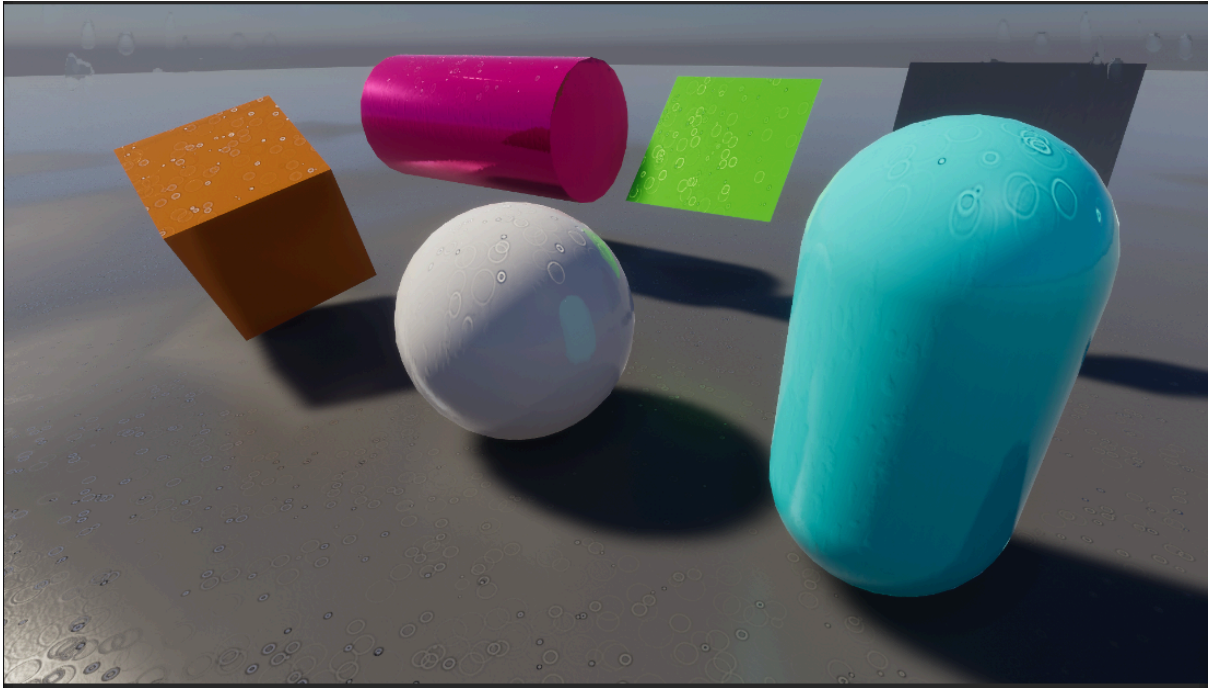


ScreenSpace Rain Wetness

Documentation



Important disclaimer

Unity version : Unity 2023.2+

Demo use raytracing, DX12 as default graphic api

Explanation :

ScreenSpace (fullscreen) RenderPass, why?

I choosed to use **Screen Space Rendering** (using Fullscreen Pass) to render few elements like RainDrops and to render directly my effect to the Normal Buffer and Smoothness buffer, this way I can write the Rain Droplet and Rain Gliding Effect to any Object in the scene, taking in account the original smoothness and normal value of each object.

Tricks inspired from : <https://github.com/alelievr/HDRP-Custom-Passes> (cf Screen Space Wetness), the smoothness and normal effect from the Rain/Wet can be discarded using various method like textures or vertex painting.

Optimisation : Reading and Writing the NormalBuffer is far more fast than executing the effect on each Shader/Materials in the scene, convenient effect for all environment assets

Artist friendly : Rain/Wet Effect can be mitigated across the scene using VertexColor directly using Polybrush for now, but we can absolutely allow artiste to create special texture to specify where the effect should be apply and where not.

One click setup : Drag and Drop the **FullScreen Wetness Controller** to wet any scene and add raining effect.

Using **Shader Graph**, **Custom Pass**, **VFXgraph** and **Custom RenderTextures**

The **RainOnCamera** effect

Is just for fun, it is a modified version from the HDRP fullscreen samples :)

VFXgraph to buffer vs Custom Render Texture

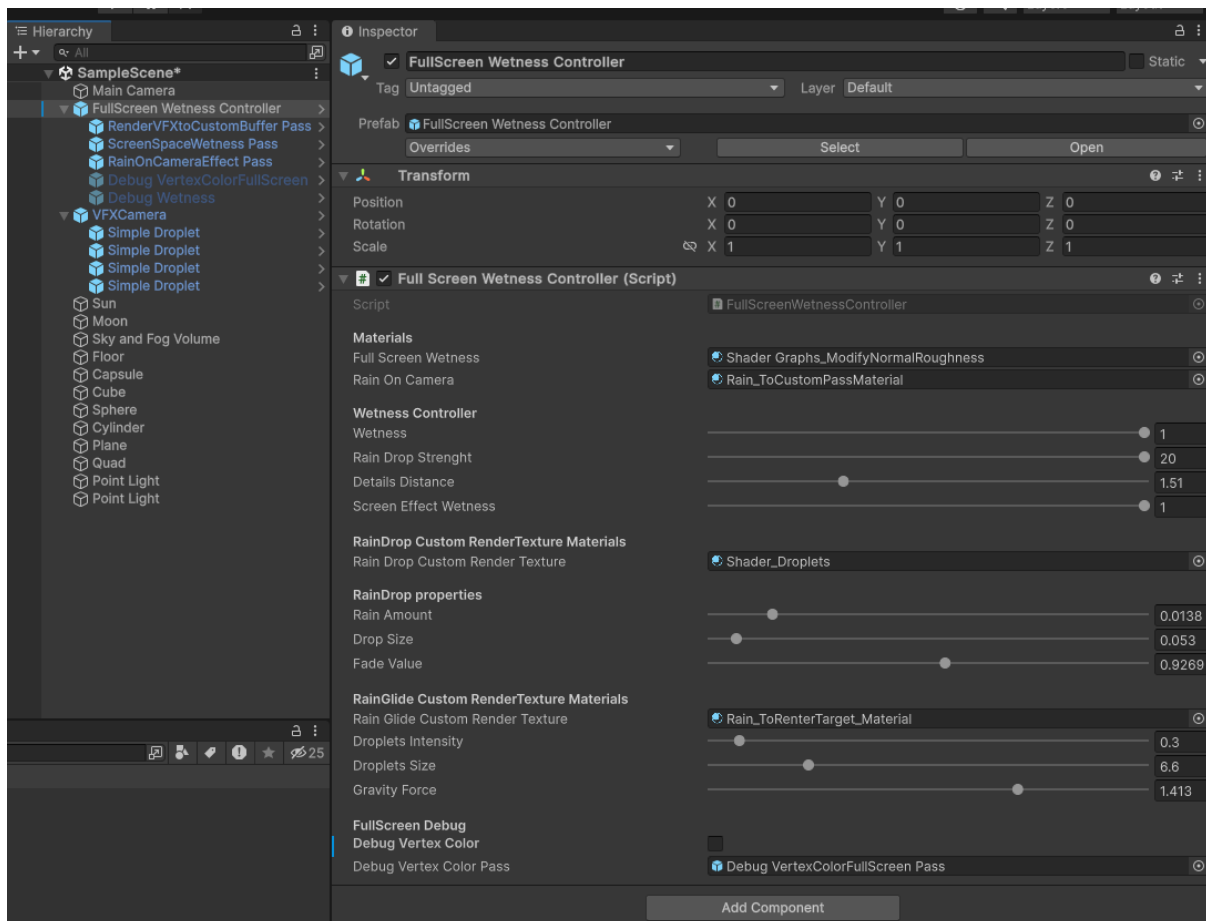
Computing a RenderTexture is faster than Rendering a VFX into a RenderTexture and can demonstrate advantages for many aspects like the effect tilling because noise math are more easily scalable

Using the VFXgraph I choosed to record 4 “flipped” (using rotation) VFXgraph to create some tillable raindrops by default.

Get Started :

Open **SampleScene** in Assets/Scenes

Look at the **FullScreen Wetness Controller** to control over the rain and wet effect



Content :

Enable or disable each effect using the **FullScreen Wetness Controller**

Main components :

- **FullScreen Wetness Controller**

GameObject saved as prefab in Assets/FarFromHereStudio/CustomPasses, It have the main control we need to apply our Wetness effect.

It control over the **ScreenSpaceWetness Pass** and the **Custom RenderTextures** Shaders

- **ScreenSpaceWetness material**

Material (fullscreen) responsible for the CustomPass which is *encoding* the HDRP **Smoothness** And **Normal Buffer** to add Wet/Rain effect to all objects.

Create the **VertexColor** pass to draw as **Custom ColorBuffer**

Use the **VertexColor** pass as mask to hide the Wet/Rain effect and paint across the scene using polybrush, alternatively we could use some textures or additional computing to define this mask over the scene.

- **Custom RenderTexture : CustomRT_Droplets**

This Rendertexture is 1024px, RGBA Sfloat32, used as trilinear projected textures by the into the NormalBuffer by the **ScreenSpaceWetness material** to create the droplet ring on the top of each object.

- **RainOnCameraEffect material**

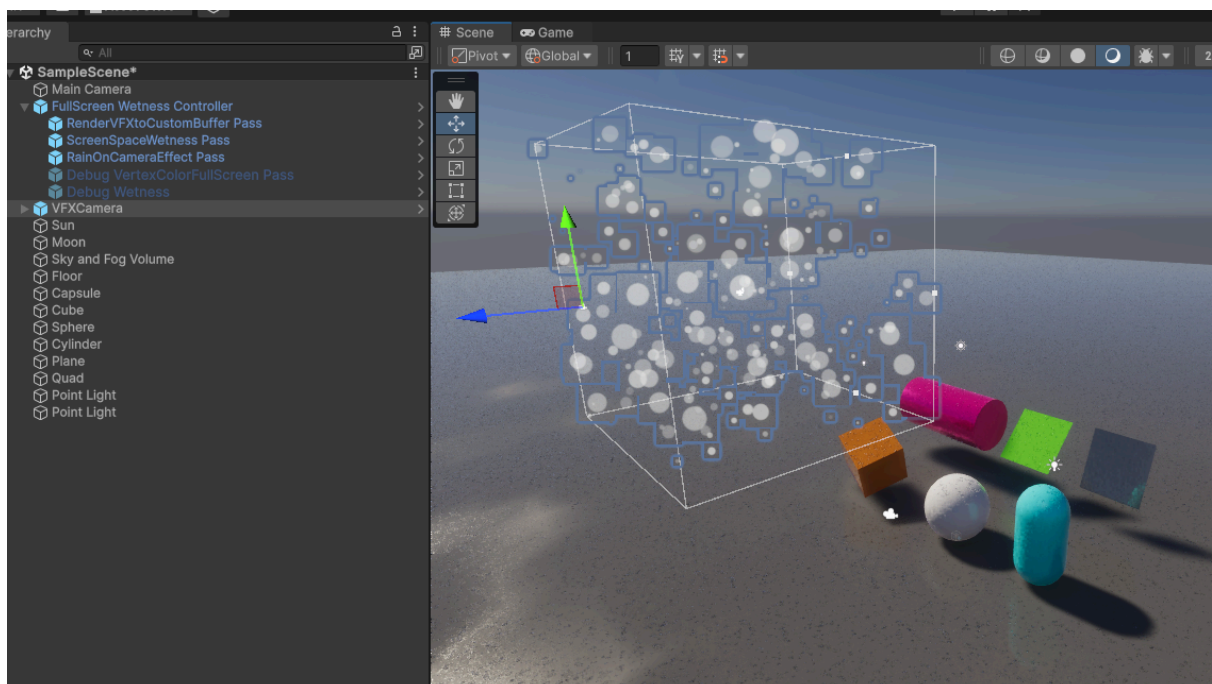
Material (fullscreen) Add a screen Effect to reinforce wetness feeling

- **Custom RenderTexture : Rain_RenderTarget**

This texture is 512px, one channel, used as trilinears projected texture use by the **ScreenSpaceWetness material** and the **RainOnCameraEffect material** to create the Rain Gliding on each object sides

- **RenderVFXtoCustomBuffer Pass**

RenderPass responsible to draw our VFXgraphs into our **VFXtarget** (rendertexture), is using the orthographic **VFXCamera** (disabled camera) properties to generate the buffer POV and cullings parameters.



this VFX setup is a simple usecase that we can manipulate in scene view (cf the Main Camera culling mask and disabled camera), each VFX are children to the VFXCamera gameObject so it stay aligned to this buffer "bouding box" which is 3x3 meters wide

- ***Debug VertexColorFullScreen Pass***

Debug the Vertex Color area painted using Polybrush to display where the wet effect is apply