Sky Clouds Manual

Table of contents

Requirements & Setup

Requirements

Unity 2022.3 or higher is required since that is the min version Unity allows for new assets in the store. However it may work just fine in older versions of Unity (though I have not tested it).

Amplify Shader Editor: This asset is based on a shader graph made with the Amplify Editor. The shader works even if Amplify is not installed yet in order to edit it you'll need Amplify.

This asset is for the Universal Render Pipeline (URP) only so you have to have set up URP.

Introduction

This asset is a combination of a some scripts (for cloud intersection fading) and a customized shader for URP (Universal Render Pipeline). You can use both in combination or each separately (though the scripts are tailored to the shader, you may have to adapt them if you want to use them for other shaders).

Quick Guide for Clouds

Basically all you need to do is go to Assets/Kamgam/SkyCloudsURP/Runtime/Prefabs/SkyClouds and pick one of the cloud prefabs.

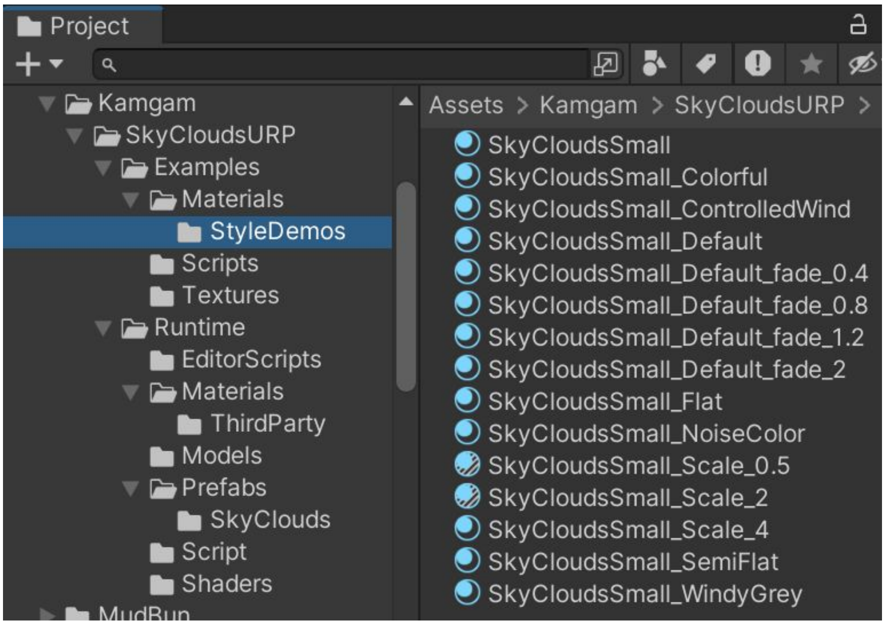
If you drag in the Cloud 01 (Small, Mid, Big) for example you can see they look a bit odd. That's because each of them is made of a certain scale. Small = Scale of 1, Mid = Scale of 2, Big = Scale of 10.

If you scale them up you will see the their material settings actually fit their size. Sadly Unity Assert Store does not allow me to add prefab with a scale other than 1,1,1 so you will have to do the upscaling yourself.

Check out the other prefabs and materials in the Assets/Kamgam/SkyClouds/Runtime folder.

Sky Cloud Material Parameters

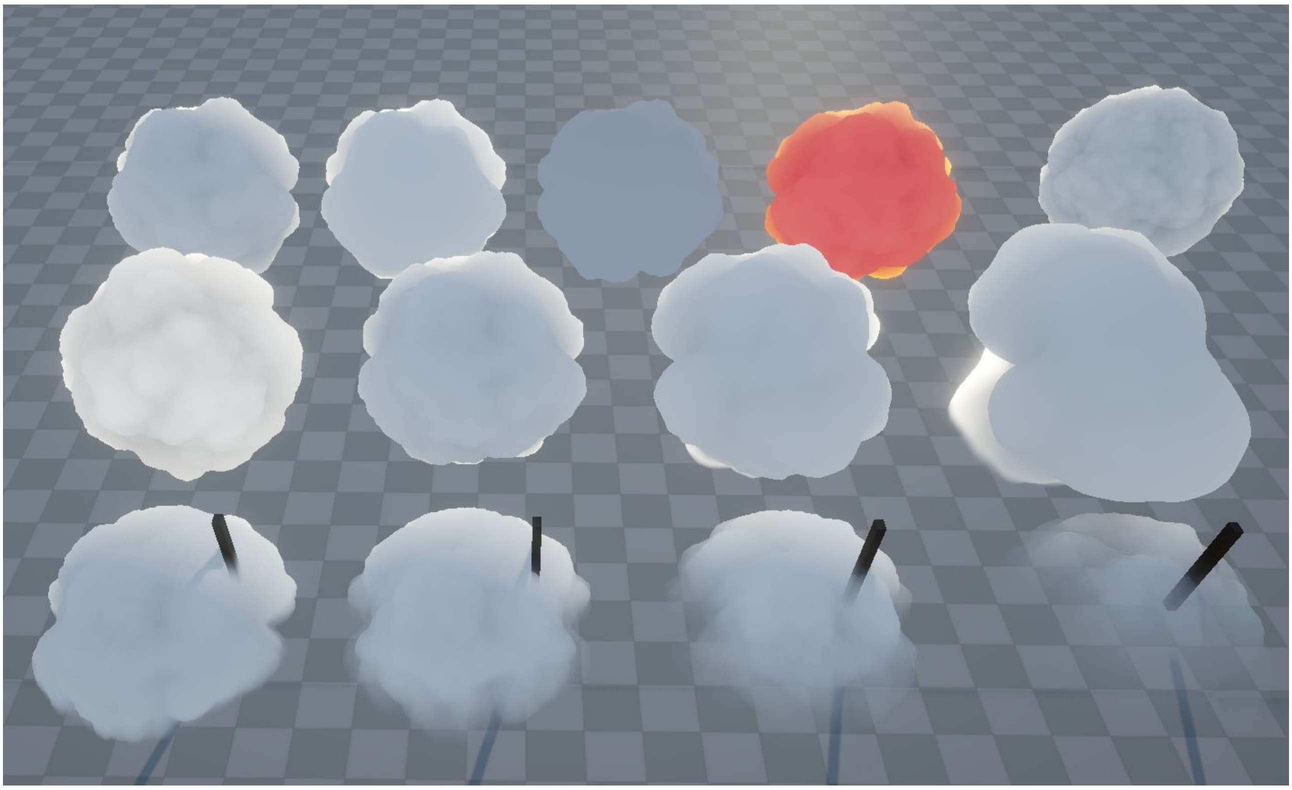
The Sky Cloud shader has a lot of parameters you can tweak. A good starting point is checking out the different „style“ materials in the demo under „Assets/Kamgam/SkyCloudsURP/Examples/Materials/StyleDemos“.



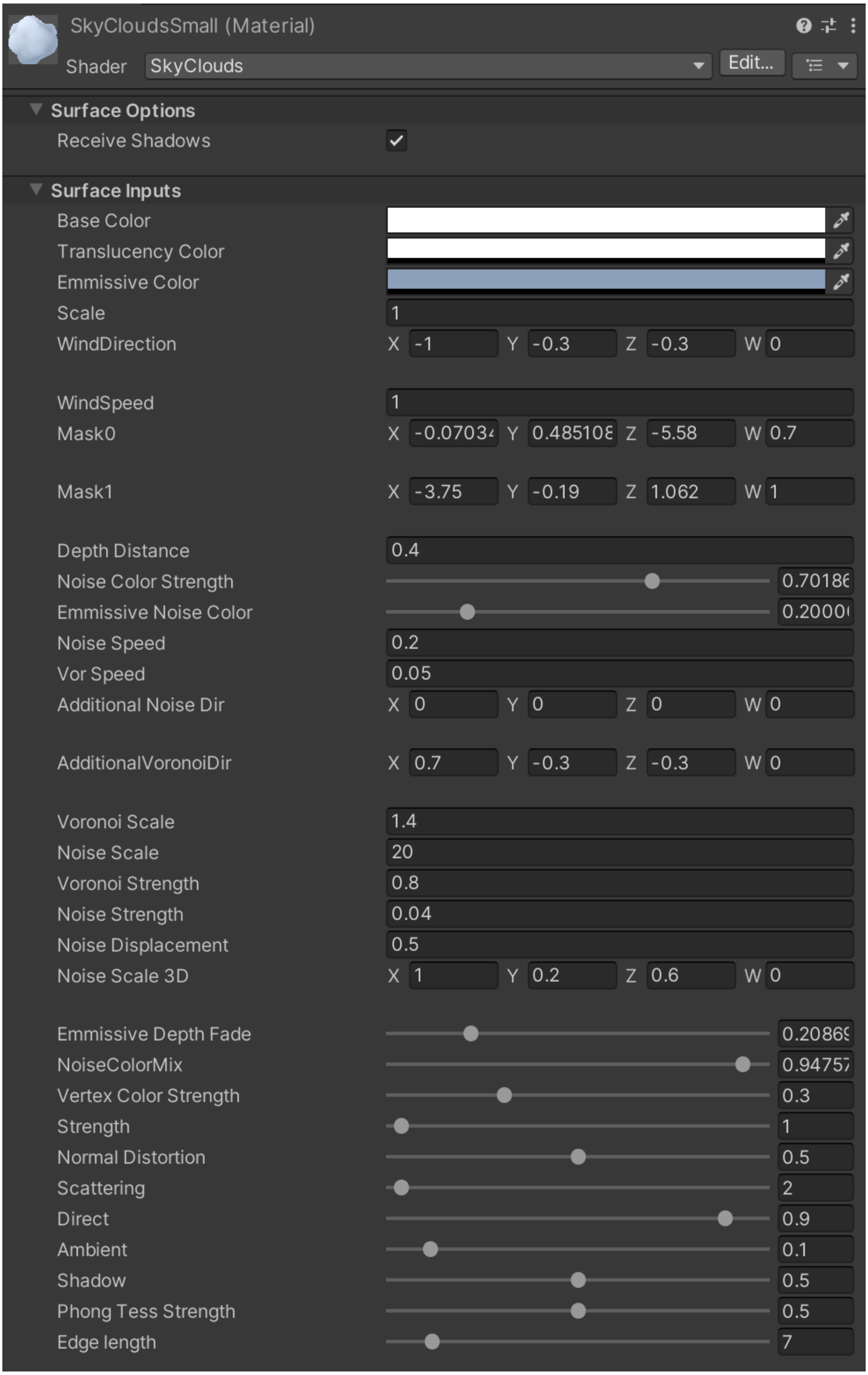
- Kamgam
  - SkyCloudsURP
    - Examples
      - Materials
        - StyleDemos
      - Scripts
      - Textures
    - Runtime
      - EditorScripts
      - Materials
        - ThirdParty
      - Models
      - Prefabs
        - SkyClouds
      - Script
      - Shaders
- MudBun

Assets > Kamgam > SkyCloudsURP >

- SkyCloudsSmall
- SkyCloudsSmall_Colorful
- SkyCloudsSmall_ControlledWind
- SkyCloudsSmall_Default
- SkyCloudsSmall_Default_fade_0.4
- SkyCloudsSmall_Default_fade_0.8
- SkyCloudsSmall_Default_fade_1.2
- SkyCloudsSmall_Default_fade_2
- SkyCloudsSmall_Flat
- SkyCloudsSmall_NoiseColor
- SkyCloudsSmall_Scale_0.5
- SkyCloudsSmall_Scale_2
- SkyCloudsSmall_Scale_4
- SkyCloudsSmall_SemiFlat
- SkyCloudsSmall_WindyGrey



All these materials use the same shader yet differ in material properties.



SkyCloudsSmall (Material)

Shader SkyClouds Edit...

Surface Options

Receive Shadows ☑

Surface Inputs

Base Color	
Translucency Color	
Emmissive Color	
Scale	1
WindDirection	X -1 Y -0.3 Z -0.3 W 0
WindSpeed	1
Mask0	X -0.0703 Y 0.48510 Z -5.58 W 0.7
Mask1	X -3.75 Y -0.19 Z 1.062 W 1
Depth Distance	0.4
Noise Color Strength	0.7018
Emmissive Noise Color	0.2000
Noise Speed	0.2
Vor Speed	0.05
Additional Noise Dir	X 0 Y 0 Z 0 W 0
AdditionalVoronoiDir	X 0.7 Y -0.3 Z -0.3 W 0
Voronoi Scale	1.4
Noise Scale	20
Voronoi Strength	0.8
Noise Strength	0.04
Noise Displacement	0.5
Noise Scale 3D	X 1 Y 0.2 Z 0.6 W 0
Emmissive Depth Fade	0.2086
NoiseColorMix	0.9475
Vertex Color Strength	0.3
Strength	1
Normal Distortion	0.5
Scattering	2
Direct	0.9
Ambient	0.1
Shadow	0.5
Phong Tess Strength	0.5
Edge length	7

Shader Properties

Base Color:

The default ambient color of the cloud, though in reality the emmissive color has more dominance in the shader.

Translucency Color:

Especially important for the edges. Use it to tint your cloud edges according to your sun/sky color.

Emmissive Color:

The main contributor to cloud color. It is used to emulate the color bleeding of clouds when shined upon by the sun.

Scale:

Probably the most useful parameter of all. Use this to scale all the noise parameters below. If you want big clouds then a big scale is your way to got (see the Small/Mid/Big materials).

Wind Direction & Speed:

As it suggests this defines how the noise of the clouds is moving in 3D and how fast it does so.

Mask 0 / 1:

Each mask is a Vector4 which is interpreted by the shader as a position and a radius (these are spherical masks). The first three values (x,y,z) are the position. W is the radius.

HINT: use one for the camera to avoid cloud intersections for as long as possible (the intersection system uses this too.

Depth Distance:

Used for depth testing. Basically this defines how transparent your cloud is if an object is intersecting it. The value is the max range, meaning that at this distance the cloud will be fully opaque.

Noise Color Strength:

The color of the noise is mixed into the ambient color. This defines how much of the noise color (darkening of the lower areas) should be visible.

Emmissive Noise Color:

Similar to „noise color strength“ only in combination with the emmissive color.

Noise Speed:

There are two noises involved in the cloud shape generation. A voronoi noise for the big bulgy shapes and a 3D noise generator for the small ripples you see at close range along the edges of the cloud. This value here controls the speed at which these ripples move.

Vor Speed:

There are two noises involved in the cloud shape generation. A voronoi noise for the big bulgy shapes and a 3D noise generator for the small ripples. This control the speed of the big voronoi shapes underlying the cloud shape.

Additional Noise Dir:

In general the noise directions are controller by the WIND direction. Sometimes you want to have additional noise direction to make a cloud appear changing even if no wind is used.

Additional Voronoi Dir:

In general the noise directions are controller by the WIND direction. Sometimes you want to have additional noise direction to make a cloud appear changing even if no wind is used.

Voronoi Scale:

Controls how big the bluges on the cloud are.

Noise Scale:

Controls how big the small ripples along the cloud edges are.

Noise Strength & Voronoi Strength:

To get the final result the voronoi noise and the 3D noise are mixed together. These values define the ratios of these noises. Usually the voronoi is the dominant part.

Noise Displacement:

Defines how far the vertices of the original shape should be displaced. Very hady for scaling how much of a displacement effect you want.

Noise Scale 3D:

Usually controlle by the wind, though you can set some default values here.

Emissive Depth Fade:

Whether or not the emmission color should also be faded out based on the depth distance check. Have your cloud intersect a plane to see this effect.

Noise Color Mix:

The mix of the two noise colors that give the final noise color result. HINT: Move the slider all the way to the left to see the 3D noise and all the way to the right to see the voronoi noise.

Vertex Color Strength:

Controls how much of the mesh vertex colors should be mixed in.

Strength, Normal Distortion, Scattering, Direct, ...:

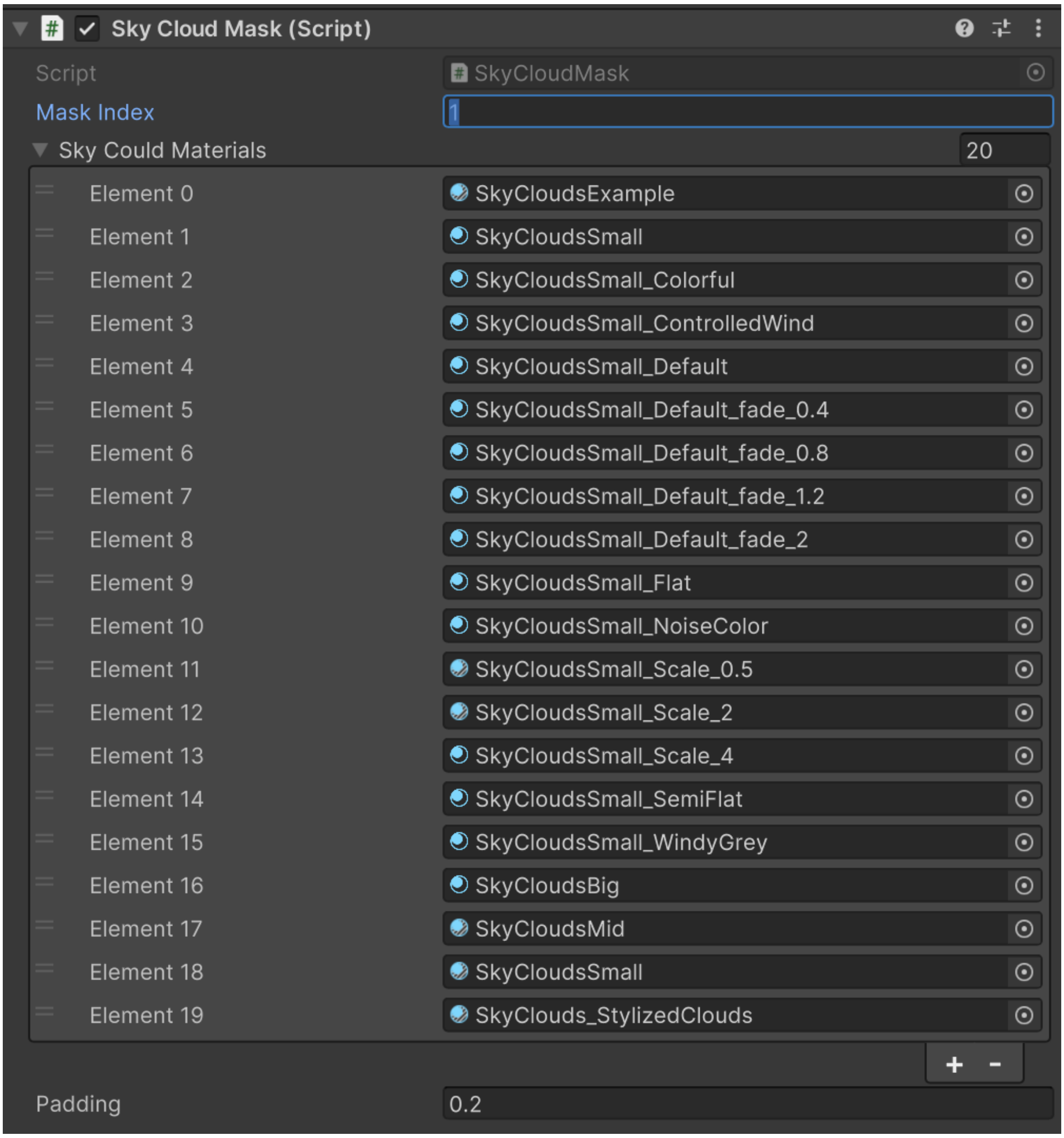
All these values are from the Amplify Shader „Translucency“ and „Tessellation“. Please refer to their documentation for details on these.

A good starting point for tweaking the material yourself is to checking out the different „style“ materials in the demo under „Assets/Kamgam/SkyCloudsURP/Examples/Materials/StyleDemos“.

Masking

The shader supports up to TWO sphere masks. I recommend you use the SkyCöpoudMask script to control the masking.

By default the mask will search the whole project upon creation and add all the SkyClouds material to its list of material. This is not strictly necessary as it also does a dynamic search (if the clouds have a SkyClouds script on them).



Sky Cloud Mask (Script)

Script	SkyCloudMask
Mask Index	1
Sky Could Materials	20
Element 0	SkyCloudsExample
Element 1	SkyCloudsSmall
Element 2	SkyCloudsSmall_Colorful
Element 3	SkyCloudsSmall_ControlledWind
Element 4	SkyCloudsSmall_Default
Element 5	SkyCloudsSmall_Default_fade_0.4
Element 6	SkyCloudsSmall_Default_fade_0.8
Element 7	SkyCloudsSmall_Default_fade_1.2
Element 8	SkyCloudsSmall_Default_fade_2
Element 9	SkyCloudsSmall_Flat
Element 10	SkyCloudsSmall_NoiseColor
Element 11	SkyCloudsSmall_Scale_0.5
Element 12	SkyCloudsSmall_Scale_2
Element 13	SkyCloudsSmall_Scale_4
Element 14	SkyCloudsSmall_SemiFlat
Element 15	SkyCloudsSmall_WindyGrey
Element 16	SkyCloudsBig
Element 17	SkyCloudsMid
Element 18	SkyCloudsSmall
Element 19	SkyClouds_StylizedClouds
Padding	0.2

The padding defines by how much the mask radius will extend beyond the mask object.

The index define what mask this will be controlling (either 0 or 1). I recommend to use 1 since 0 is the default index of the observer Mask.

All the mask is, is a position and a radius. It's entirely up to you how you construct these. The Sky Mask script is only a helper to make it easier.

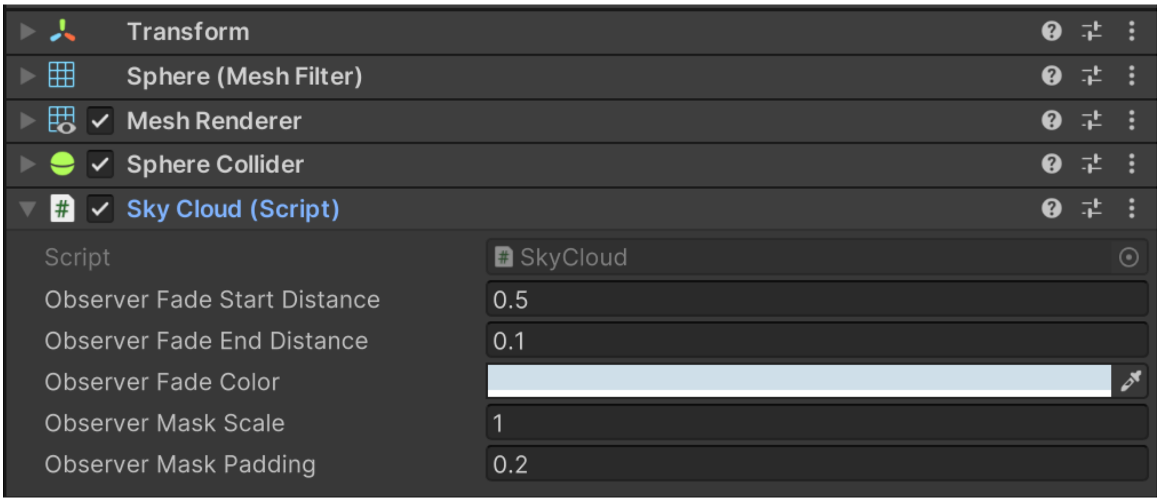
Camera Cloud Intersection Fading

Hint: Check out the Demo. It uses the intersection fade system.

There are some things you will have to set up for the intersection fade system to work.

Prepare the clouds for the intersection fade

Since the intersection fade is controlled by a script and not the shader each cloud that should trigger the fade need a Sky Cloud script attached to it.



Script: SkyCloud

Observer Fade Start Distance: 0.5

Observer Fade End Distance: 0.1

Observer Fade Color

Observer Mask Scale: 1

Observer Mask Padding: 0.2

The script will set some per-cloud data that tells the observer how to behave if it gets close to (or intersects) the cloud.

NOTICE: The distance detection is based on colliders (triggers), therefor you will have to add a collider that matches the surface of the cloud as best as possible.

Observer Fade Start Distance: This is the distance (observer ↔ cloud surface) at which the fade to color will start. As a rule of thumb the default values are okay for clouds of size 1 and should be increased (scaled) the bigger the cloud is.

Observer Fade End Distance: This is the distance (observer ↔ cloud surface) at which the fade to color will end (it will be fully opaque). As a rule of thumb the default values are okay for clouds of size 1 and should be increased (scaled) the bigger the cloud is.

Observer Fade Color: The color of the fade. This should match the overal cloud color. HINT: Make it a little brighter than the cloud, this gives a nice effect.

Observer Mask Scale: If the observer has a mask added (see below) then this will control how much displacement (how big) the mask will be. As a rule of thumb the default values are okay for clouds of size 1 and should be increased (scaled) the bigger the cloud is.

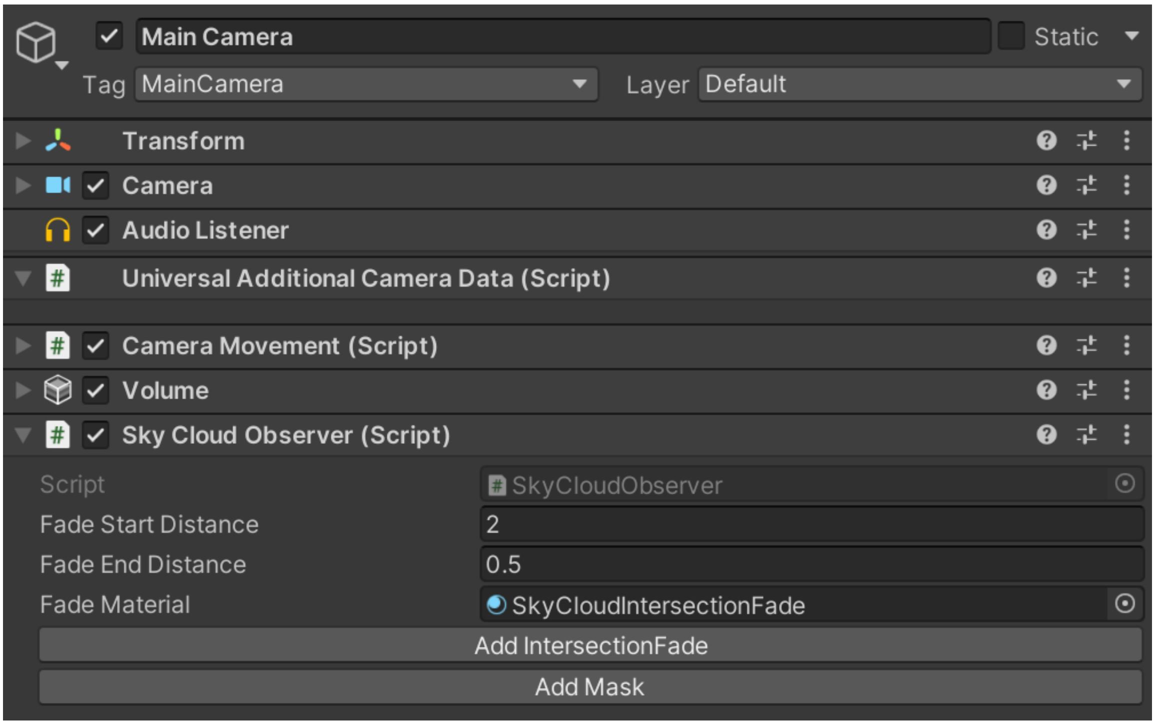
Observer Mask Padding: Same as Mask Scale, just for padding.

Prepare the camera for the intersection fade

If you want your camera to use the intersection fade you will have to add the „Sky Cloud Observer“ script to it. This script will keep inventory of all the clouds and fade a color quad in and out accordingly. It can also add a mask to make the effect even more convincing.

Here is how to do it:

1) Add the „Sky Cloud Observer“ script to your camera game object.



Tag MainCamera

Layer Default

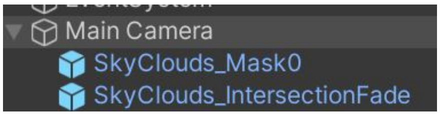
Script SkyCloudObserver

Fade Start Distance 2

Fade End Distance 0.5

Fade Material SkyCloudIntersectionFade

2) Click the „Add Intersection Fade“ and „Add Mask“ buttons. These will add the required prefabs to your camera:



Main Camera
- SkyClouds_Mask0
- SkyClouds_IntersectionFade

HINT: Make sure the Fade Material is set. Usually the script does this automatically but if it fails then please drag it in manually.

And that’s it. Now all that is left to do is configure your clouds (colors and scale) and you are done.

Frequently Asked Questions

Do I need Amplify Shader Editor to make this work?

No, but if you get the Amplify Shader Editor then you can conveniently edit the shader graph. I highly recommend getting it. It's a killer asset and well worth the money.

Can I use this in BiRP or HDRP too?

No, but if you get the Amplify Shader Editor then in theory you could edit the graph and target another render pipline. No guarantees though! This one is tailored too and tested with URP only. I will not give support on BiRP or HDRP ports (sorry). However, if the asset sells well I might do dedicated BiRP or HDRP versions of it.

Are these real volumetric clouds? I saw you flying through a cloud in the demo video.

No, but if you follow the „Camera Cloud Intersection Fading" guide from above then you can get a decent emulation of volume.

If, for some reason, you want the real deal then I suggest you research „ray marching". This approach will give you nice clouds under many circumstances, though it comes at a bit of a performance cost.

Does this work an any hardware? I am targeting low end mobie devices.

Yes, and no. Tessellation requires some base functionality from the graphics hardware in order to work and even then it may not be feasible in terms of performance. That's why there is a shader veriation without tessellation included. The clouds will not look as impressive close up but from a distance people won't really know the difference. You can also work around that by using high res meshes. Most mobile devices are suprisingly good at dealing with lots of vertices.

The cloud looks all weird in Prefab Isolation Mode?

Yes, not idea why though (if you know let me know ;-). It does have no effect on the play mode or builds, just some Unity + Amplify quirk I guess.

My mask is not masking (or only masking some materials). Why?

Have you checked the Mask INDEX. The observer uses index 0 by default so your mask should use index 1. If it uses index 0 the it will fight with the observer on who gets to control mask 0 and thus will not work properly.

Have you checked that the masks materials list contains all your could materials? The mask uses the list to set the mask position and radius of all the materials.

I see some GC allocation for materials and strings in the profiler?

These are all in the Edior only. In builds the system is GC free (except for some setup allocations).

The intersection system does not work in the editor.

It works only at runtime in the GAME VIEW. It indeed does not work in the editor scene view.