechanical.

Viscosity How thick the liquid is (0 – 1). Thin liquids such as water move quickly, with fine, lively surface detail. Thick liquids such as lava, honey, or tar crawl, with a heavier, slower surface response and a glossier finish.

Volatility How energetic the surface is (0 – 1). Low values produce an almost still surface; high values produce tall waves, strong ripples, and pronounced vertex displacement.

Patch Preset The behaviour of the patches floating on the liquid, as one of seven presets: **None**, **Subtle**, **Sparse**, **Balanced**, **Dense**, **Crust**, and **Slick**. **None** disables the patches entirely; **Crust** covers the surface in large, hard-edged plates of cooled crust; **Slick** produces a thin, oily film. The color of the patches is not part of the preset – set it on the material, with the **Patch Color** property.

NOTE

The component writes its values through a `MaterialPropertyBlock`, which means it never modifies your material asset. Disabling or removing the component restores the material to exactly the values you authored in the inspector, and two Sprite Shapes sharing one material can still be given different Quick Liquid Setup settings.

All five settings are also available in code, as the `MovementSpeed`, `MovementDirection`, `Viscosity`, `Volatility`, and `Patches` properties. Assigning to any of them re-applies the setup immediately, which makes it straightforward to have, for example, lava thicken and slow down as it cools. The **Apply Quick Liquid Setup** context-menu entry on the component does the same thing on demand.

i NOTE

Because **Quick Liquid Setup** overrides the movement, wave, ripple, and patch properties of the material, changing those properties on the material itself will have no visible effect while the component is enabled. Remove or disable the component if you would rather author every property by hand.

⚠ WARNING

Quick Liquid Setup requires a **Renderer** on the same **GameObject**, and only one instance of the component per **GameObject** is allowed. If nothing happens after attaching it, make sure the **GameObject** actually carries a **Sprite Shape Renderer** (or another **Renderer**) with a **Liquid Shader** material assigned to it.

2.1.2 2D Renderer

The 2D Renderer is the default URP Renderer when creating a 2D project. Please use the **/2D Renderer** folder exclusively for this Renderer.

Setup

The setup is the same as described in subsection 2.1.1. The only difference is that you should select **Project G.A.M.E.D.E.V./Essential 2D Shaders/Liquid Shader - 2D Renderer** as the shader source for your materials.

Shader options

The shader options are a subset of 2.1.1. Keep in mind several options are not available on the 2D Renderer as they are not supported there, namely **Ambient Occlusion**, **Metallic Map**, and **Metallic**.

NOTE

Since the 2D Renderer combines emission and albedo color in a single channel, keep in mind that your materials will look differently on the 2 different Renderers.

API

Similarly to the shader options, the API for the 2D Renderer version of this shader is a subset of 2.1.1, with **Ambient Occlusion**, **Metallic Map**, and **Metallic** missing.

Liquid Sprite Shape Geometry Creator

Please consult with 2.1.1. The Geometry Creator is independent of the Renderer – the very same assets are used on both.

Quick Liquid Setup

Please consult with 2.1.1. The component is likewise independent of the Renderer, and only writes the properties that the material it finds actually has.

Section 3

Support

In case you experience any problems with the shader, can't manage to set it up properly, or would simply like to propose an improvement, don't hesitate to contact us at contact@project-gamedev.com, or fill out the form at <https://project-gamedev.com/contact.html>.

If you are satisfied with this product, please consider leaving a review on the Unity Asset Store page (<https://assetstore.unity.com/packages/slug/402766>).

Best of luck with your game development journey!