

Azure[Sky]

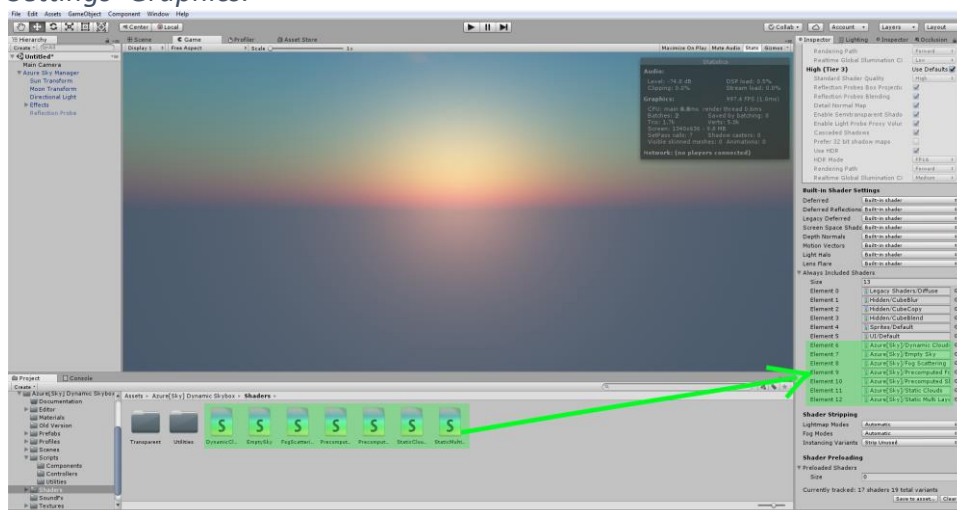
Dynamic Skybox

Document Version 5.0.1

English is not my native language, so I apologize for possible grammar errors.

First Steps

- It is always a good practice to save a copy of your project or creating a new project for testing before importing any package.
- The Anti Aliasing somehow seems to affect(blur) the depth buffer information and consequently generating artifacts when using the fog scattering effect. The fog scattering effect is depth-dependent, so it is recommended to disable anti-aliasing in the project settings by the menu: *Edit>Project Settings>Quality*.
- Azure[Sky] works in both color spaces but is advice switching the project to linear color space, since all sample scenes have been set up in linear color space. *The rain color may be very white in the gamma color space, you can easily fix this by adjusting the rain color to a darker color in each day profile.*
- Unity includes in the build only the shaders that are currently in use by the materials at the time when the project is built. If you build the project with any sky setting and while in gameplay switches to a scene that uses a different sky setting, there will be an error in the sky material rendering(pink shader error) because Unity did not include in the build the shader variation required for render the sky with the new configuration. To avoid future errors after building the project, add all variations of the sky shaders in the "**Always Included Shaders**" list located in: *Edit>Project Settings>Graphics*.



Getting Started

To start using Azure[Sky], all you have to do is drag the [Azure Sky Manager](#) prefab into your scene, the skybox material should be automatically replaced by the Azure sky material.

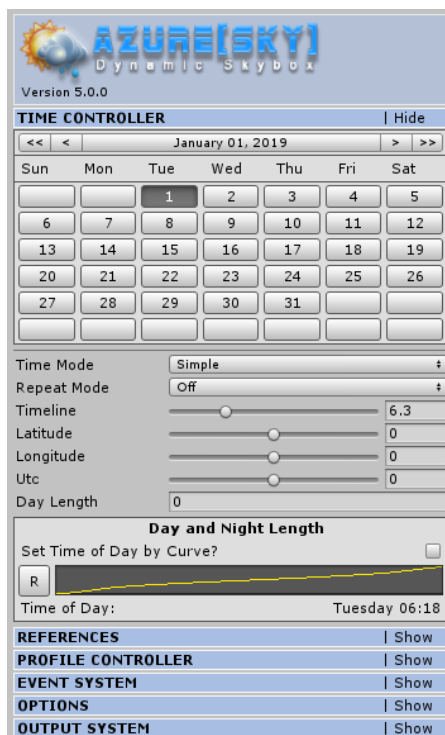
Also(*Optional*), add the fog scattering effect to the camera by dragging the [AzureSkyFogScattering.cs](#) script to the Inspector or by the menu:

Component>Azure[Sky]>Fog Scattering.

- First, remove any directional light from the scene because the prefab already comes with a directional light attached to it.
- The fog scattering distance is set by default to a value greater than 5km, you must adjust the fog distance (*at the “Fog Scattering” tab of each day profile*) to a value that best suits your project.
- The prefab is set by default as follows:
 - Simple Time Mode;
 - Stylized Sky Model;
 - Dynamic Clouds;
 - Reflection Probes Off;
 - Weather Zones Off;
 - Event System Off;

Important: The profiles were customized based on the timeline mode.

When you select the sky manager prefab in the scene, the Inspector should look like this:



In the **Time Controller** tab are all the options for setting time, date, and location.

- You can navigate through the calendar buttons to change the date.
- The date you set in the Inspector will be the date the sky system will start when you play the game.
- Clicking on the middle button displaying the current date will show a hidden menu useful to quickly jump to a custom date.
- Changing the time mode to "Realistic" will accurately set the position of the sun and moon in the sky based on the time, date and location(*latitude and longitude*).
- Day Length is the day cycle duration in "minutes". Set to zero if you want the time to be static.
- The "Time of Day" below the "Day and Night Length" curve, displays the timeline(Float) converted to "hours and minutes"(Vector2). This is the time taken into account by the Event System.

In the **References** tab are attached all materials, transforms, textures and other resources used by the sky manager. There is not much to do on this tab, but if you want to change for example the textures of the sun or moon, this is the place.

The **Profile Controller** tab is responsible for all things regarding the use of the day profiles such as global weather, local weather zones, and default day settings.

- **Local Zone Trigger:** The Transform used to perform the local weather zones blends. Set to null if you want to disable the weather zones feature to save performance.
- **Global Weather Profiles:** A list you can add or remove as many day profiles as you want, you should use it to perform the global weather. Next to each profile in the list has a field to set the transition duration time for that profile and a button for quickly test in editor.
Note: The global weathers are overrides by the local weather zones.
- **Default Day Profiles:** A list where you can add day profiles to work as a regular calendar day (the default weather for that particular day).
 - * A random profile from this list will be set to work as default day profile every time a new day starts.
 - * The game will start always using the first profile from this list.
 - * If you want some date to start with a specific day profile, you can setup an Event Action to replace the current profile in use by the new profile just calling the:
`public void SetNewDayProfile(AzureSkyProfile profile).`
- **Local Weather Zones:** After creating the weather zones in the scene, you must add them to this list and arrange them according to the priority. For performance and organization reasons, the sky manager will process only the weather zones contained in the list.

In the **Event System** tab, you can add, move or remove as many "Event Actions" as you want to control all sorts of things in your game. You can create an "Event Action" to explode a volcano at a certain date and time. You can use the event system to create or delete game objects, call methods in other scripts, etc...

- **Scan Mode:** As the Event Actions are performed according to the date and time, there is no need to waste performance by checking each event every frame, instead, this is done only when the time of day changes. So the scan mode sets when the sky manager will scan the event list to check if it needs to invoke some Event Action.
- **Events:** The list with all event actions.
- You can force the event system to ignore dates or times by setting the value -1 to the field you want the scanner to disregard when checking the event list. You can set a real value only for the time fields and set -1 to date fields, this will cause the event to be called every day at that time.

In the **Options** tab, you define the sky model used to render the sky, change the clouds mode, enable the reflection probe and set the position of the starfield.

In the **Output System** tab, you can create extra properties to be set in each profile and extends the sky system functionalities. You can create an extra property to control for example the snow cover of your favorite terrain shader, this way you can set a different coverage value in each day profile. The sky manager will perform all blends and weather transitions to this property and store the output result in a list where you can access and use for control all sorting of things.

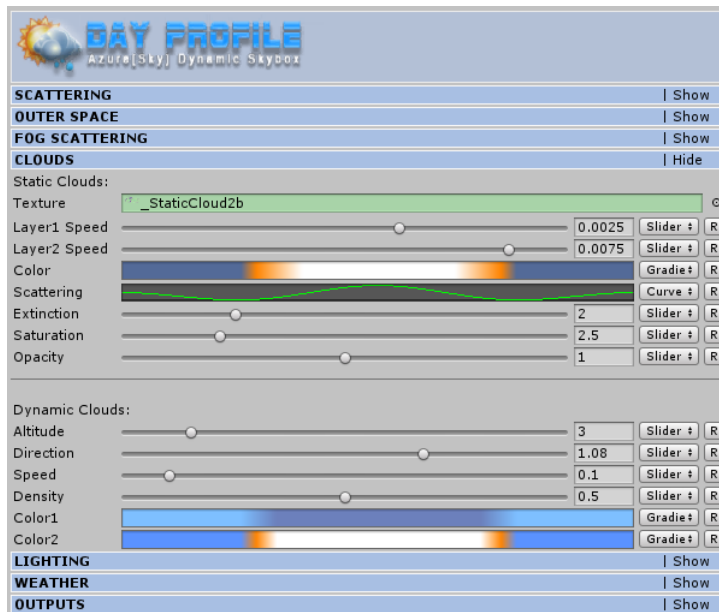
Creating a Day Profile

To create a profile is very simple, just click on the menu: *Create>Azure[Sky] Dynamic Skybox>New Day Profile*. This way will create a profile without the proper configuration with all colors and gradients set to white. The best way is to select the _Default profile and click on the keyboard shortcut "Ctrl + D" to duplicate the existing profile and starting the new customization from there.

- To view your profile changes in real time while you customize it, the best way is to temporarily add the new profile to the first element of the *Default Day Profiles* list in the sky manager's Profile Controller tab.
- You can create as many profiles as you want.
- The profiles can be shared with the team members.

Customizing Your Profile

The customization of the profiles is very simple and has no secrets, you just need to move some sliders until you find the result that you consider ideal.

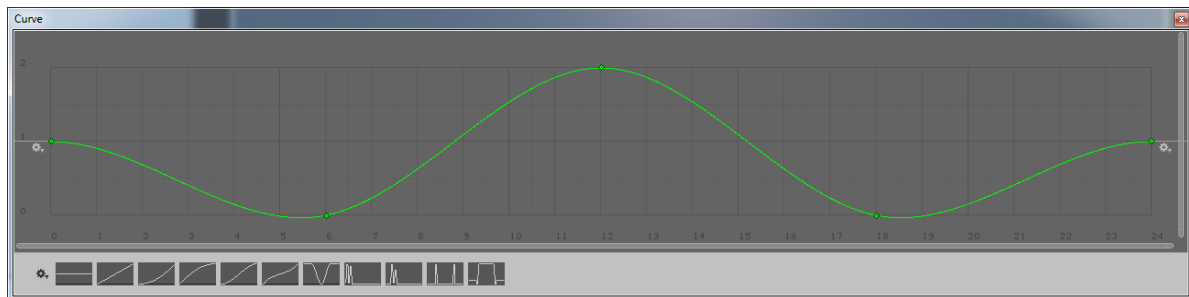


- Next to each property there is a popup to select the property type. In most cases, you will need to use only a slider, but on some occasions, you will want the property to have different values depending on the time of day, so you can change the property type to curve field and set different values for each time of day.
- The same for the color fields, if you want the color to change according to the time of day, simply change the property type to gradient field.
- There is also a button to reset the property.

How Do The Curves Work?

Curves allow each property to have dynamic values according to the time of day.

Timeline Curves: The customization of this curve type is based on the timeline. When the time of day changes, the property value also dynamically changes to the value you set on the curve for that time of day. See in the screenshot below how the value of the property varies between 0 and 2 along the 24-hour day cycle.



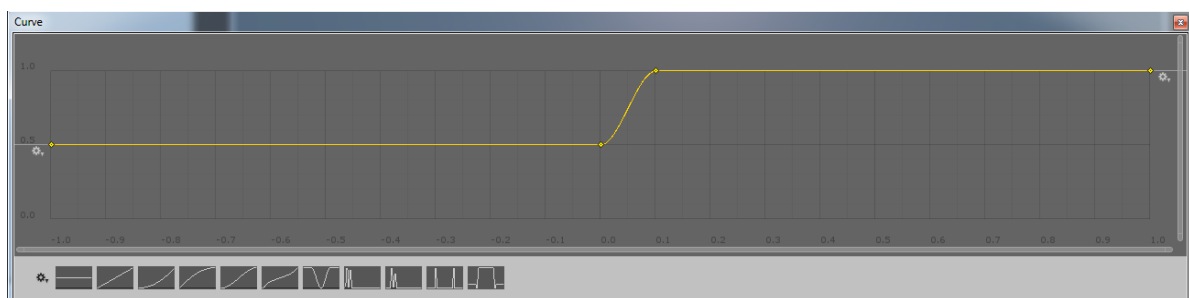
Sun Elevation Curves: The customization of this curve type is based on the sun elevation in the sky. Let's suppose you customize a property using the curve mode based on the timeline and set, for example, the intensity of the directional light near zero at 20 o'clock taking into account that it is already night. If you are using realistic time mode, depending on the location and date, in some places in the world it can still be daytime at 8pm making the curve customization out of sync. In this case the curve should be customized based on the elevation of the sun in the sky, if the sun is above the horizon line, then it is daytime, otherwise, it is night. This curve type allows you to customize dynamic values depending on how high the sun is in the sky.

When the curve time is -1, it means that the sun is below the horizon line at 270°, in other words, it must be midnight...

When the curve time is 0, it means that the sun is at the horizon line at 0°, it must be sunset or sunrise.

When the curve time is 1, it means that the sun is above the horizon line at 90°, in other words, it must be noon.

See in the screenshot below how the curve is set to 1.0 at daytime and is set to 0.5 at night based on the sun elevation.

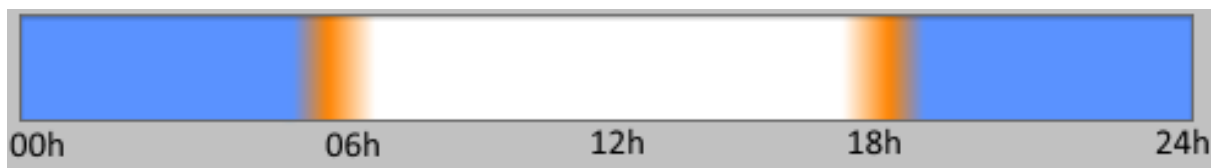


Moon Elevation Curves: The same as the sun elevation curves, but in this case it is based on the moon elevation in the sky.

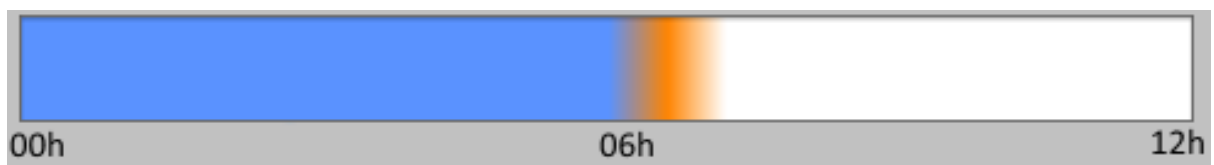
How Do The Gradients Work?

Gradients allow each color property to have dynamic values according to the time of day. The gradients work in the same way as the curves, but different from the curves it is not possible to visualize the time, but there is a trick that can help when customizing a gradient.

When the gradient is based on the timeline, the left side is referent to the 00h, the center is referent to the 12h and the right side is referent to the 24h. You can use the image below as a time reference.



When the gradient is based on the sun/moon elevation, the left side is referent to the 00h, the center is referent to the 06h and the right side is referent to the 12h. You can use the image below as a time reference.

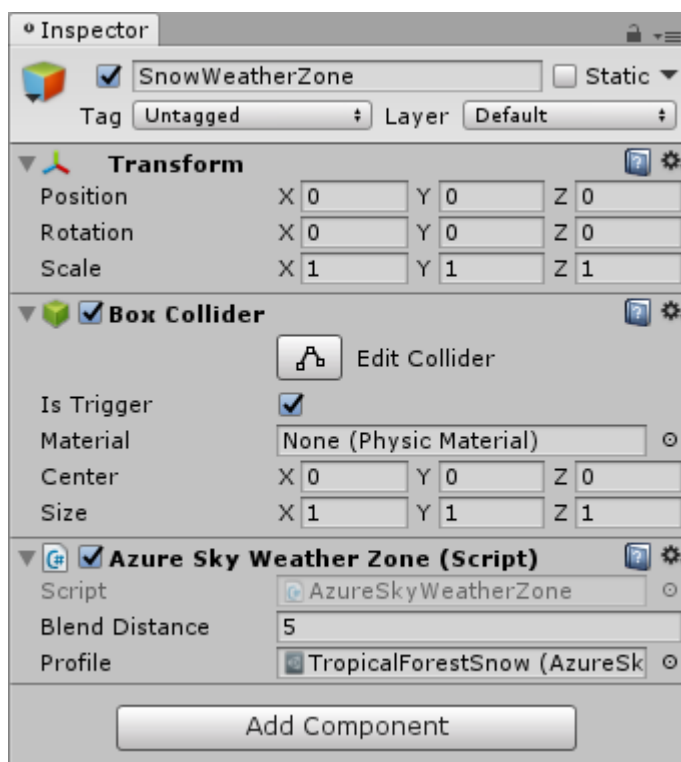


Weather Zones

Create a weather zone is very simple:

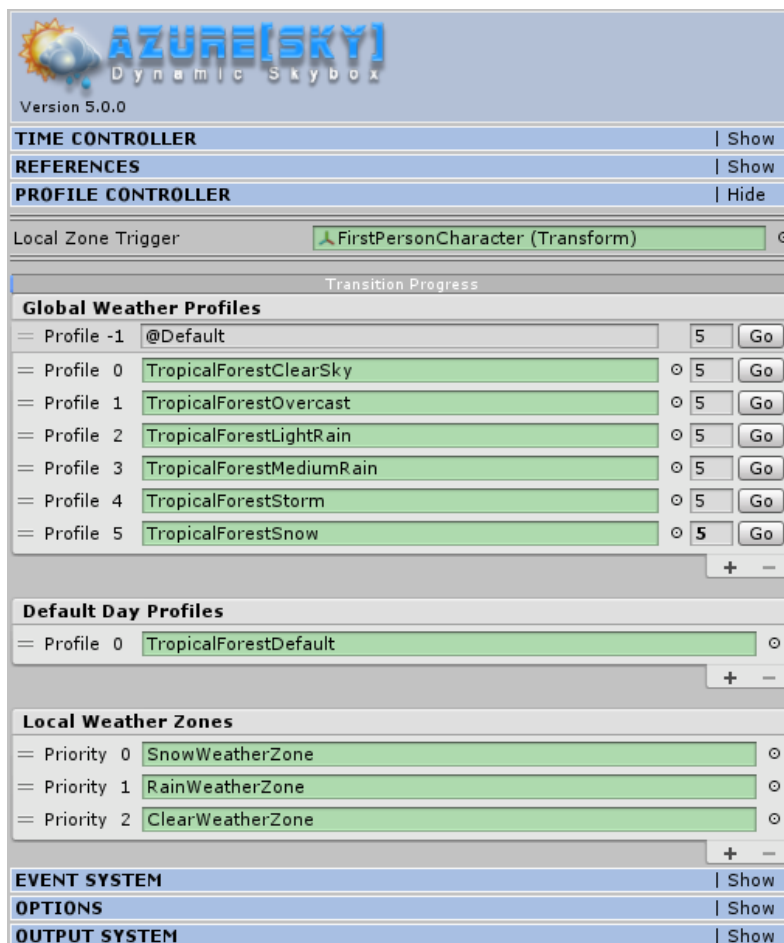
- Create an empty game object.
- Add a collider to it and check the “Is Trigger” option. It supports box collider, sphere collider, and mesh collider. Convex mesh is not supported.
- Add the weather zone component to it by dragging the **AzureSkyWeatherZone.cs** to the Inspector or by the menu: *Component>Azure[Sky]>Weather Zone*.

The Inspector of the game object should look like this.



Each weather zone component contains a field to place a day profile with the sky settings for that zone and also the option to set the blend distance.

You can now set the scale and location of the weather zone in your scene. It is also necessary to add this new weather zone in the "**Local Weather Zones**" list on the sky manager's "**Profile Controller**" tab and arrange according to the priority.



For performance, all the weather zones in the scene that is not attached in the "Local Weather Zones" list will be ignored by the controller.

Add the trigger transform to the "**Local Zone Trigger**", usually the camera or player is used as a trigger. Setting the "local Zone Trigger" to null will disable the local weather zones (the global weather will still work).

Now the weather zone system should already be working, when moving the trigger into the zone collider, the sky settings should automatically change to the setting defined in that zone.

Output System

The output system is a very useful feature that extends the sky system functionalities. For example: Do you need a property to control the wetness of your favorite terrain shader? No problem just create a slider output and set the appropriate value in each profile, eg: In the rain profile, you set the value of this output to the maximum, in the sunny day profile you reduce the output value to zero. The sky manager will perform all blends and weather transitions to this output and store the result in a list where you can access with a simple script and send it to your terrain shader.

In the **Output System** tab of the sky manager, you can create as many outputs as you need, set the type and write a description of its functionality to help you remember why you created it.

In the screenshot below I have created a slider output to control the terrain wetness.



The output is not available for customization in the profiles until the "**Apply Changes to Profiles**" button is pressed. By pressing the button, the sky manager checks each profile in the project and creates the output only in the profiles with the same output tag.

As I developed this feature, I figured out that it is not a good idea to make changes to all profiles from the project, because depending on the case it is possible that the user will use the Sky Manager in more than one scene and in this way the Sky Manager will overwrite all the outputs created by the other scene and broken the system. For this reason, it was necessary to create an output tag system to separate the outputs into groups, so the Sky Manager will create and modify only the outputs from the profiles in the project that contain the same tag defined in the Output System tab of the Sky Manager.

Now that the output has been created, you can select the profiles and customize the new property that was created in the Outputs tab of each profile.



The sky manager will perform all blends and weather transitions to this output and store the result in the `public List<AzureOutput> outputSystemList;` where you can access with a simple script and send it to your terrain shader.

Example script:

```
using UnityEngine;
using UnityEngine.AzureSky;

public class GetOutputExample : MonoBehaviour
{
    public AzureSkyManager azureSky;
    private float m_terrainWetness = 0.0f;

    // Others outputs types examples
    //private Color m_myColorOutputExample = Color.white;
    //private Vector2 m_myVector2OutputExample = Vector2.one;
    //private Vector3 m_myVector3OutputExample = Vector3.one;

    void Update ()
    {
        m_terrainWetness = azureSky.outputSystemList[0].floatOutput;

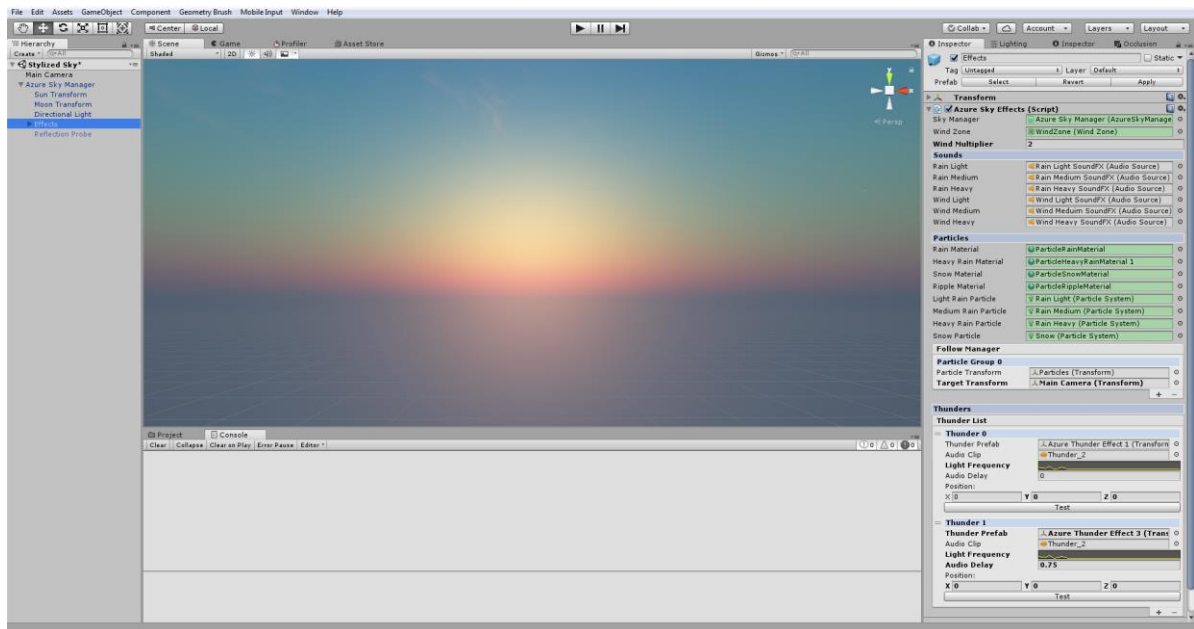
        // Others outputs types examples
        //m_myColorOutputExample = azureSky.outputSystemList[0].colorOutput;
        //m_myVector2OutputExample = azureSky.outputSystemList[0].vector2Output;
        //m_myVector3OutputExample = azureSky.outputSystemList[0].vector3Output;
    }
}
```

Very Important:

- If you create an output in the sky manager and do not save the scene, the output created will continue to exist in the profiles, but no more in the sky manager. Remember to save the scene every time you create an output otherwise you will have to recreate the outputs when you open the scene again.
- If you press the “Apply Changes to Profiles” and there is no output in the sky manager output list, all the outputs will be deleted from the profiles using the same output tag.
- If the outputs from the sky manager do not match with the outputs from the profiles, the output blend transitions will be aborted.

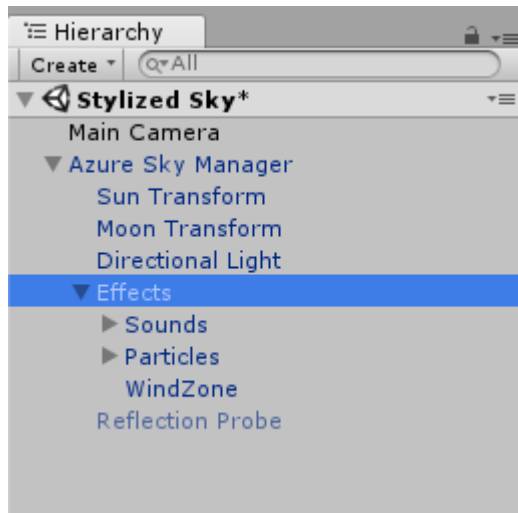
New Particle System

The particle system has been completely redone and improved. The particles, thunders, and sound effect systems are now completely independent and work in a separate GameObject inside the sky manager prefab. Now it is easier to manage the effects and if you want, you can even delete the effects system without breaking the sky system.

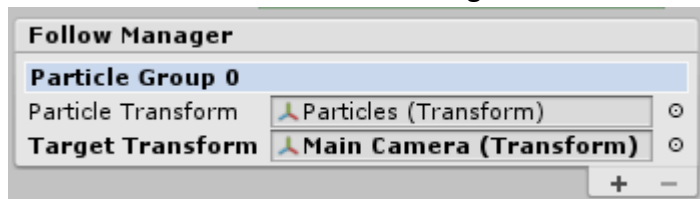


Making the particles follow the camera

- Select the “Effects” game object that is inside the sky manager prefab.



- Add in the Inspector a new particle group to the “Follow Manager” list.
- Attach the “Particles” game object to the “Particle Transform”.
- Attach the MainCamera to the “Target Transform”.



- If you need the particles in different locations, you can duplicate the "Particles" game object and add it to a new group in the Follow Manager list, set another target/position that it should follow.