Modern Water 2D Documentation

official documentation of this asset

If you have any problems or you don't understand something in the docs, you can write to me on my official discord server. It's a server for :

- bug reports

- feature suggestions

- update reports

- quick fixes

## Invite Link :https://discord.gg/GTBNdFg7hW

Adding Water To Your Project

Rendering Pipeline

Right now **this asset only supports the URP pipeline**, the most popular pipeline for 2d games.

So you will need to either : create your project with urp, or upgrade your already existing one from built-in to urp

! To **create a project with URP pipeline** choose 2d urp template in unity hub :
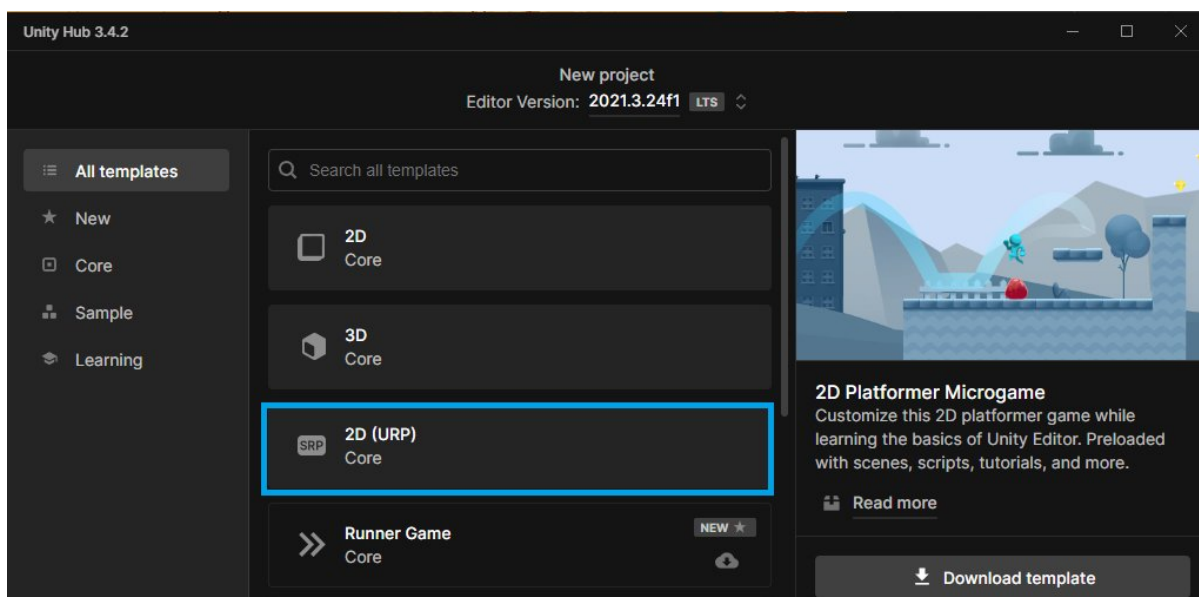
! To see how to **upgrade your existing project to URP pipeline**, watch this video :
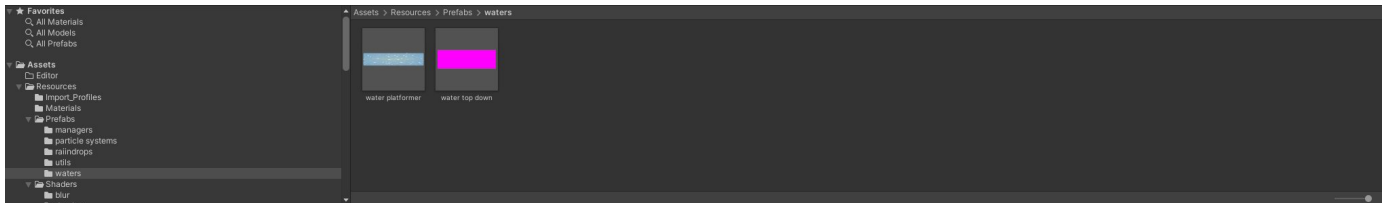
Unity 2020 and older

Unity 2021 and newer

Adding Water

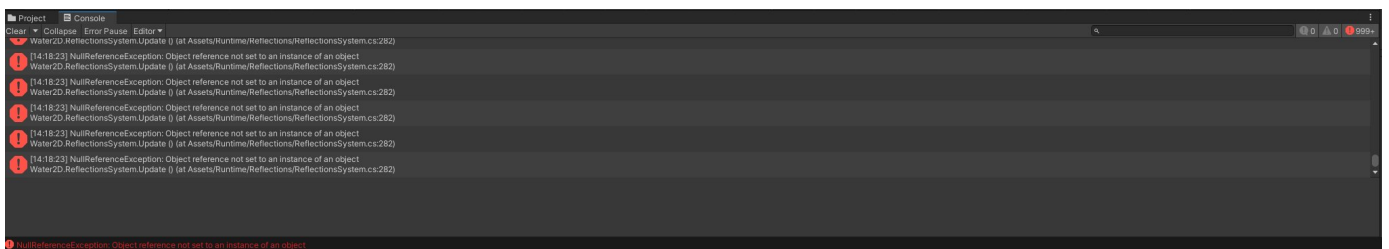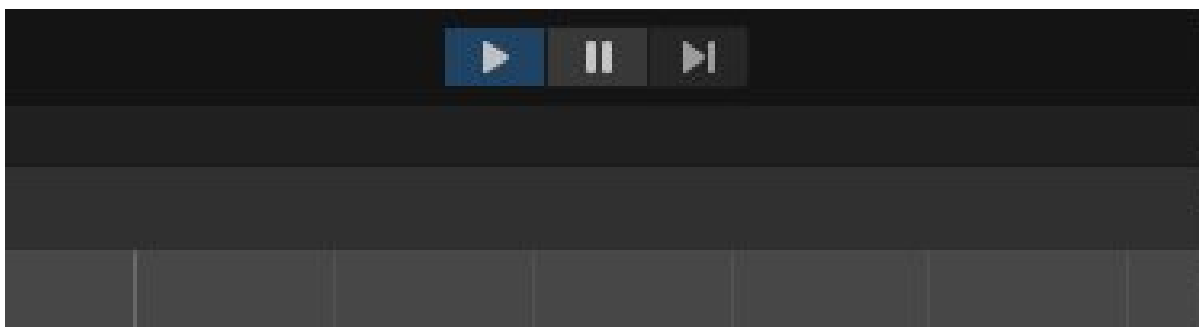Adding Water to your project is fast and easy

---

! Firstly, **go to the water prefabs folder** in *Modern2DWater* asset hierarchy

---

! Then **place the type of water you want** in your scene
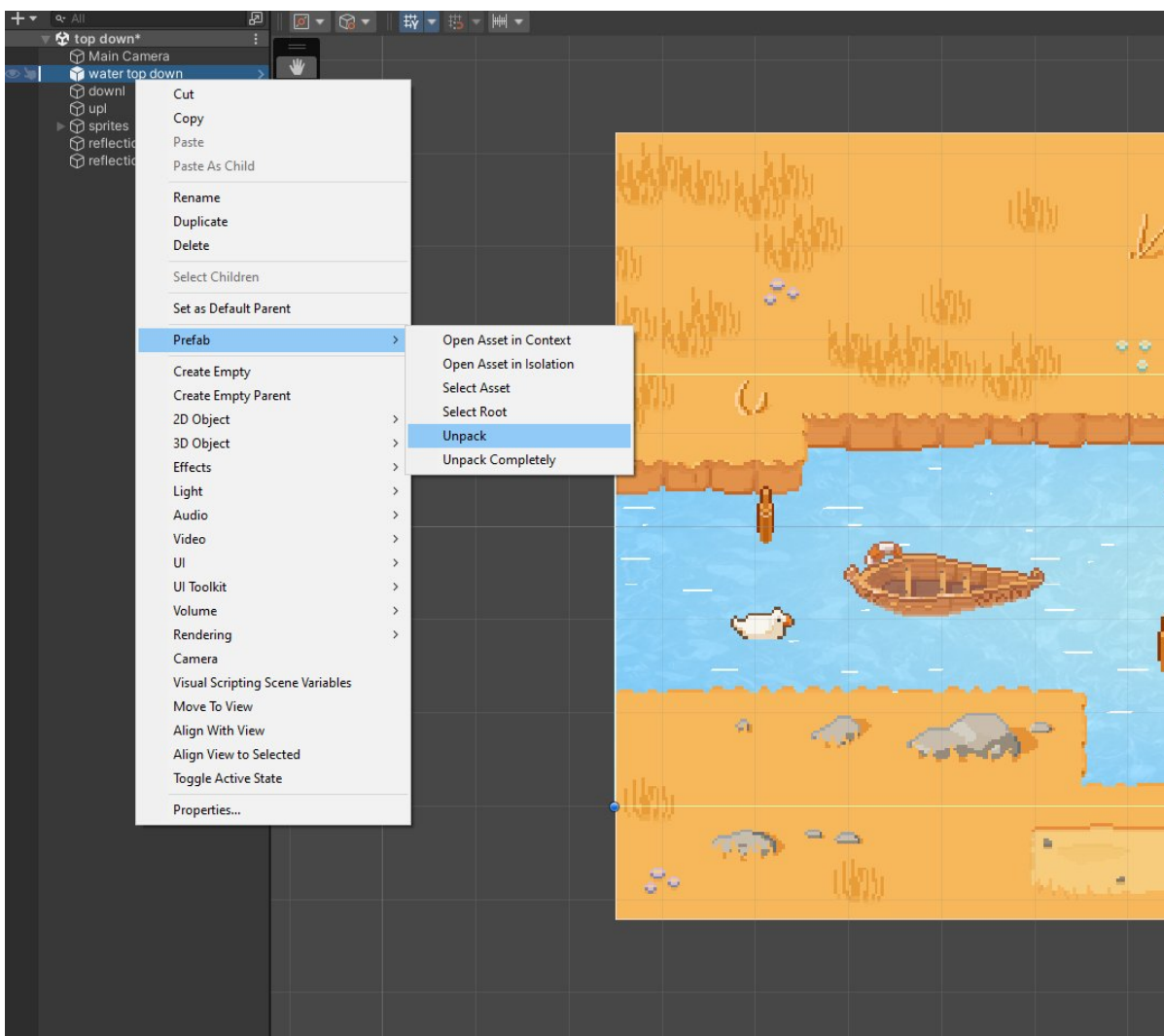

! If there are any null reference errors after that, **just click play.** These errors should fix themself after that.

! Now, for the future systems that will be added, we want to disable two layers in our scene view.
Disable Obstructions and Reflections Layer to have clean scene view.

The last step is to **unpack the asset**, since any changes made to prefab won't be saved.

Good job, now your water should look more or less like this. All that's left for you is to costumize it's

look or add some cool features listed below.

> We can improve this scene by adding **white foam** under objects (called obstruction in my asset)

> And we can add **dynamic reflections** to the objects in the water, or even **simulate** waves
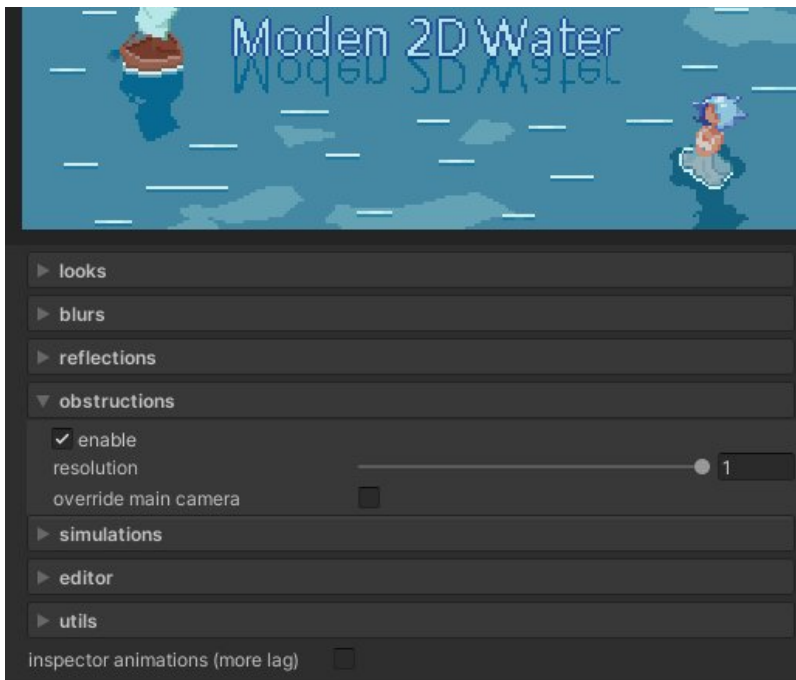
Adding Obstructions

Obstruction can improve the look of your game by adding white foam, and is used by water simulation covered below

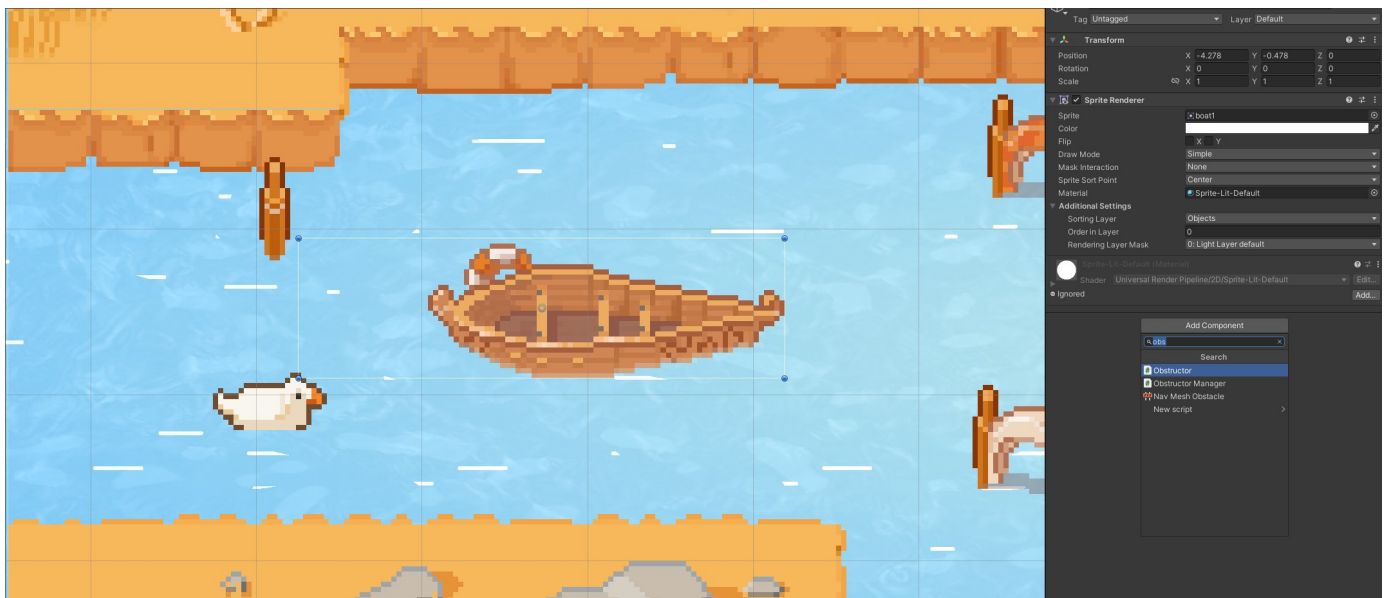> ! To Add Obstruction, **just check the checkbox in the obstruction category** of ModernWater2D
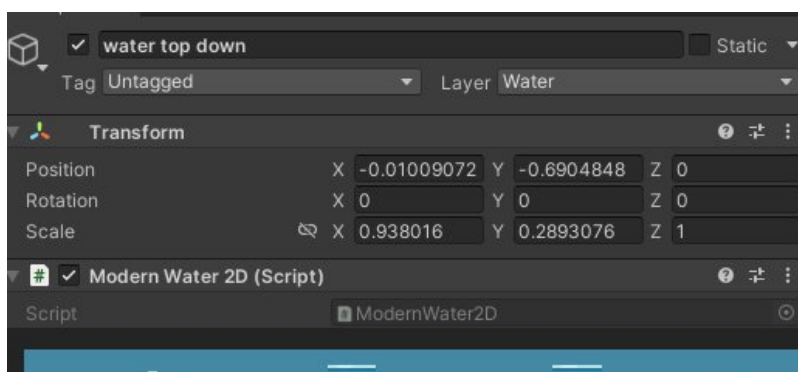
> ! You may notice that nothing changed. That's because **we still need to add a component to the sprites that will cast obstruction** in the water

> ! Click on a sprite that will be placed, or can find itself in the water and **add an Obstructor Component** (In this example we're using the boat)

> ! After adding the component, choose how much of the sprite will be in water. It's indicated by the **red line** in your scene window.

> Then **click Create** and you're done! A white border should be placed around your sprite when it's in water, but will be invisible if the boat gets on the ground.
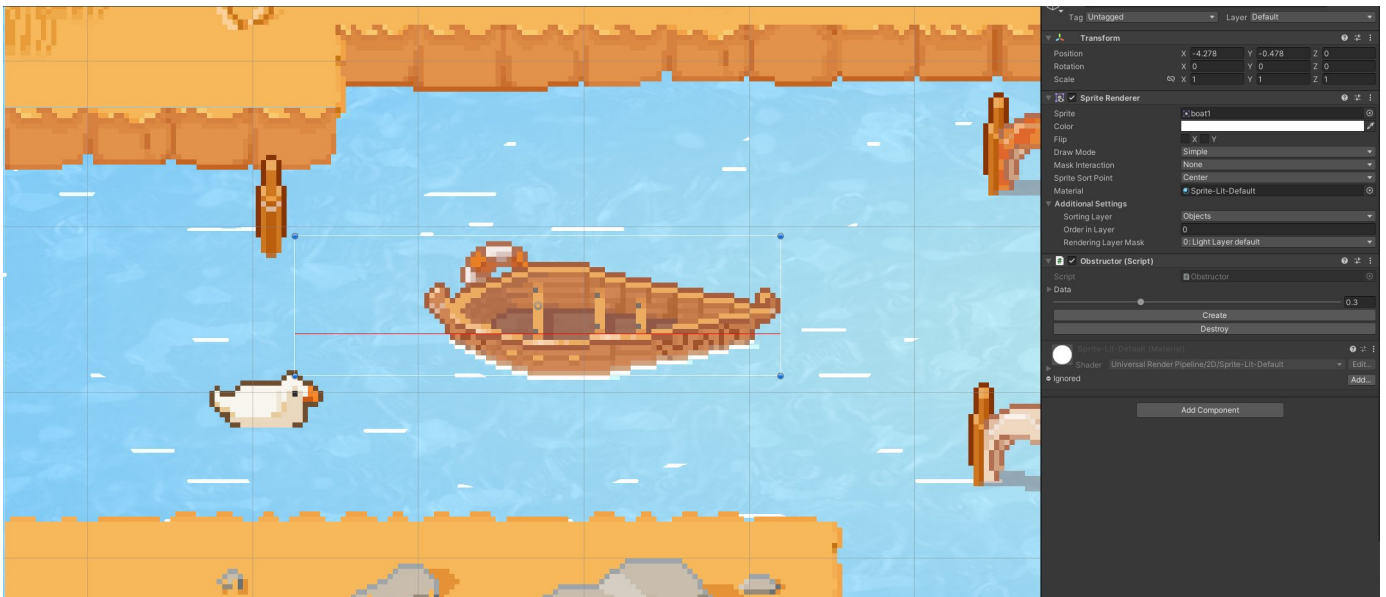
> In case you have a lot of sprites that are in the water, you can use multi editing to save time adding components. You can also write your own scripts that automatically adds the component to sprites you want.

> If the preformance is important and you have a lot of sprites, you can decrease the resolution of renderer obstructions in water settings

Adding Reflections

> There two main types of reflections :

- **Top Down Reflections** (they reflect every sprite in the water pool)

- **Plaftormer Reflections** (they reflect the whole screen at certain height)

> Note that platformer reflections are a bit faster since they don't need to create new reflections, and just copy the screen
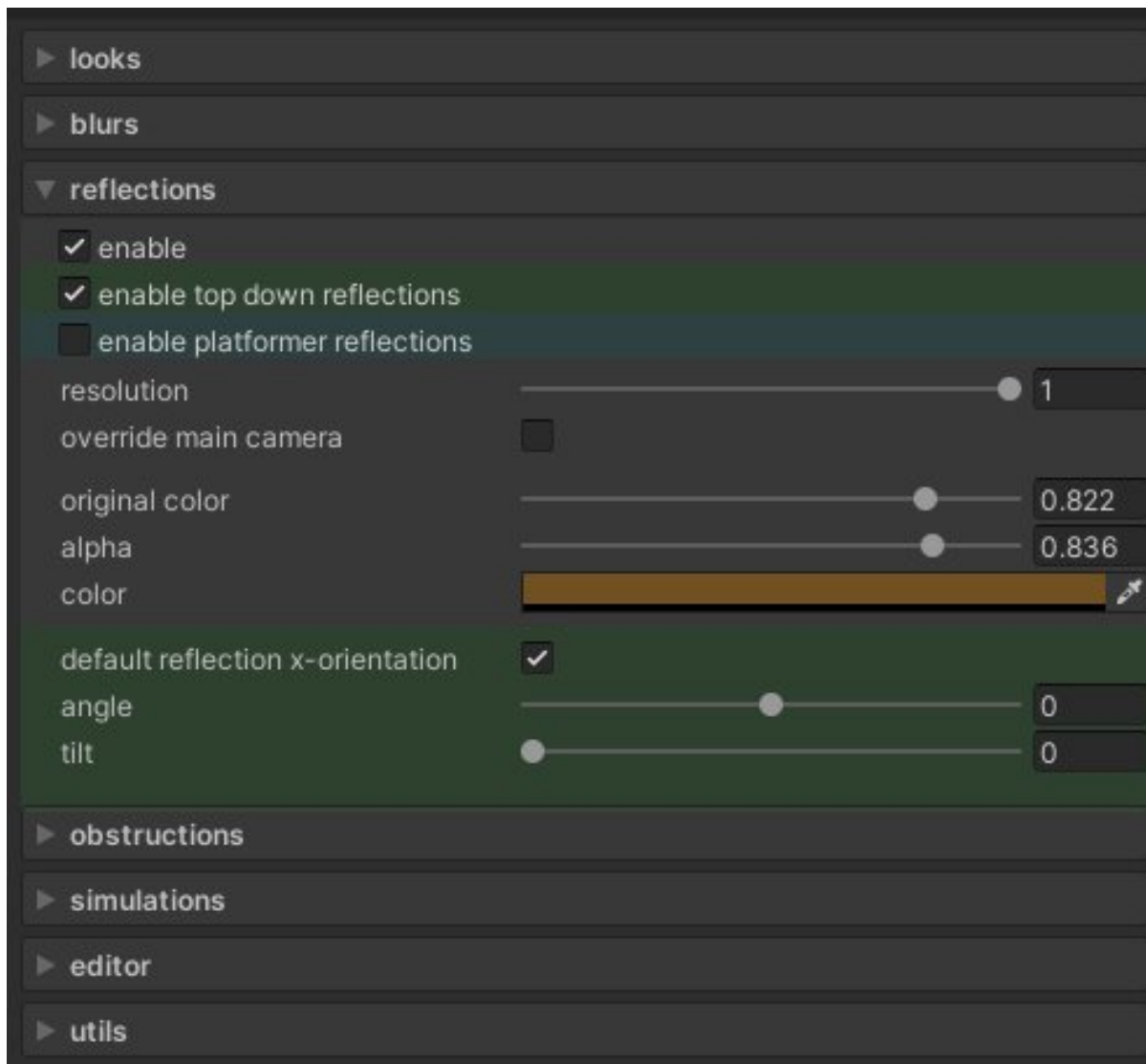
> **!** You can use both reflection systems at the same time to create an interesting scene. There is an example scene of this mix in the scenes folder.
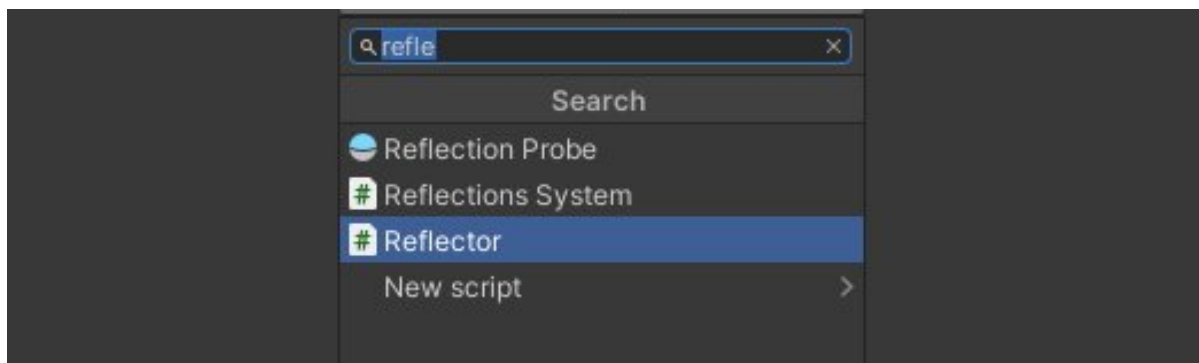
Adding Top-down Reflections

> **!** To add top down reflections, **we (again) need to enable it** in water settings
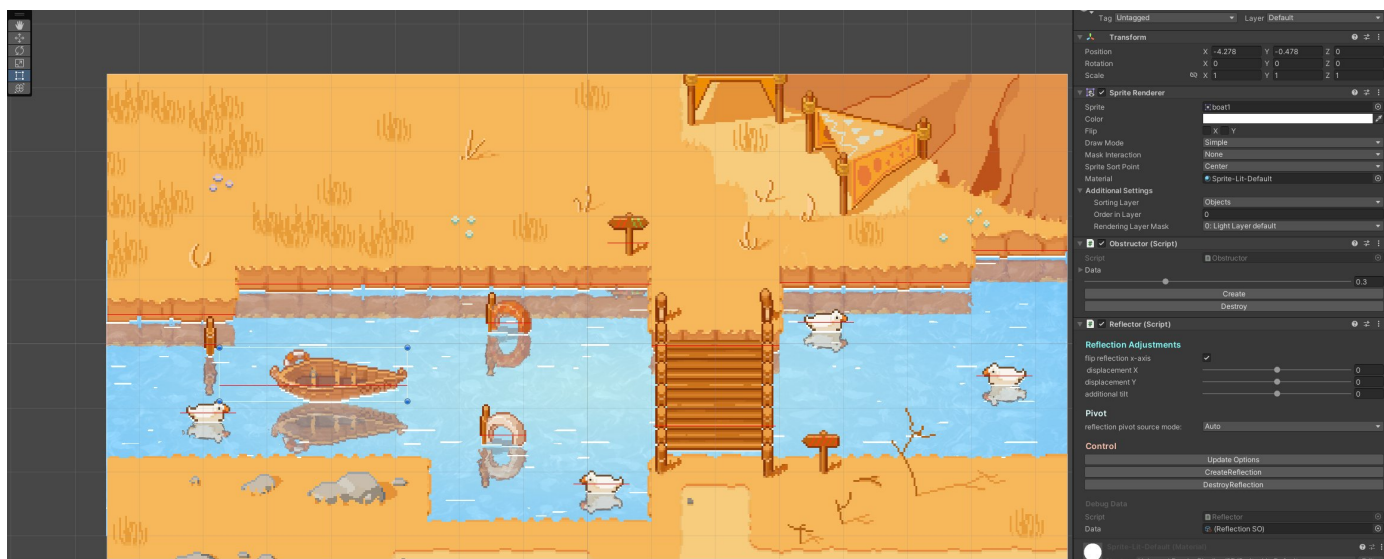

> **!** You may notice that (again) nothing changed. That's because **we still need to add a component to the sprites that will cast reflections** in the water

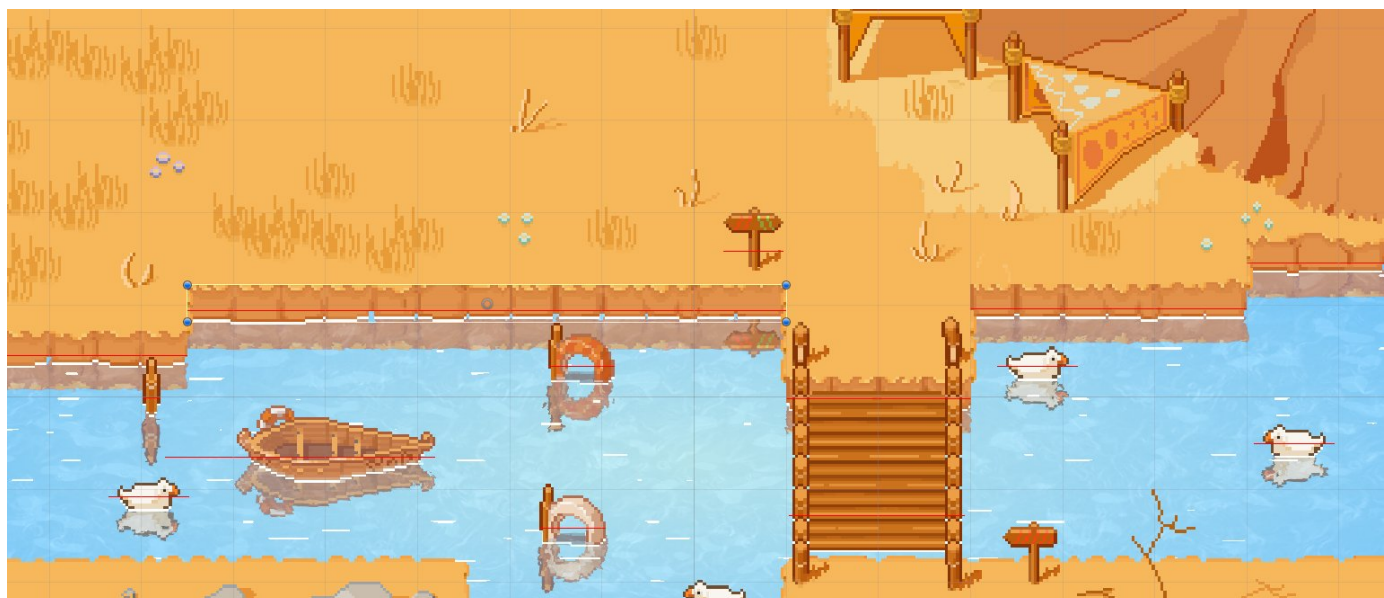> **!** Some of them may look weird. Reflections are made by rendering the sprite in an opposite way, so you may need to regulate its pivot (a place where the sprite touches the water).

Here we will move some of the reflections in the y axis using the displacement y slider in the reflector component.
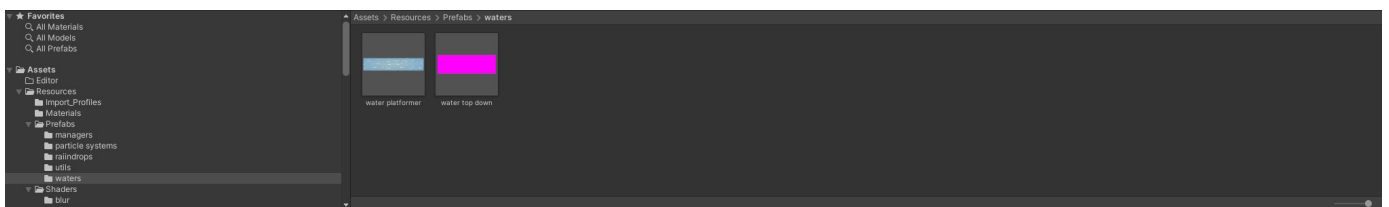
I also used multi-editing to speed up this process.

Adding Platformer Reflections

! For platformer reflections, **use the platformer version of water prefab, by moving it to your scene**, **and follow previous steps** *(in Adding Water section)* to get the water running

! Now make sure that the **platformer reflections are enabled** (blue color) and your **reflective layers** (layers of your sprites, which you want to reflect in the water) are set up correctly.
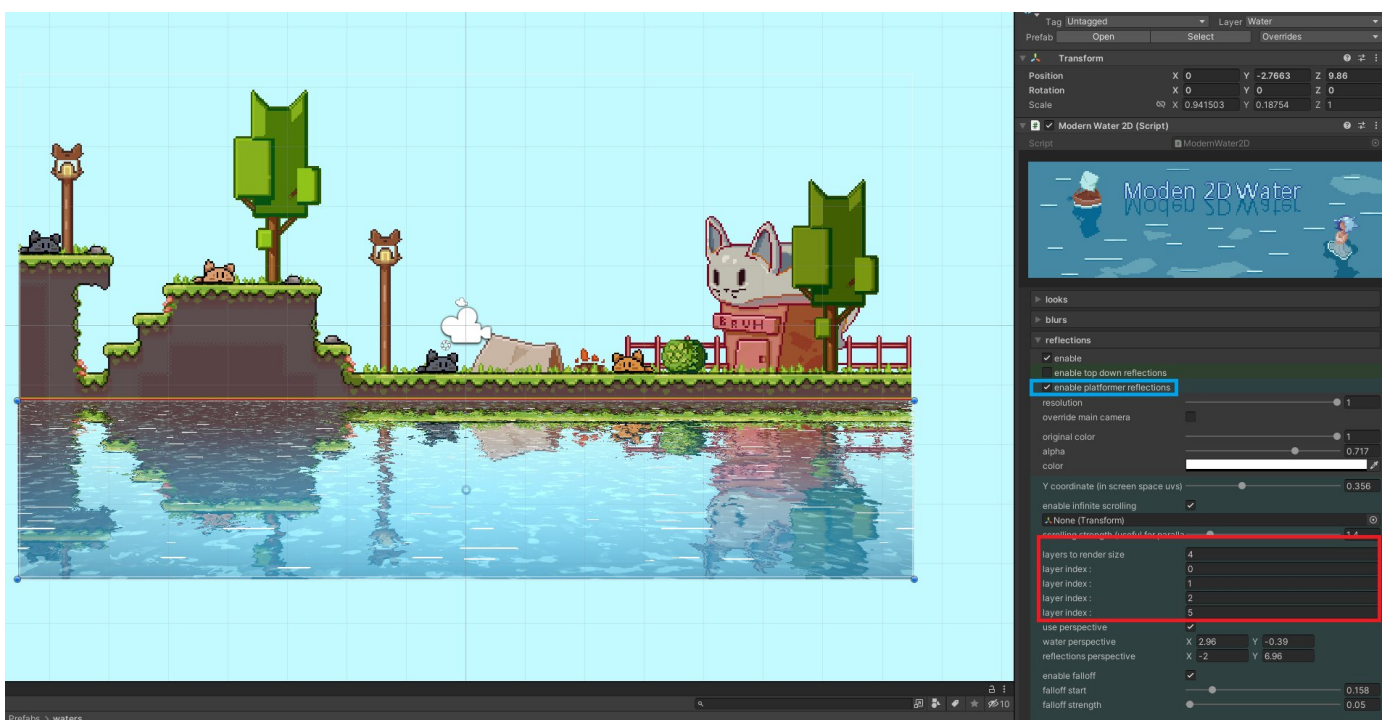
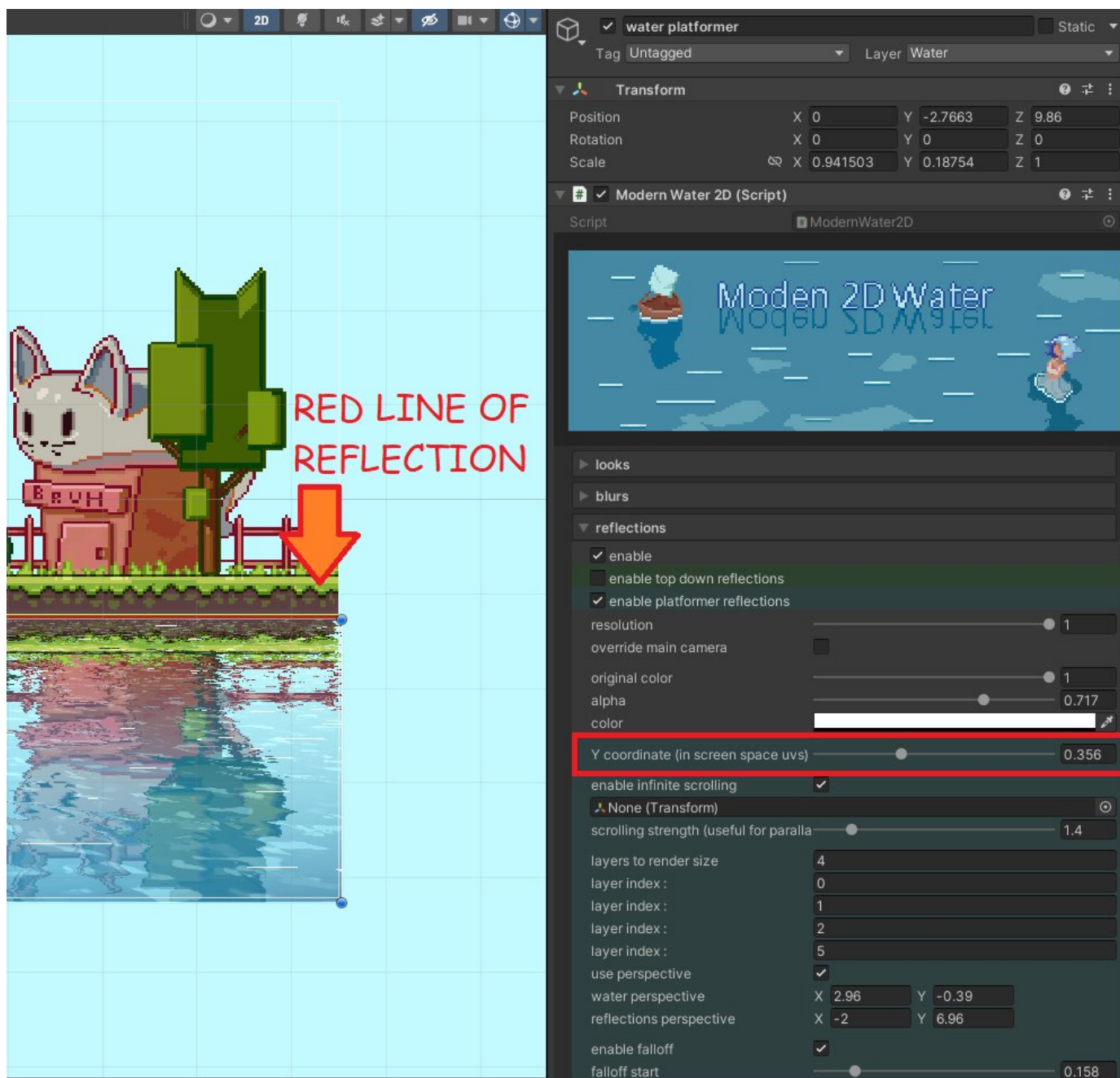! All there is left is to allign the red line to the edge of your sprites, and start of water. The slider for this is boxed in red.


note that contrary to top-down reflections, we don't need to add reflector component to every sprite in the game, whitch makes this setup a lot faster and easier.

Turning On Simulation

! **Simulation works by computing wave equation in the separate texture** and adding the computed normals (from height) to the water distortion field.

! The source of waves is the obstruction texture, so in **every object** (that you want to generate waves) **you need to have obstructor generated**. (and you need to have obstructions turned on in your settings) *If you didn't add obsctructions before, scroll up to the "Adding Obstructions" section and follow its steps.

! If you have obstructions set up already, just **enable simulation in water settings** and you're ready to go.
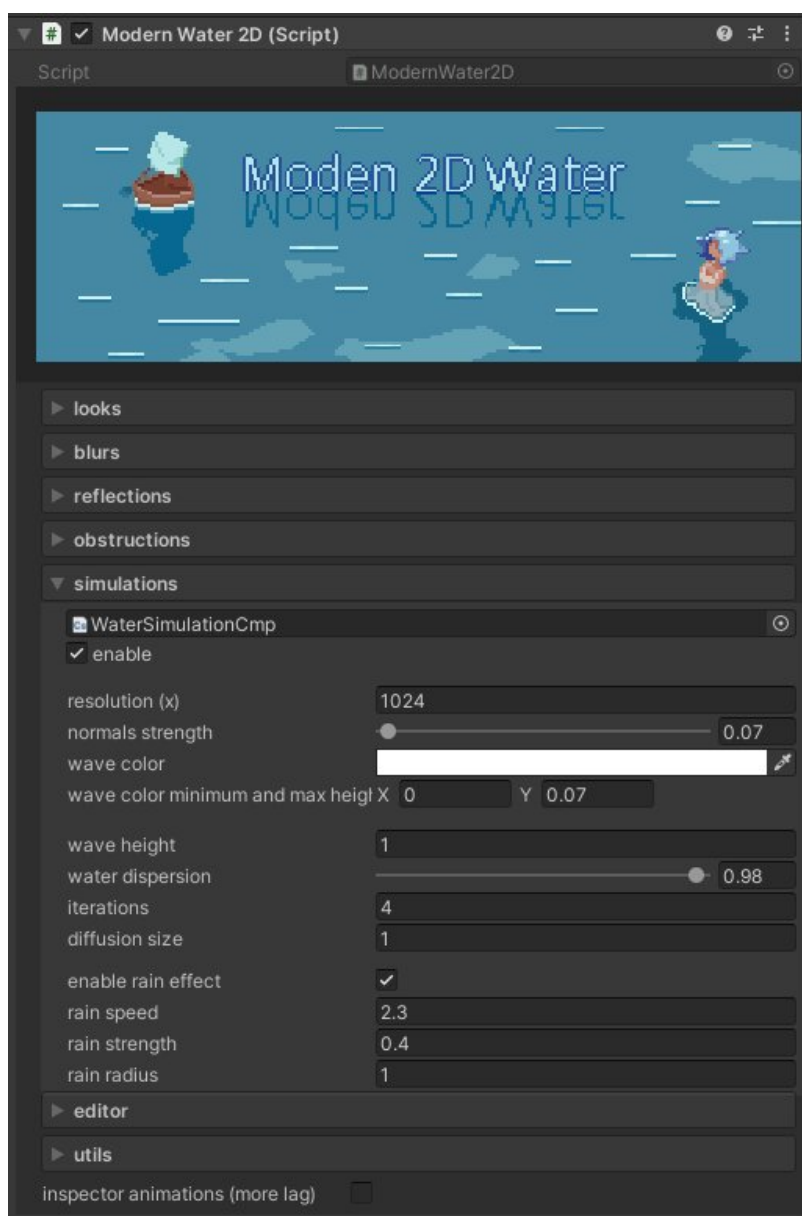
now play with the settings to make it look good in your project.

moving ducks create waves behind them

Demo Scenes

There is a lot of **simple demo scenes** in which you can see how the water is set up in certain game scenarios and even copy the water components.

The sprites, scene effects and particle systems are made by me, so, while I personally wouldn't do that, **you can use them in your projects**.
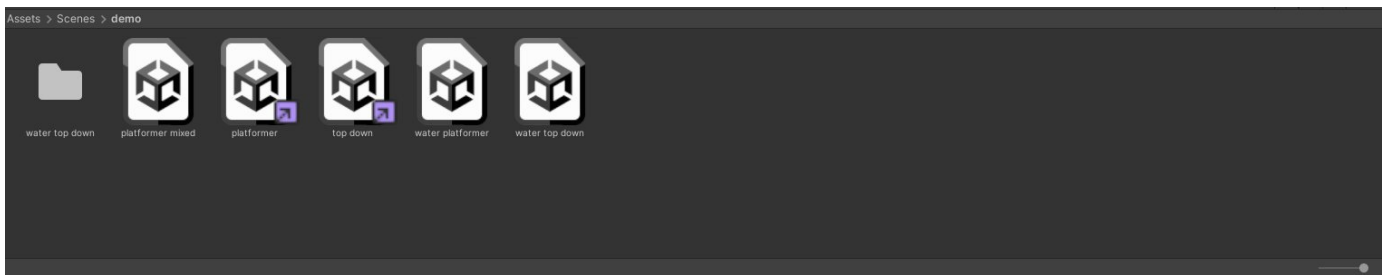
! I setup sorting layers in demo scenes with a c# script, but on some versions of Unity it doesn't work. In this case you have to setup those sorting layers by hand. Preview of how they should look:
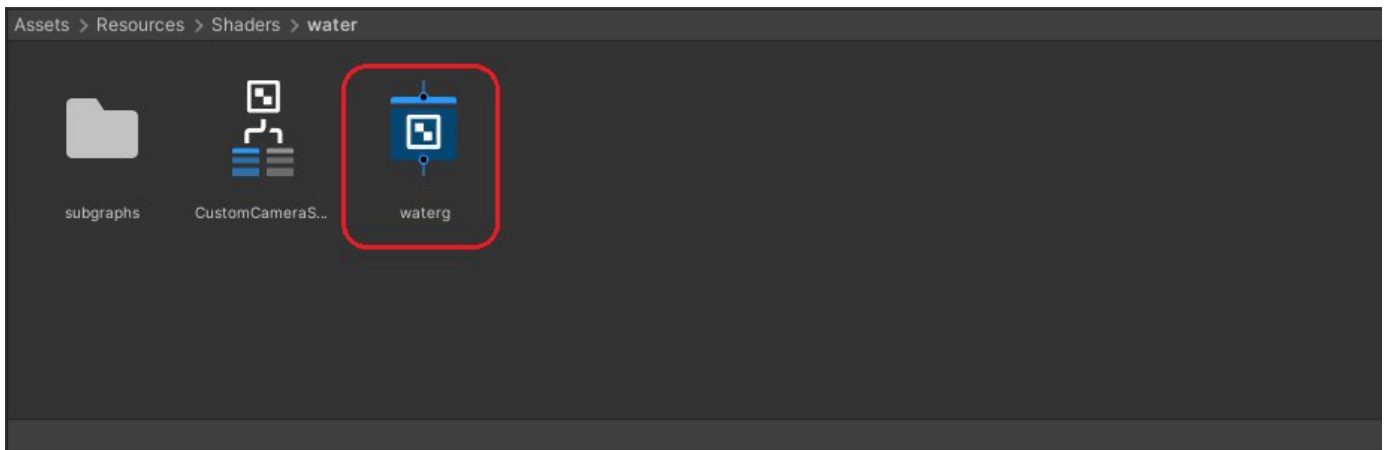
- Reflection_Plane
- plat
- Objects
- Reflection_Plane2
- droplets

^^ Demo scenes in **AssetRoot/Scenes/demo** path

Shaders

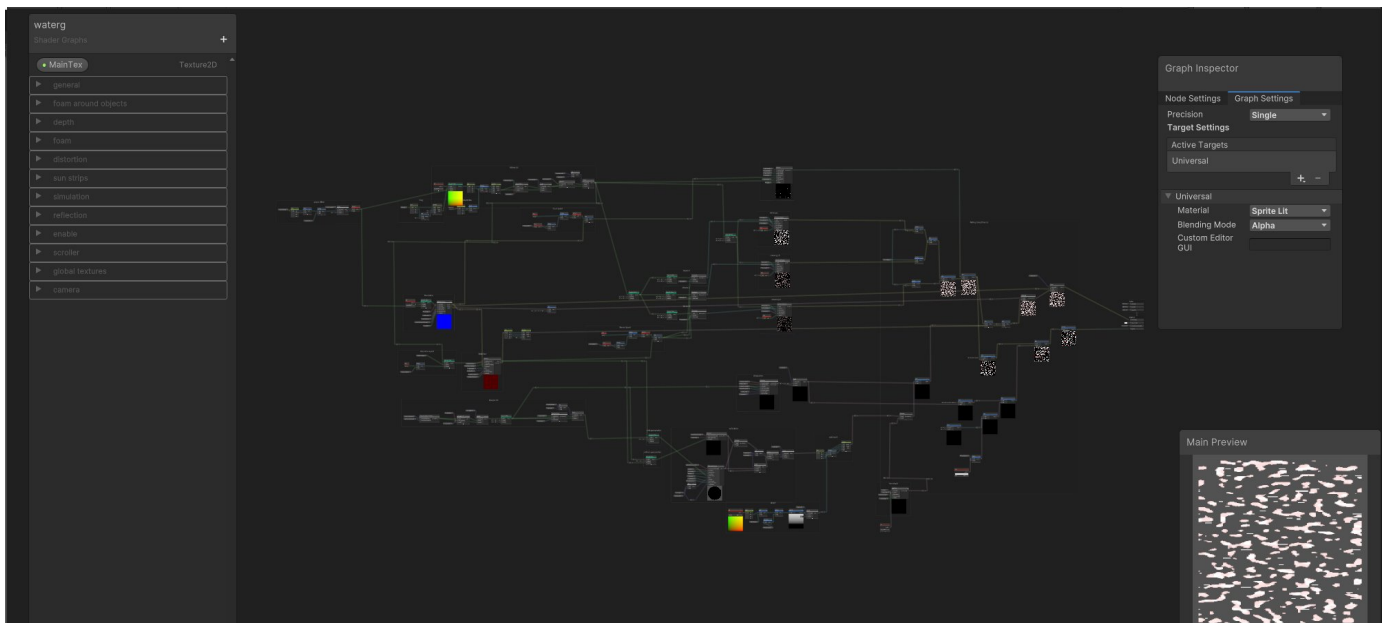> In order to significantly change how the water looks, you would need to edit the water shader

water shader location : AssetRoot/Resources/Shaders/water/waterg.shadergraph

Its cleaned up and divided into different subgraphs, so it's fairly easy to change certain elements without understanding the whole shader.

> shader zoomed out

- There are other shaders you can tune to your liking like :

≡ blur shaders : AssetRoot/Resources/Shaders/blur

> you can change 2d water blurs

droplets shaders : AssetRoot/Resources/Shaders/droplets

> you can change the look of rain post processing shader

water simulation compute : AssetRoot/Runtime/SurfaceSimulation/WaterSimulation

> you can change the inner workings of 2d water waves simulation

Optimizng Reflections

(all) reduce the size of the reflections system render texture (resolution slider in reflections section of water settings), as setting the resolution to 0.5 will reduce the number of pixels needed to render by 4 times, and 0.25 will reduce the number by 16 times

(topdown) find close sources of water and add the reflector component only when gameobject is close to the water

Optimizng Obstructions

reduce the size of the obstructions system render texture (resolution slider in reflections section of water settings), as setting the resolution to 0.5 will reduce the number of pixels needed to render by 4 times, and 0.25 will reduce the number by 16 times

find close sources of water and add the obstructor component only when gameobject is close to the water

Optimizng Blur

(box blur) keep the sampling area under 12

(gaussian blur) keep the sampling area under 12

(bokeh blur) keep the quality under 12

in case of top-down games, don't have more than one water pool with blur active at the same time, and deactivate water pools far away from player

note that the quality and sampling area increase the sample rate quadratically ($\sim n^2$ *samples*), so quality of 12 means 144 samples for pixel, whitch is reasonable for computers from last 6-8 years, but impossible for mobile phones. For them you would need to go even lower and can only apply a slight blur.
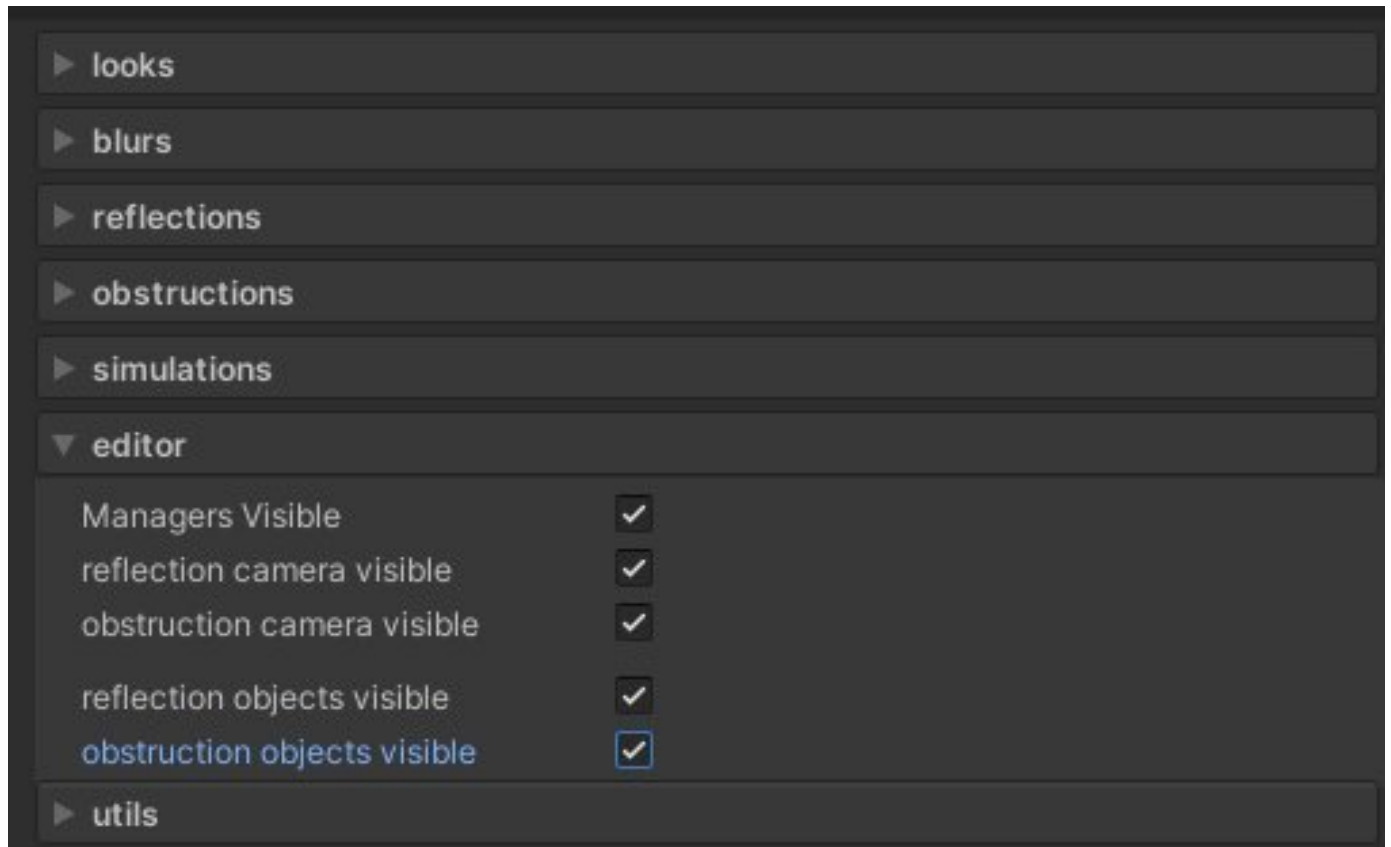
Optimizng Simulation

keep the simulation resolution low, (recommended 2048 , max 4096) for computer games.

in case of top-down games, don't have more than one water pool with simulation active at the same time, and deactivate water pools far away from player

Debugging Setup

! They will show you hidden managers and objects involved in inner workings of 2d water systems, which are normally hidden for clarity and accessibility purposes.

! Also please note that most of the changes made to these components will be overriden by 2d water anyway, so it is recommended to use them as visual reference.