Modern 2D Shadows Documentation

PLEASE READ BEFORE USING THE ASSET!

Table of Contents

ELEMENTS SETUP

Unity Graphics Render Pipeline

Currently Modern 2D Shadows Shaders work in Universal Render Pipeline for 2D only.

If you don’t know how to set it up, check out this 1min. YouTube tutorial:

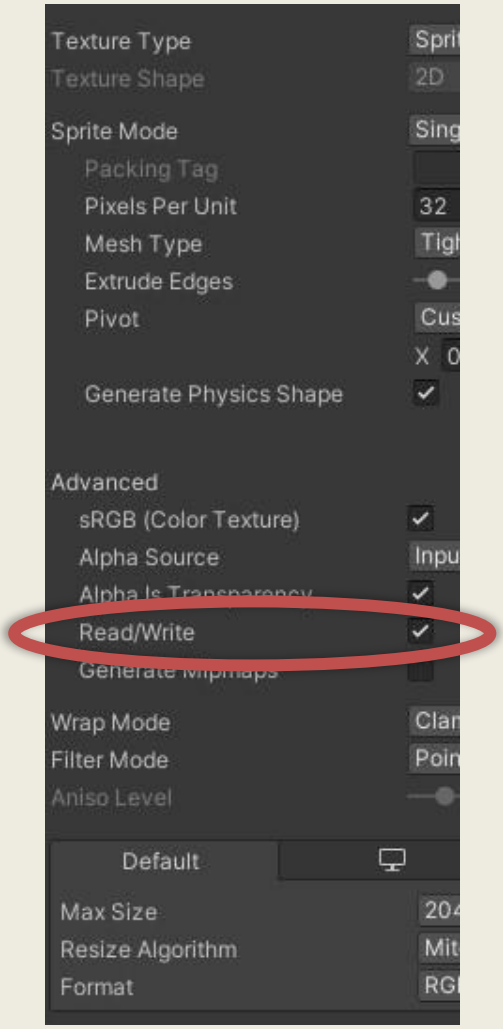
https://www.youtube.com/watch?v=pjpcitJim04

I’m not sponsored btw.

Build-In renderer compatibility is in consideration, tell me if it’s important for you, so I will know how important it is in development order.

Texture Import settings

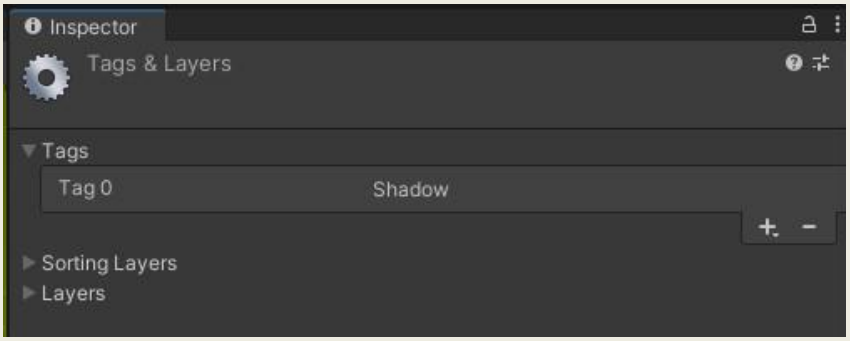
To create System Shadows, you must make your textures readable in scripts:



Layers and Tags Setup

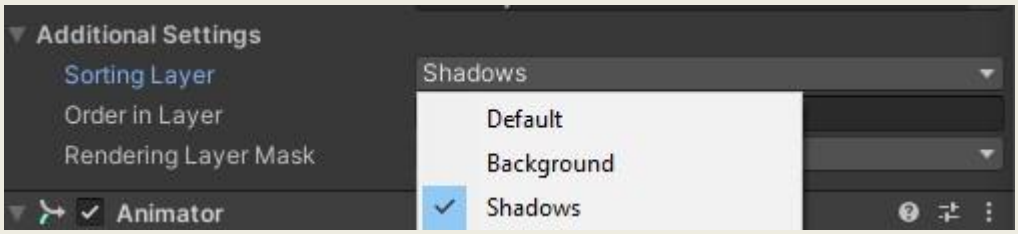
- NOW LAYERS AND TAGS SHOULD BE CREATED AUTOMATICALLY

To start working with the system, you must add Shadow tag:



*Used for finding shadow holders in the scene

And then add SHADOWS sorting layer in the SPRITE RENDERER:

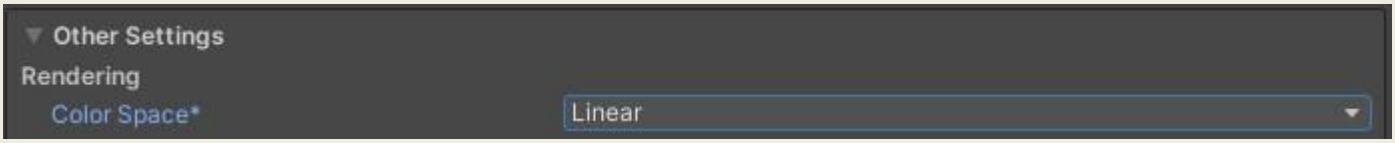


*This sorting layer will be used for our generated shadow spriteAnd, done, now the project and the shadow system should be set up properly.

Player Settings!

If your scene preview looks like this:

You must change your Colour space settings from gamma to linear! (they're in the project settings)



Adding System Shadows

System shadows are directional shadows that can be edited in real time, including things like: Direction,
Floor angle,
Ambient Colour,
Falloff,
and more.
They are generally good for 2.5D games, with objects drawn in perspective.

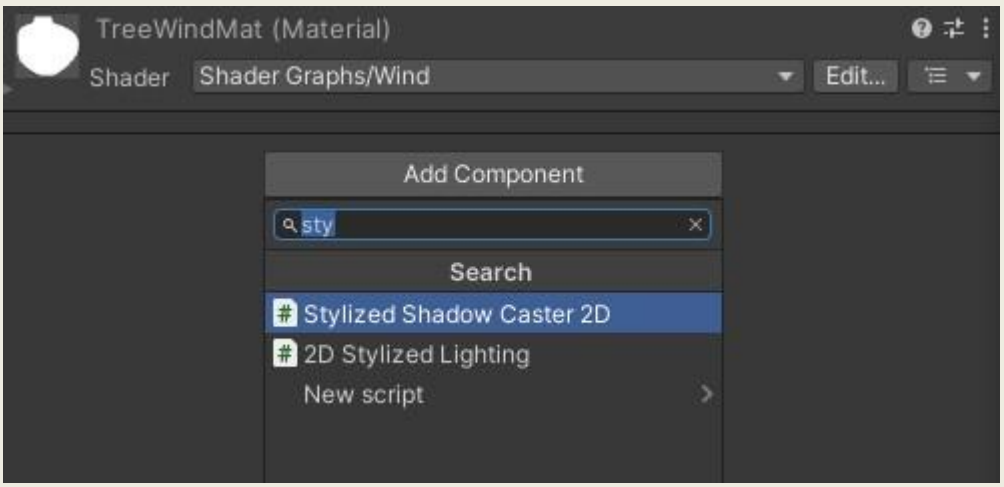
Before adding System Shadows:

After adding System Shadows:

To add advanced directional shadows, that can be customized at real time, and respond to a moving light source, first we must put in the scene the 2D Stylized Lighting System, that will take care of them. (It's in prefabs folder)

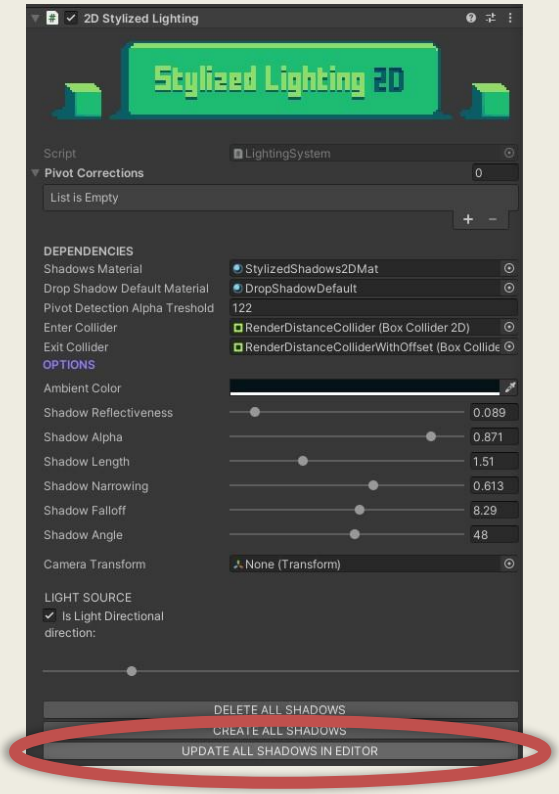
To use the system, you must add the camera transform (for updating shadows close to the camera at real time)

Now the only thing left is to add stylized shadow casters to sprites that we want to10/08/2022 cast shadows.



Now, you must be wondering why after building the shadows, they're not displaying.

Every time you want to update the shadows look; you have to click the "UPDATE ALL SHADOWS IN EDITOR" button in the 2D Stylized Lighting System;



Now you should finally see you shadows!

All there's left is to tune them to your liking and making your game look 3x better!

Adding Drop Shadows

Drop shadows are perfect for 2D and 2.5D games alike.

I like to combine them with the system shadows, by adding them to objects that would look weird/unnatural with system shadows.

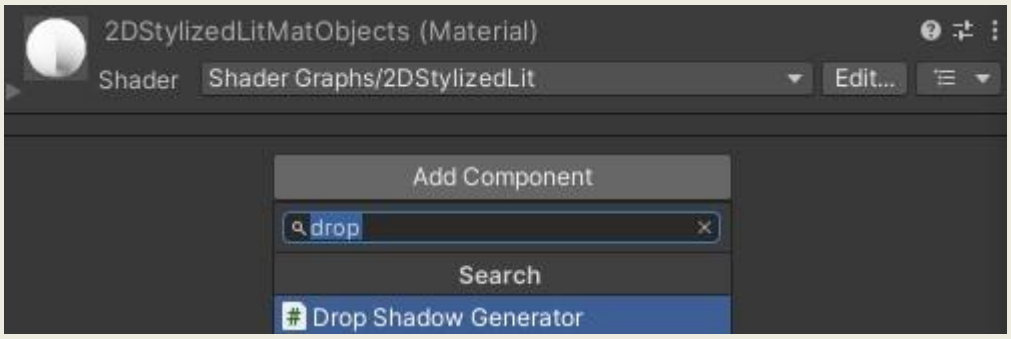
It's also important to note that they are not updated in real time, so you won't be able to edit them after starting the game (in contrary to the system shadows).

Before adding the Drop Shadow:

After adding the Drop Shadow:

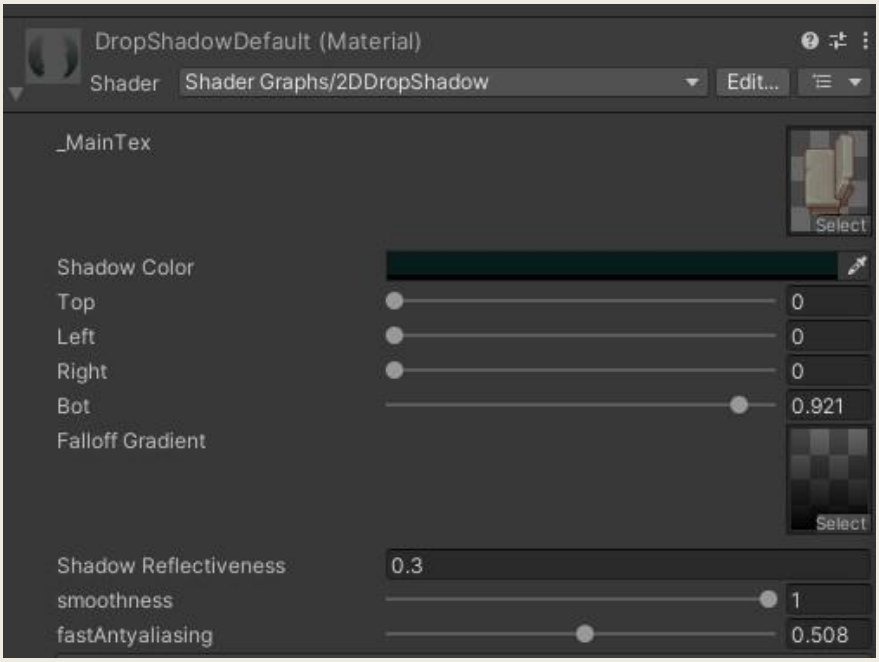
First, we must put in the scene the Stylized Lighting System, that will hold a reference to their default material.

Now, let's add the Drop Shadow Generator to an object with sprite renderer, that we want to have a drop shadow.



Then after generating the shadow, just scale it, put it in right place, control the soft edges with the Drop Shadow Material, and you're done.

Drop Shadow Material Variables:



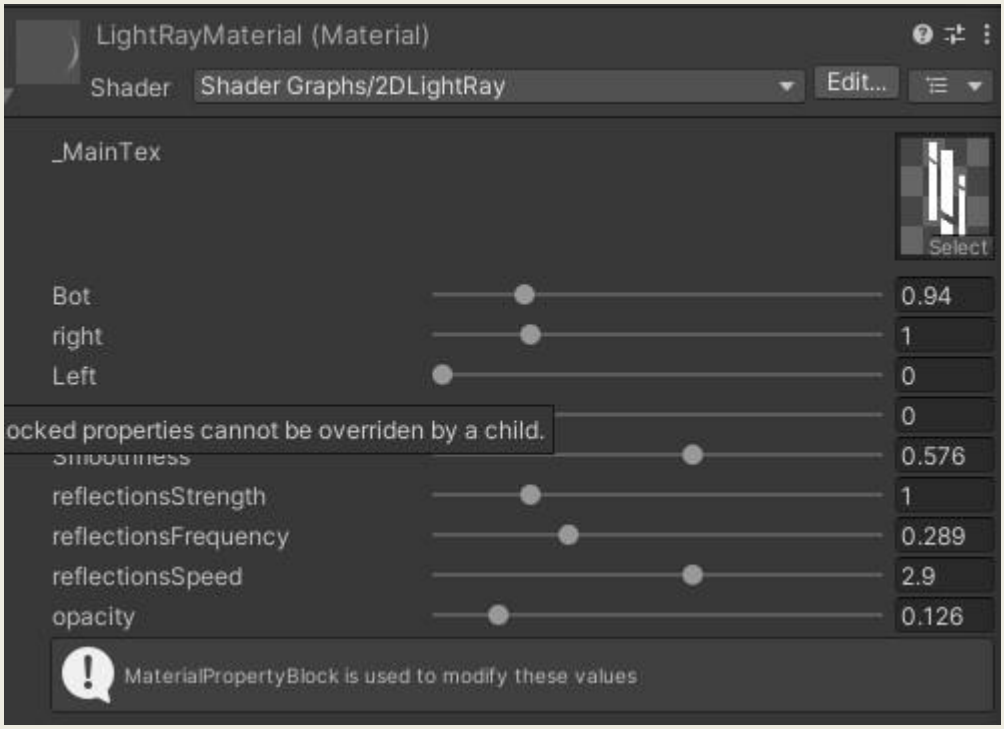
Just play with them until the shadow feels right.

Adding Light Rays

Adding light rays is easy, all you must do, is to drag and drop their prefab.

I would also advise you to make your own light ray textures that will suit your game shape design, as it's a quick process that can incredibly boost your game looks.

Of course, you can also edit a lot of different light ray properties, in their shader.



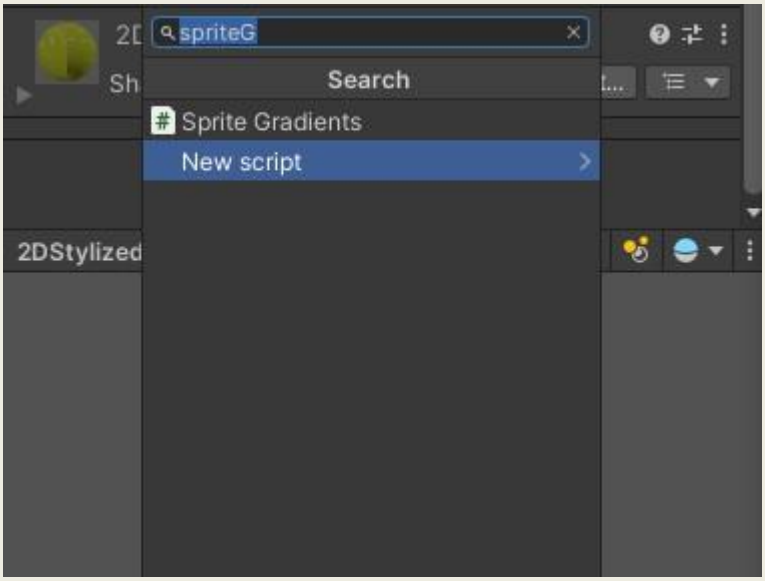
Adding Sprites fake Rim Light Generator

Sprites fake Rim Light Generator is a tool for adding coloured gradients for each side of your sprite via shader.

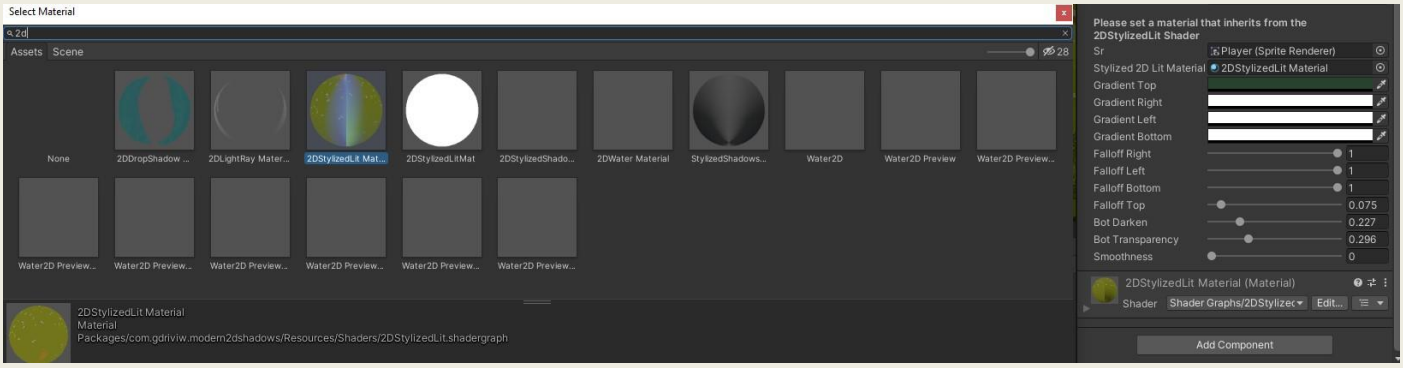
It can be used to blend your sprite with surrounding/ make it stand out/ fake local light and with many more scenarios.

Setting it up is easy.

Just add Sprite Gradients script:



And set the 2D Stylized Lit Material and your sprite Renderer



Now you can play with the values until it looks good.

Adding “Fake Water”

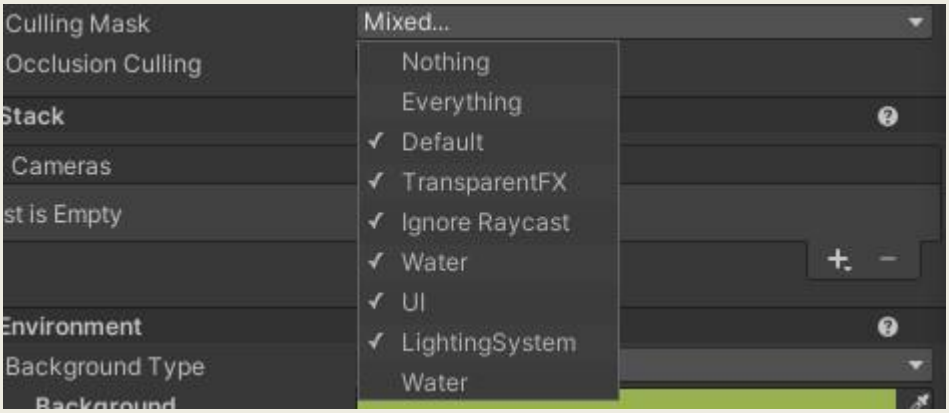
Fake water is a useful tool for nice natural environments, but it’s situational and can take a lot of attention from other important elements, so use it with caution.

First, why does I call it fake?

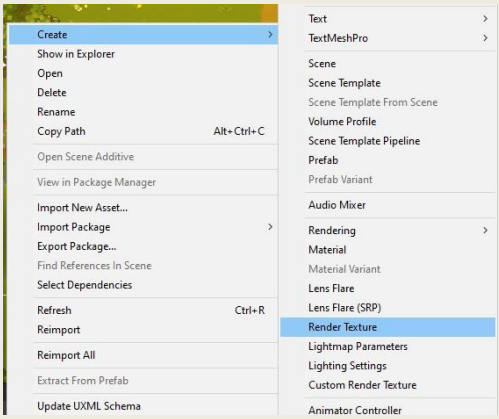
It’s because this water doesn’t reflect objects and have a lot of limitations. It copies the camera render texture and projects it with an offset, then its ads a ton of effect to give a feeling of a “real” water.

First add Water Camera prefab and assign those settings:

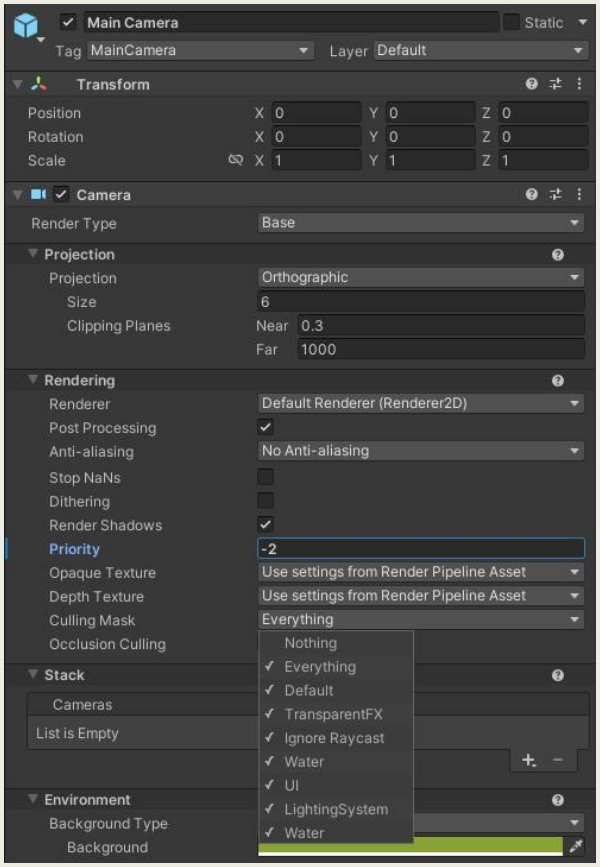
- Make the size the same as your main rendering camera - Create a 2DWater layer and exclude it from rendering



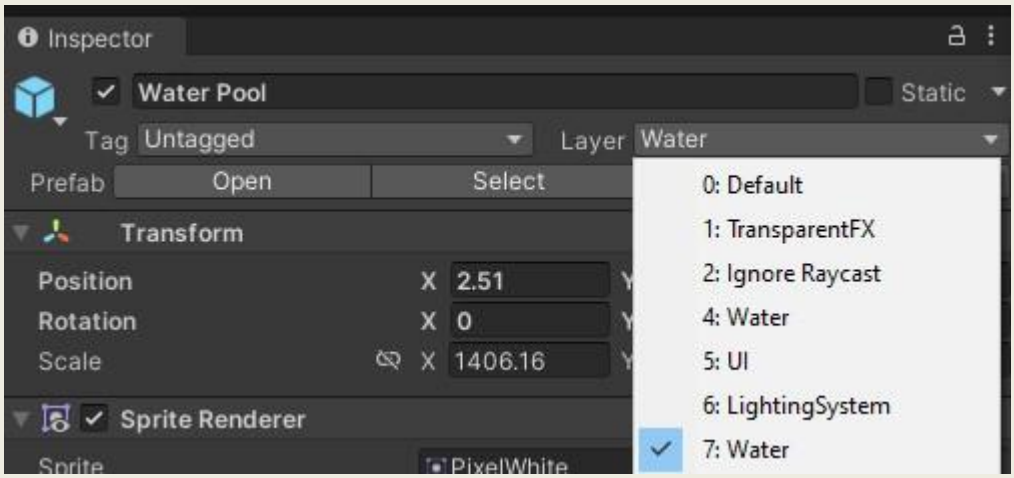
- Create and Assign a Render Texture that will hold the camera rendered images

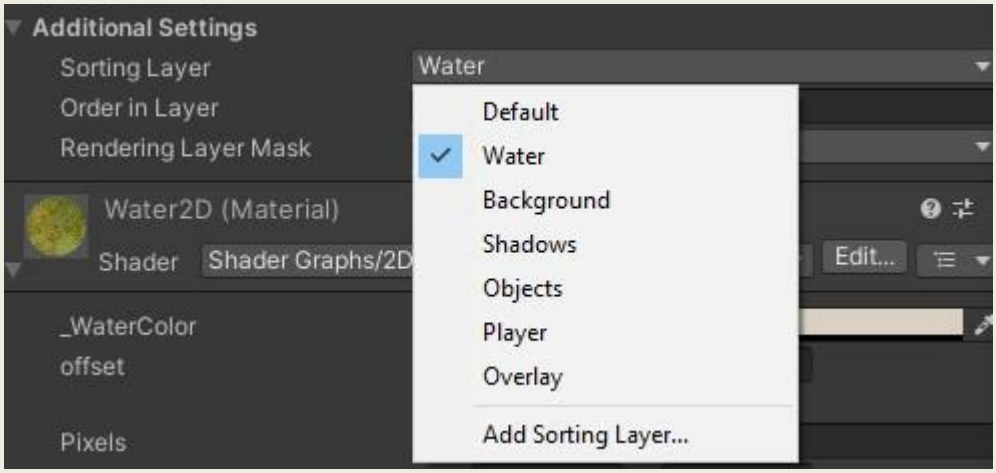


- Create a second camera (or camera that existed already before adding Water Camera) that will render the final image + our water, and include all layers in the rendering



Now all left to do is to add the pool of water from prefabs, set its layer to “2DWater”, and layer it properly, so it’s behind your environment.



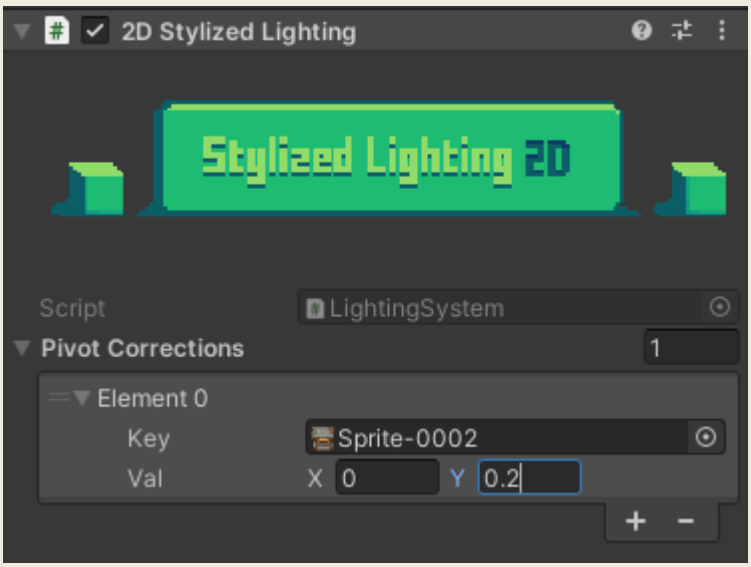


And done! Have fun playing with settings!

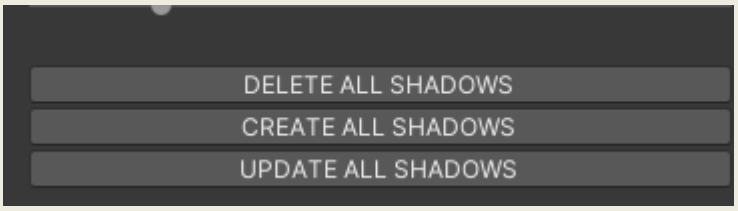
Shadow Pivots (#Update 1.1)

In case your sprite shadow looks a bit unnatural like this:

You should change its pivot position, as yeah, the shadow origin is placed too near the edge of a sprite. But not by changing shadow pivot transform, but by the “pivot corrections” dictionary in my “2D Stylized Light Component”.



Now just delete and create shadows to reset them.



And you’re done: