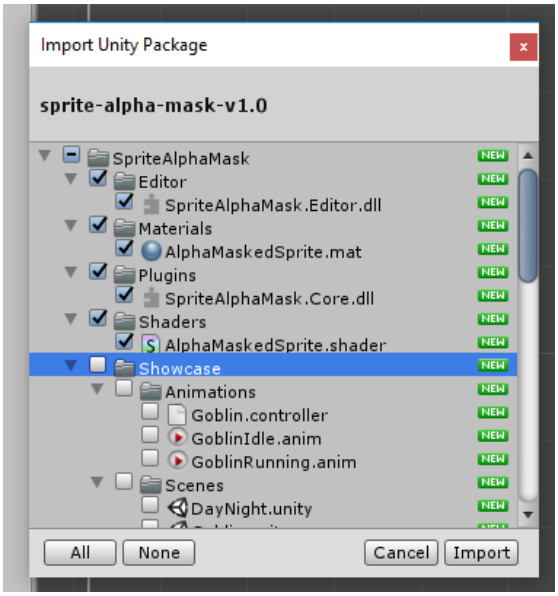
SpriteAlphaMask

Introduction

SpriteAlphaMask offers the possibility of establishing masks that define the partial or total visibility of those Sprites over which they have influence.
A mask is a greyscale Sprite with a special component that lets you define which Sprites will take effect. The area of influence of a mask is specified through a range of Sorting Order values.
Visibility of masked Sprites depends on the mask color values (White, Black or any intermediate value).
It is possible to apply masks that only affect Sprites whose sort order is within a specified range.
In order for a Sprite to be affected by a mask, it must use a Material provided by this package able to modify the level of visibility depending on the location of the mask on the screen.

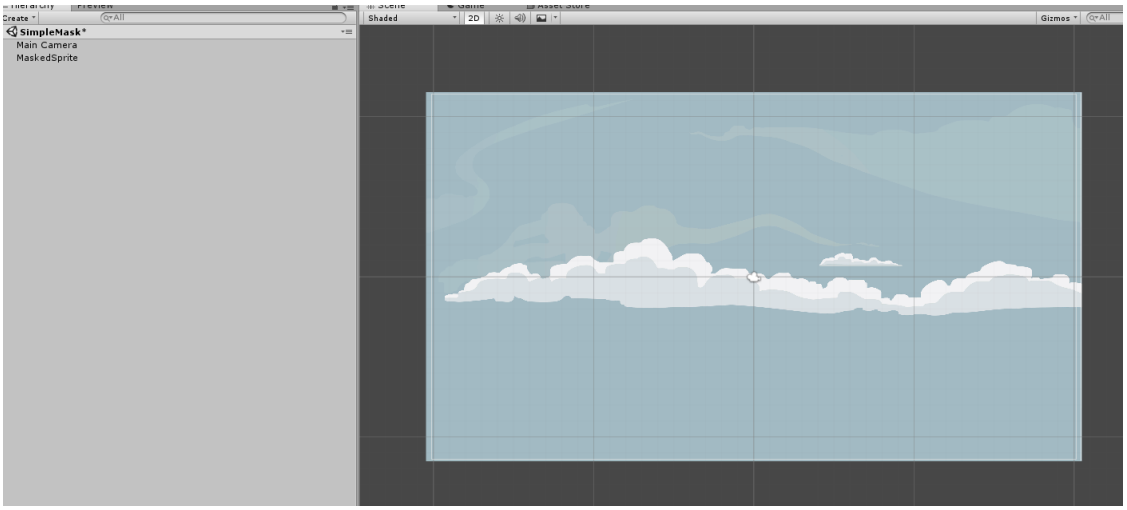
Installation

The package import procedure in Unity is straightforward, the only required folders for using all the features are: Editor, Materials, Plugins and Shaders.
Is recommended to import the Showcase folder if you are willing to follow the current document tutorials.

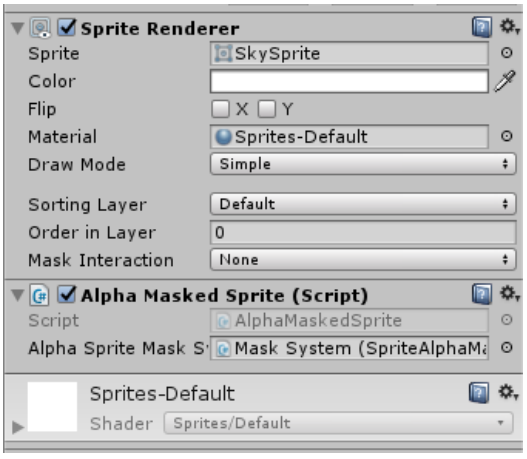


Getting started

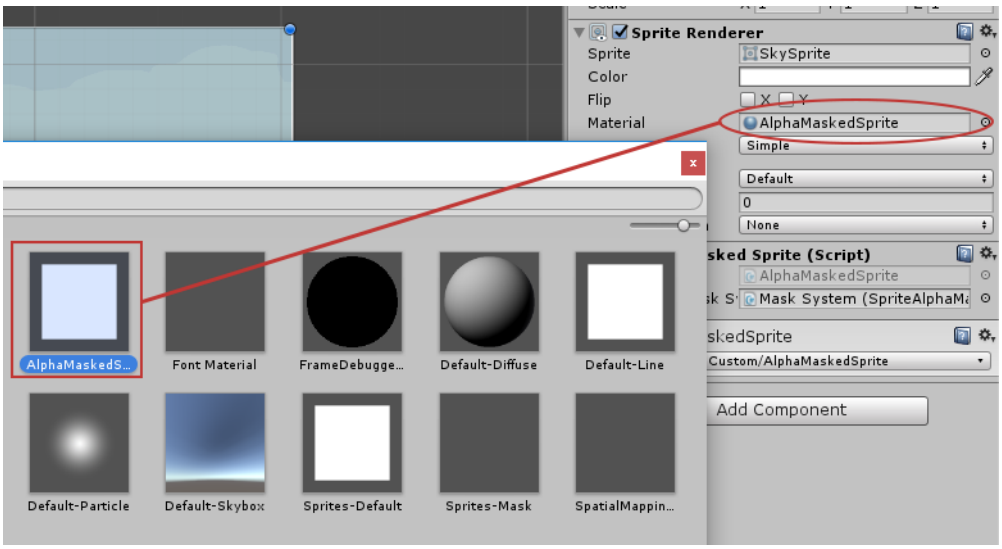
In this tutorial we will create a simple scene where a mask will be applied on a Sprite.
The first thing we will do is define the Sprite on which we will apply the mask. For this we will generate a new Sprite GameObject called *AlphaMaskedSprite*, we will choose as Sprite one of the ones provided by the *SpriteAlphaMask* package: 'SkySprite'.



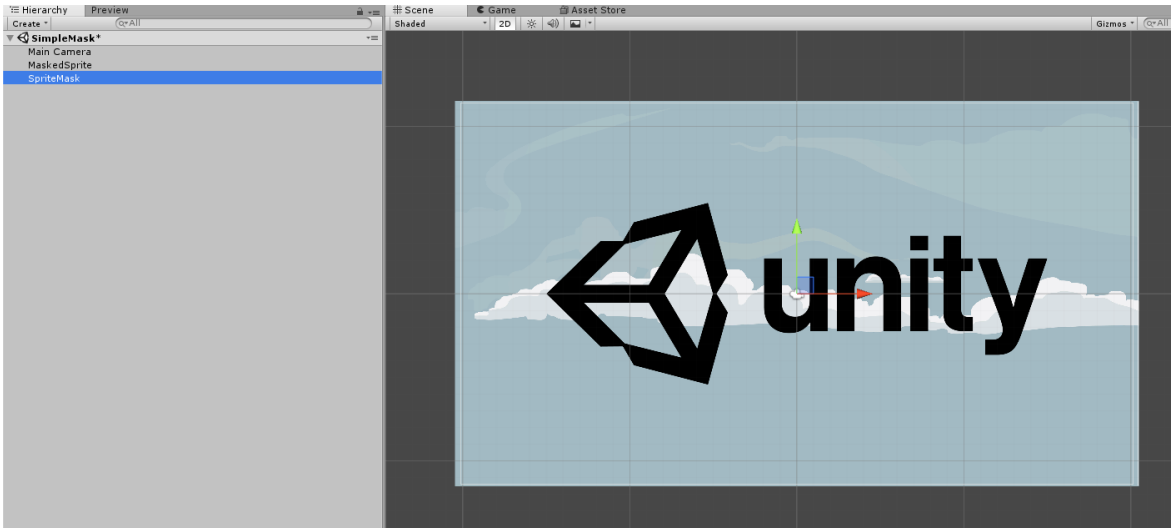
The next step is to add the *AlphaMaskedSprite* component to the recently added GameObject. This component will allow the Sprite to be masked by other Mask Sprites.



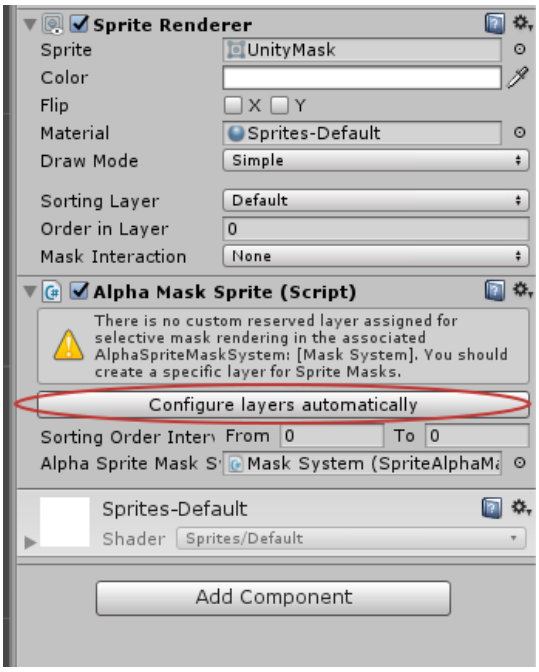
We will also assign a different material to the one that comes by default in the SpriteRenderer component. We can obtain the material within the same SpriteAlphaMask package: *AlphaMaskedSprite*.



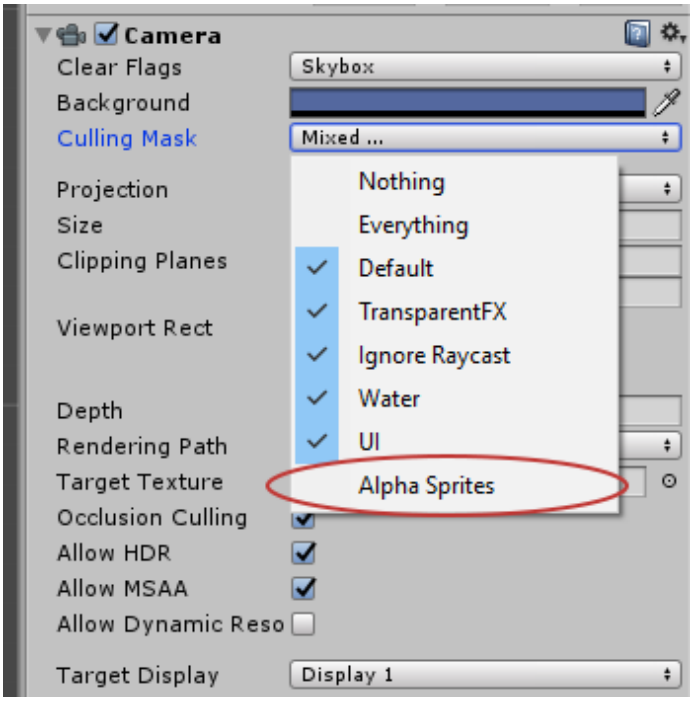
Now it is necessary to define the mask to apply, for this we will create a new Sprite GameObject named 'SpriteMask'. The Sprite to be used as an example will be obtained from the *SpriteAlphaMask* package: 'UnityMask'.



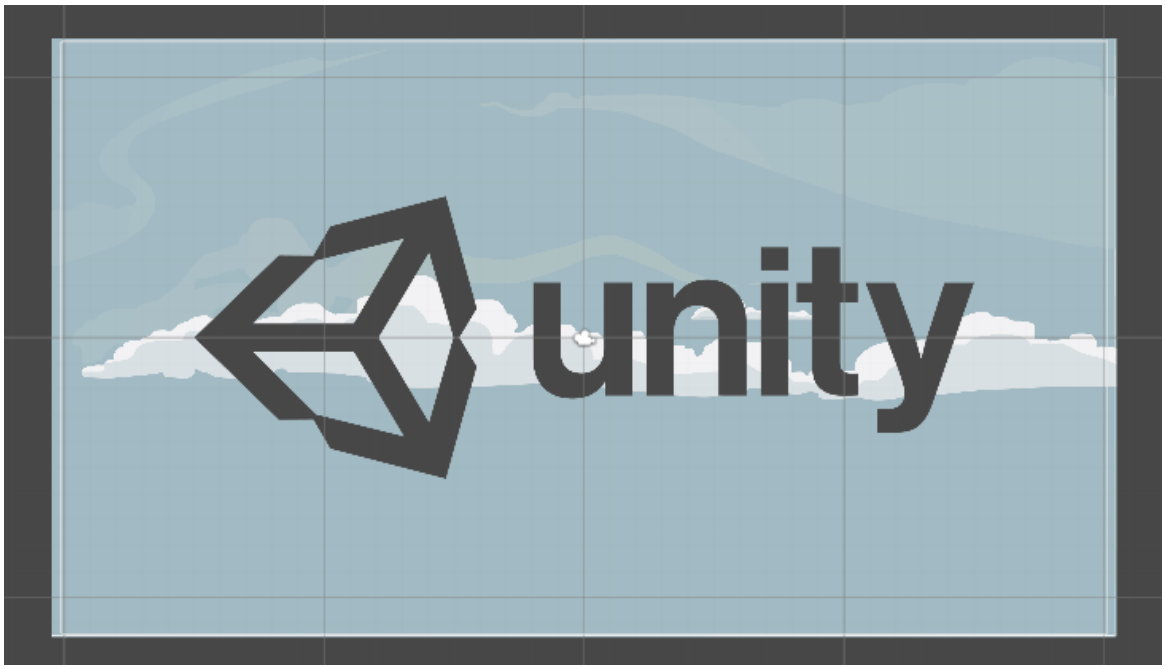
Now we need to add the *AlphaMaskSprite* to the SpriteMask GameObject, this behaviour will allow the Sprite to act as a Mask of other Sprites.
The inspector will warn us that it is necessary to configure a custom reserved layer for the correct operation. At the moment we can correct the problem automatically by clicking on 'Configure layers automatically'.



Before finalizing, we will take care of hiding the layer created automatically in the previous step in our camera.



Done!, the mask is now being applied on the background and it is possible to see the result directly in the editor without having to run our game.



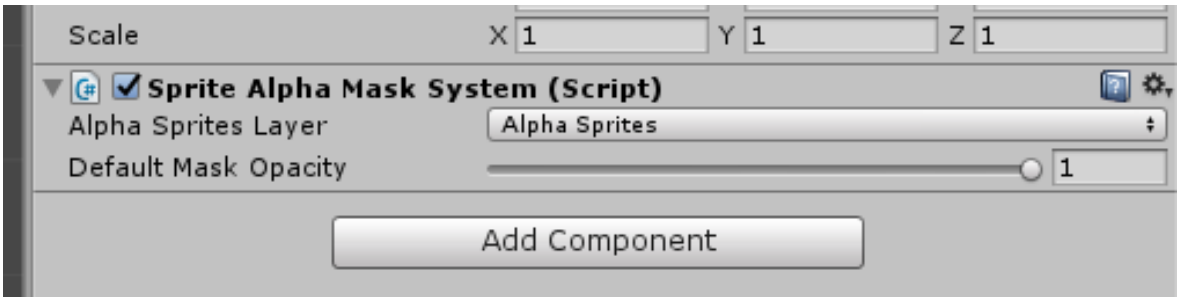
In this tutorial we saw how to generate a mask from one Sprite and apply it to another Sprite. During the process, some things happened automatically to avoid complications.

There are still things to discover, its recommended to keep reading to get the full potential of this tool.

Advanced features

The *SpriteAlphaMaskSystem*

SpriteAlphaMaskSystem is the component responsible for managing, optimizing and rendering the existing masks in the scene.
In most cases, it is only necessary that this component is defined only once in the scene. It is not necessary to do it manually since a new GameObject with this component is generated when an AlphaMaskSprite or AlphaMaskedSprite is created for the first time.



SpriteAlphaMaskSystem configuration

Alpha Sprites Layer: This layer is used internally by the system to make selective rendering of masks. The user should specify an unused layer reserved for this system. In most situations this layer is generated automatically and assigned to the system.

Default Mask Opacity: Defines the opacity value of the area in the screen where there are no masks defined. Its useful for specifying inverted masks.

Inverted masks

There are lots of situations where its useful to have a Sprite hidden by default and a mask that only defines visible areas of the Sprite.

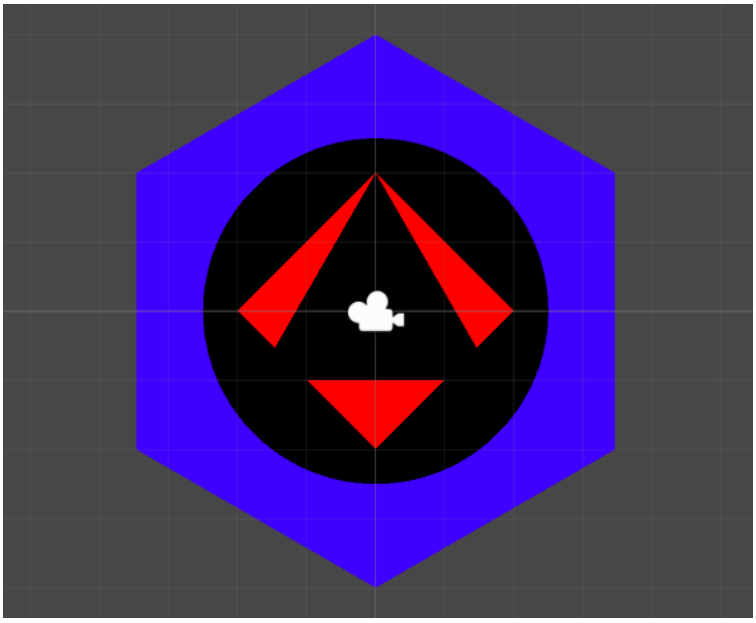
This can be achieved setting the *Default Mask Opacity of the SpriteAlphaMaskSystem to zero.*

The scene named DayNight in the package showcase is an example of this situation and how to handle it.

Multiple sprite masking and sorting order intervals

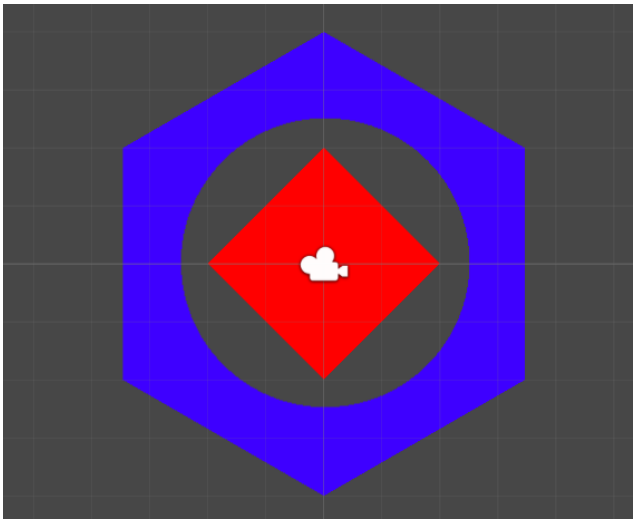
One of the main features of SpirteAlphaMask is the possibility to apply a single Sprite mask to multiple Sprites. But if you need to apply the mask only to a subset of the Sprites you will need to configure the sorting order interval of the mask to define the area of effect of the mask.
The

Lets create a simple Scene with 4 shaped Sprites (Blue Hexagon, Black Circle, Red Diamond and Black Triangle) and assign each Sprite a different sorting order: 0 Hexagon, 1 Circle, 2 Diamond, 3 Triangle. You can modify sorting order and color from the *SpriteRenderer* component in each Sprite. The scene should look like this:



Now lets assign *AlphaMaskedSprite* Component and *AlphaMaskedSprite* material *to the Blue and Red Sprites (Hexagon and Diamond).*
Then assign *AlphaMaskSprite* Component *to the Black Sprites (Circle and Triangle).*
Apply automatic layer configuration to each *AlphaMaskSprite.*

The scene should look like this:

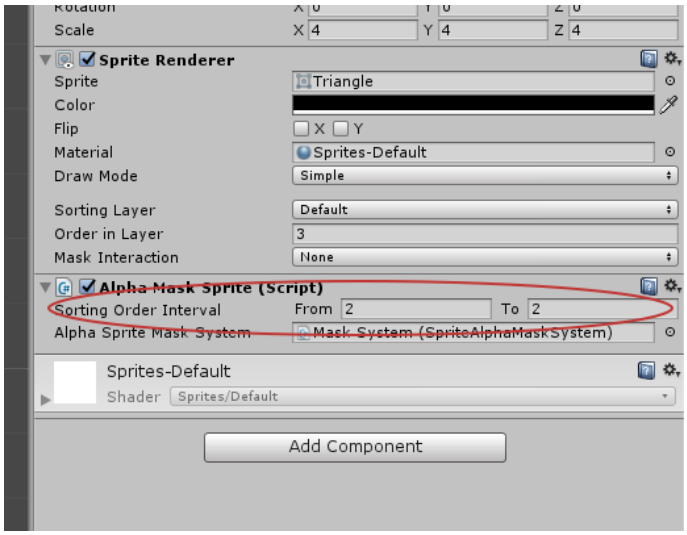


Where did the Triangle mask go?. Why the Diamond Sprite is not being masked?. These are the questions we will answer with this tutorial.

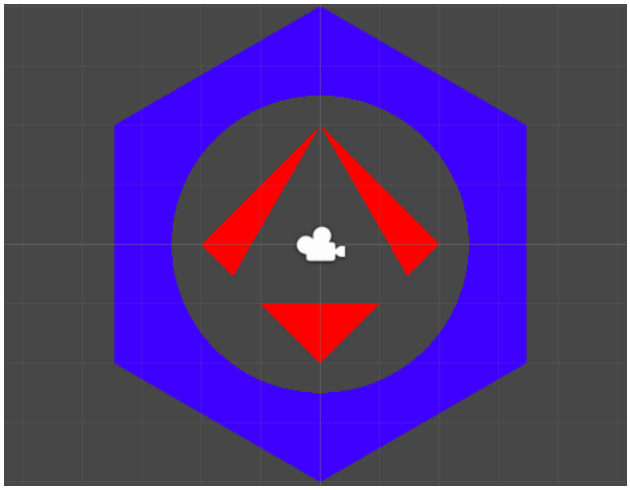
At the moment, the only Sprite affected by the masks is the Diamond Sprite (sorting order 0). The Triangle mask is there, but the area of effect of the Circle mask is bigger, so we can’t see any difference with the Triangle mask.

Lets make some changes so we can use the Triangle mask to affect only the Diamond Sprite.

Select the Triangle mask from the scene hierarchy and change the sorting order interval in the inspector to [2, 2].



Now the Triangle mask will only affect those Sprites inside the [2,2] sorting order interval (our Diamond Sprite).
By default the affected interval is [0,0] so the Cirlce mask is only affecting the Hexagon Sprite.



More information, contact and support

If you need support, you can contact me by the following means:

E-mail: gcaseres@gmail.com

Unity Forums:

Username: gcaseres
Thread: https://forum.unity.com/threads/coming-soon-sprite-alpha-mask.520933/

Youtube:

Channel: https://www.youtube.com/channel/UCX4K9orpgxxJSXFnWvJx8eg

World wide web:

url: http://gcaseres.gitlab.io/unity