Celestial Cycles: Dynamic Day-Night System

March 07, 2024

Product Overview

Welcome to "Celestial Cycles: Dynamic Day-Night System," a state-of-the-art Unity editor extension designed to breathe life into your virtual worlds through a dynamic simulation of natural lighting and atmospheric conditions. This manual will guide you through each component and feature, enabling you to harness the full potential of day-night cycles in your Unity projects..

How to Setup

Step 1: Open Your Unity Project

Start Unity: Open your Unity project where you plan to add the dynamic day-night cycle.

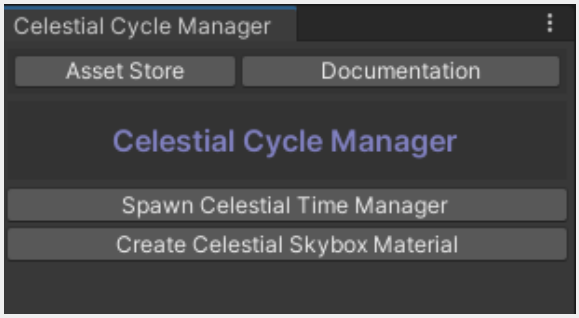
Step 2: Accessing Celestial Cycle Manager

Find the Tool: Once the "Celestial Cycle Manager" script is added to your project, it integrates into Unity's menu bar. Navigate to Windows> Celestial Cycles > Celestial Cycle Manager.

Step 3: Spawning the Celestial Time Manager

Spawn Celestial Time Manager

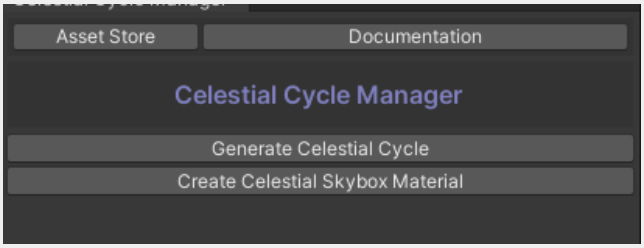
Add the manager to your scene. This action creates a new GameObject with the CelestialTimeManager component attached. This component acts as the core controller for the day-night cycle in your scene.



Step 4: Setting Up Celestial Cycle Periods

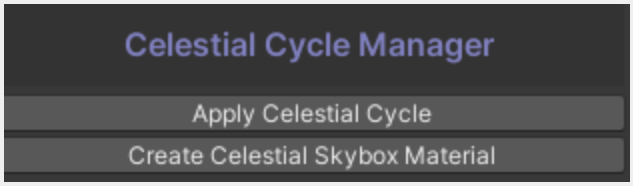
Select Generate Celestial

This step creates profile for predefined time periods (e.g., morning, noon, evening, night) with customizable settings for each. In subfolder of your scene file.

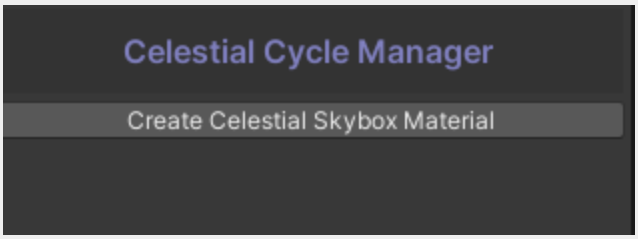


Apply Generate Celestial

This step applies the generated profile to **Celestial Time Manager**

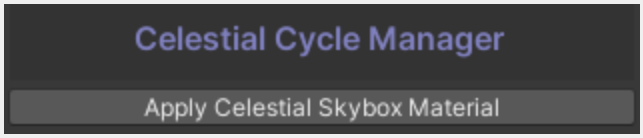


Step 5: Create Celestial Skybox Material



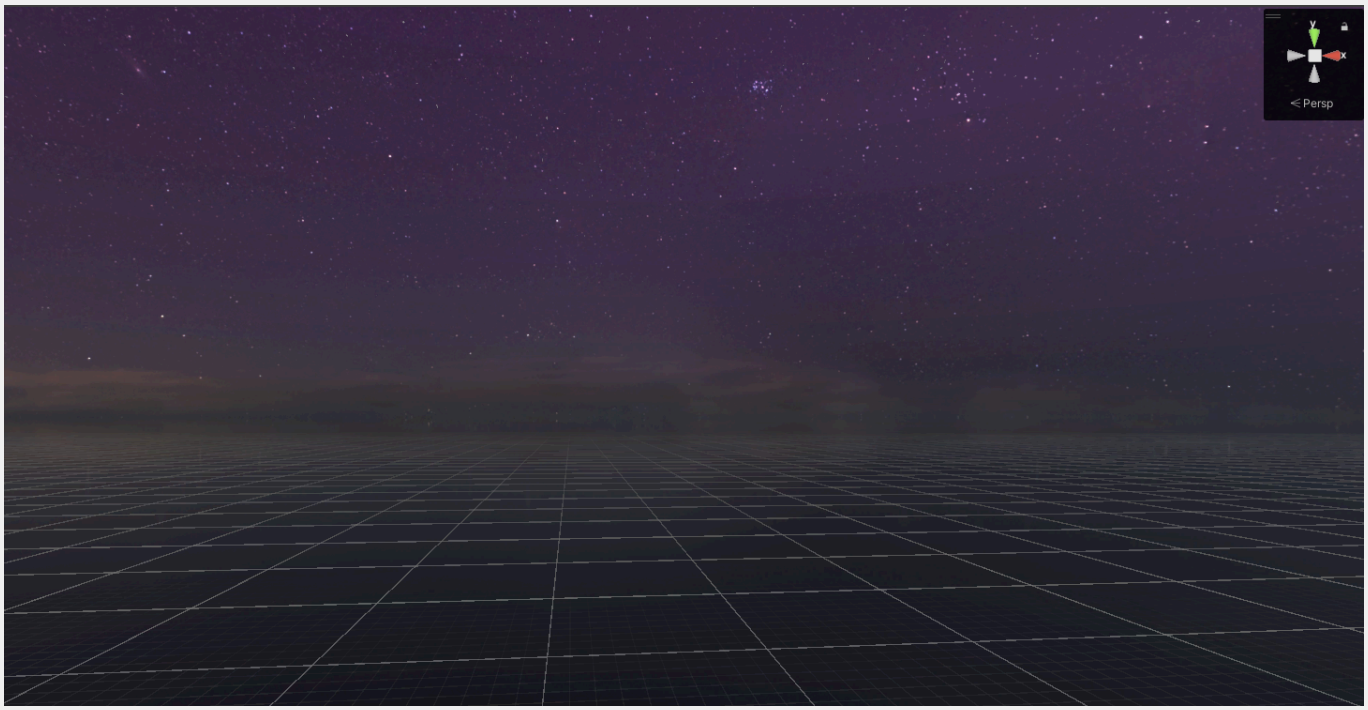
Create Celestial Skybox Material

This step generate copy of Celestial Skybox Material in the subfolder of your scene file



Create Celestial Skybox Material

This step applies generated skybox to your scene. Now your scene skybox will look like below image



Step 6: Configuring the Day-Night Cycle

Adjust Settings: Celestial Cycle Manager will now have all the settings appeared on the window instead. You can either edit those settings in Celestial Environment Manager gameobject or this Editor Window

Step 7: Testing and Tweaking

Preview the Cycle: Use the Unity Editor to preview the day-night cycle.

Enable Trigger and adjust Current Time Of Day to see the changes in environment. Adjust the cycle's settings in real-time to achieve the desired effect, thanks to the [ExecuteAlways] attribute of the Celestial Time Manager.

Fine-Tune: Experiment with different settings for light intensity, skybox materials, and ambient colors for each time period to refine your environment's look and feel.

Set Current Time to Morning Start Time and Edit the Morning Setting Below. All the changes will take effect in real-time.

Follow the same step for Noon, Evening, and Night

Step 8: Applying Final Adjustments

Finalize Settings: Once satisfied with the day-night cycle, ensure all settings are saved, and your scene is configured as intended.

Build and Run: Build your project to test the day-night cycle in your final game environment.

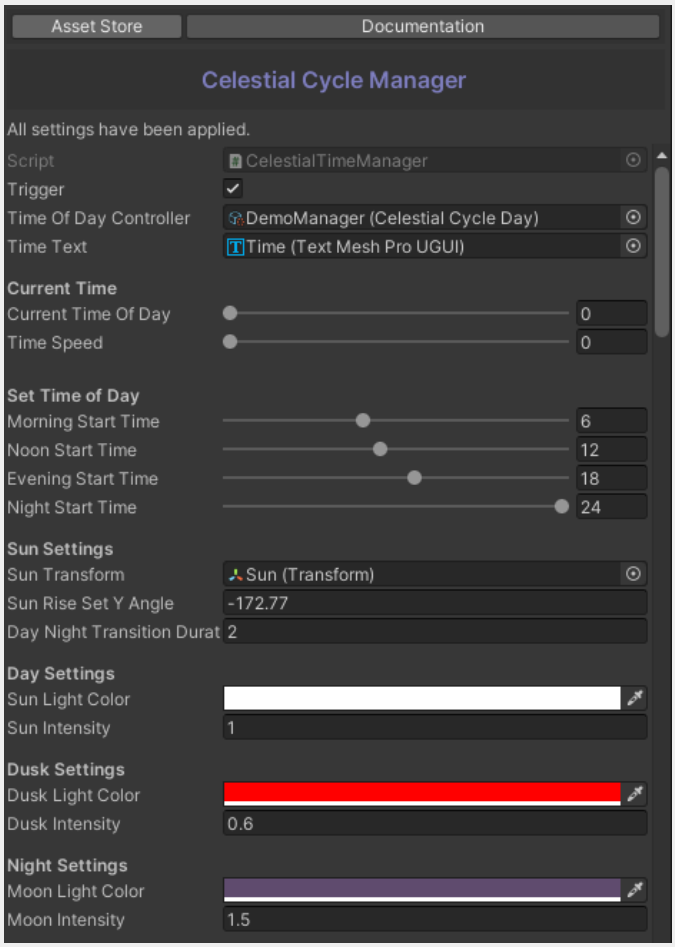
Step 9: Troubleshooting

If any of the step fails to work from the Editor Window. Work directly in Celestial Environment Manager and make sure there is no empty fields. More testing is required for the Editor Window version.

Time Text field can be left blank if not in use. Everything else must have proper objects applied to the fields.

Core Components

Celestial Time Manager



Purpose: The heart of the "Celestial Cycles" system, orchestrating seamless transitions between various times of day to mimic the Earth's natural cycles.

Functionality: Tracks in-game time, adjusting environmental settings based on predefined CelestialCyclePeriods to reflect the dynamic changes from dawn to dusk and beyond.

Trigger: Turn makes the Time Manager Active

Real-Time Updates with [ExecuteAlways]

The CelestialTimeManager component is enhanced with Unity's [ExecuteAlways] attribute, a powerful feature that significantly boosts your development workflow by enabling the system to run not only during play mode but also while editing in the Unity Editor. This means that any adjustments you make to the CelestialTimeManager or its associated CelestialCyclePeriods are applied and visible in real time, providing instant feedback on your environmental settings.

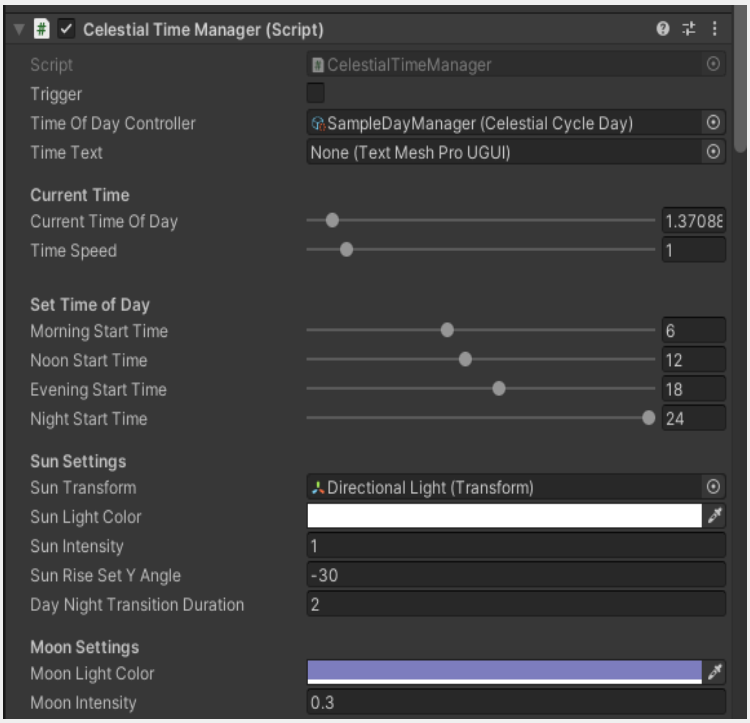
Time of Day Controller: where you input **Celestial Cycle Day** component explained below.

Time Text: Can add TextMesh Pro Text to here to visually represent the time in UI in 00:00 format

Current Time of Day: Moves from 0 - 24 representing 24 hours a day. Moving this slider will make the time move in the environment.

Time Speed: Used to increase how fast the time travels.

Set Time of Day: Celestial Cycle Period is designed to represent a specific phase of the day, such as morning, noon, evening, and night. The start times for these periods—morning start time, noon start time, etc.—determine when each respective environmental setting begins to apply in your game world.



Sun Transform: In Unity, a directional light is often used to represent the sun (during the day) and the moon (during the night) because it casts parallel light rays across the entire scene, similar to how sunlight or moonlight behaves in the real world.

Sun Light Color: Choose the light color for day time

Sun Intensity: Choose how bright the daylight is going to be

Sun Rise Set Y Angle: Can determine when sun rises and falls in morning and evening

Day Night Transition: Determine how fast the transition between day skybox to night skybox

Moon Light Color: Choose the light color for night time

Moon Intensity: Choose how bright the nighttime is going to be

Celestial Cycle Day

Purpose:This component is a fundamental building block of the "Celestial Cycles" system, designed to encapsulate and manage a comprehensive set of environmental parameters for specific times of the day, such as dawn, noon, dusk, and night. Each instance of CelestialCycleDay is a representation of a unique phase in the day-night cycle, equipped with its own set of customizable settings

to accurately reflect the natural variations in lighting, color, and atmosphere.

Functionality: Houses settings for different **Celestial Cycle Period** like Morning, Noon, Evening, and Night

Celestial Cycle Day

Purpose: CelestialCyclePeriod serves as a blueprint for a specific time phase within the day-night cycle, holding essential environmental parameters such as lighting intensity, sky and equator colors, and skybox materials. The core functionality of CelestialCyclePeriod involves transitioning these settings from one period to the next as the in-game time moves forward.

Celestial Cycle Day Properties

- **Skybox Shader Property**
  - **Tint**: The Tint property allows you to change the color of the skybox. By applying a tint, you're essentially overlaying a color on top of the skybox's existing colors. This can dramatically alter the atmosphere of your scene, simulating different times of day or weather conditions.
  - **Exposure**: Exposure controls the brightness of the skybox. Adjusting the exposure can simulate the intensity of the sun or moon, affecting how light or dark your scene appears. It's like adjusting the exposure setting on a camera; higher exposure results in a brighter image, while lower exposure darkens the scene.
  - **Contrast**: Contrast affects the difference between the light and dark areas of the skybox. Increasing contrast will make the bright areas brighter and the dark areas darker, leading to a more dramatic and visually striking sky. Reducing contrast, on the other hand, softens the differences, creating a more subdued effect.
- **Sun / Moon in Sky**
  - **Sun Color:** Will change the color of physical Sun and Moon in the sky
  - **Sun Size:** Will change the size of physical Sun and Moon in the sky
  - **Moon Phase:** Can be used to make moon change shape, Full Moon, Half Moon, Crescent
- **Skybox Gradient Settings**
  - **Sky Color Blend:** Will determine amount of effect of properties below
  - **Sky Color**: Sky Color determines the color of the ambient light that comes from above. It simulates the light that the sky casts on objects in your scene. This setting is particularly influential on surfaces that face upwards or are exposed to the sky directly.
  - **Sky Blend**: Change how much of the sky is effected by **Sky Color** change
  - **Equator Color**: Equator Color controls the ambient light's color at the horizon line or the middle part of the scene. It's meant to simulate the scattering of light around the equator where the sunlight is most direct but gets diffused across the atmosphere.
  - **Equator Blend:** Change how much of the horizon is effected by **Equator Color** change
  - **Ground Color**: Ground Color sets the color of the light that's coming from below, simulating the light reflected from the ground back onto objects. This setting is crucial for objects that are under other objects or have surfaces facing the ground.
  - **Ground Blend:** Change how much of the ground is effected by **Ground Color** change

- Shader Fog Setting - More explanation about Shader Fog in **CelestialSkybox** Shader section below
    - **Custom Fog Color:** Can change color of fog in Skybox horizon. (Need to enable Custom Fog option in the material to enable color change)
    - **Fog Thickness:** Can determine how dense the fog is in the horizon
    - **Fog Smoothness**: Makes the fog expand further to the sky
    - **More properties in CelestialSkybox material**
- **Lightning**
    - **Skybox Intensity Multiplier:** This setting adjusts the overall intensity of the ambient light in your scene. Ambient light is the general illumination that lights up everything in the scene equally, without coming from a specific direction. The Environment Lighting Intensity Multiplier scales this light's brightness, making the scene appear brighter or darker without altering the direct light sources like the Sun or lamps.
- **Environment Reflection**
    - **Reflection Intensity:** This setting controls the intensity of the reflections generated by the environmental lighting in your scene. Unity uses Reflection Probes and Light Probes to capture and apply these reflections on surfaces and objects. The Environment Reflections Intensity Multiplier scales the brightness of these reflections, affecting how strongly objects reflect the environment around them.
- **Fog Setting**
    - **Fog Enable:** Turn Unity Fog On and Off
    - **Fog Mode:** Can select from one of Unity Fog Modes (Linear, Exponent, Exponent Squared)
    - **Fog Color:** Fog Color determines the visual hue of the fog within the scene.
    - **Fog Density:** Fog Density controls how thick or thin the fog is, affecting its opacity. This setting is especially relevant for Exponential and Exponential Squared Fog modes.
    - **Fog Start Distance**:Fog Start Distance specifies at what distance from the camera the fog will begin to appear. This setting doesn't affect the entire scene uniformly but rather establishes where along the camera's view the fog's effect starts.
    - **Fog End Distance**:Fog End Distance determines the distance from the camera at which the fog reaches its maximum density or opacity. Beyond this point, objects are fully obscured by the fog.

Shader Components

Celestial Cycle Skybox

Shader Capabilities

The core of "Celestial Cycles" visual magic lies in its sophisticated shader, designed to dynamically simulate natural lighting and atmospheric conditions. This shader utilizes a combination of cubemaps, color adjustments, and celestial bodies to create immersive day-night transitions.

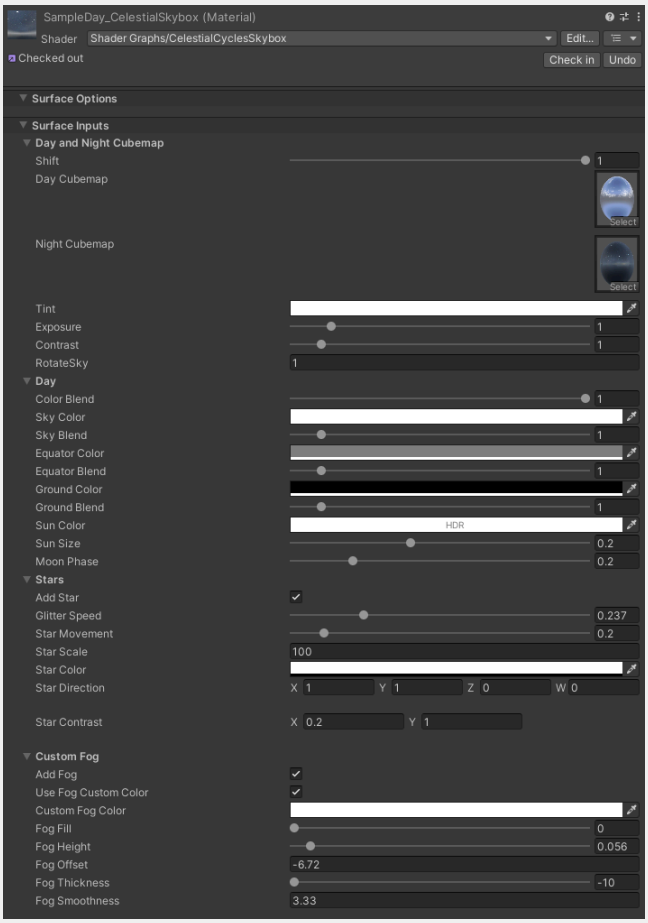
Cubemaps for Day and Night

Functionality: The shader employs two cubemaps—one representing daytime and the other nighttime. As the CelestialTimeManager progresses through different times, it seamlessly blends between these cubemaps reflect the transition from day to night and vice versa.

Customization: Users can specify their own cubemaps for day and night, allowing for a high degree of visual customization to match the game's artistic style.

Dynamic Ambient Lighting

Sky, Ground, and Equator Colors: Similar to Unity's built-in ambient lighting settings, this shader allows for the dynamic adjustment of sky, ground, and equator colors. These colors shift according to the CelestialCyclePeriod settings, ensuring the ambient light matches the time of day.



to

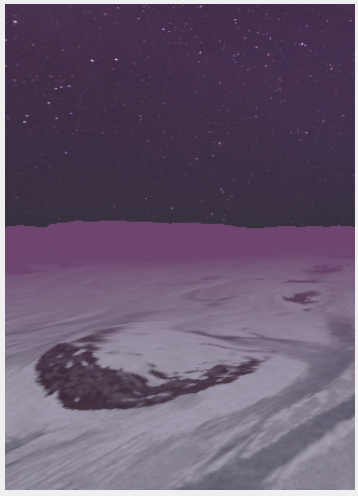
Celestial Bodies: Sun, Moon, and Stars

Sun and Moon: The shader incorporates mechanisms for displaying the sun and moon within the skybox, positioning them accurately in relation to the directional light in the scene. This ensures that their movement across the sky mimics natural celestial patterns.

Stars: At night, the shader introduces a glittering starfield, enhancing the night sky's realism. The intensity and visibility of the stars are tied to the night-time cubemap and CelestialCyclePeriod settings.

Custom Fog On

Custom Fog Off



Custom Fog System

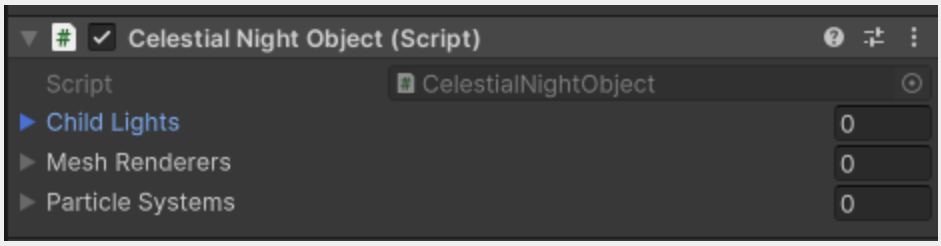
Seamless Integration: A custom fog feature is included to ensure a smooth visual transition from the horizon to the sky, effectively eliminating seams between the scene's fog and the skybox.

Dynamic Adjustment: The fog's intensity and color can be adjusted per CelestialCyclePeriod, allowing for atmospheric changes that align with the time of day, such as denser fog at dawn or dusk.

Additional Components

Celestial Night Object

Capabilities



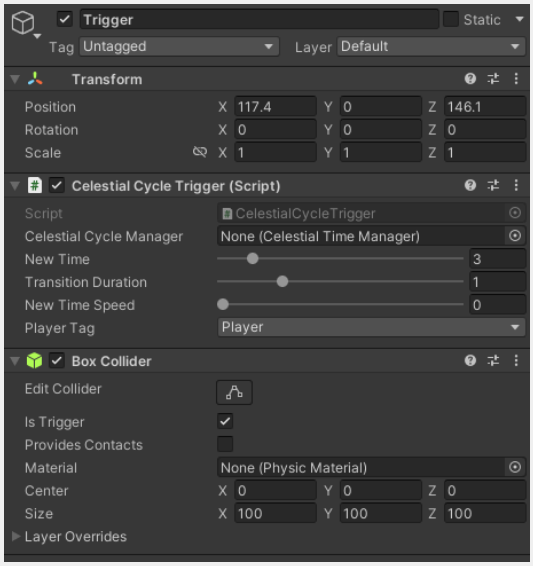
The CelestialNightObject is like a smart switch for the lights and special effects in your game's world that knows exactly when it's day and when it's night. During the day, it turns off certain lights and effects, making everything look as bright and normal as you'd expect on a sunny afternoon. But as the night falls, it magically turns these lights and effects back on, casting a beautiful glow and bringing a lively ambiance to the scene, just like how streetlights illuminate when it gets dark outside.

How to use: Create a GameObject with **Celestial Night Object** Component, and add all the lights, gameobjects, and particle system that requires to be toggled off during daytime and toggled on during nighttime. Script will automatically grab all the Lights, mesh renderers and particle system and toggle it on and off. (For Mesh Renderer it will toggle on and off the emissive feature in the material

For the emissive materials to be toggled on and off it requires _EmissiveIntensity as a property. Sample shader is provided that works with the script

Celestial Cycle Trigger

Capabilities



The Celestial Cycle Trigger is like a smart switch for the time that you can place on different area to toggle different time in different part of the scene.

How to use: Create a GameObject with **Celestial Cycle Trigger** Component, Add any collider and set value and select the collect Tag for the player to make it trigger

Comprehensive Environmental Control

Each of these shader features is designed to be managed through the **Celestial Cycle Period** settings, allowing developers to create detailed environmental profiles for different times of day. From the golden hues of sunrise to the cool blues of twilight, "**Celestial Cycles**" empowers creators to bring their virtual worlds to life with stunning authenticity.

By harnessing the power of the Celestial Time Manager, users can ensure these diverse settings seamlessly transition throughout the day, providing players with a dynamic and immersive experience that reflects the natural beauty of the passing time.

Conclusion

The "Celestial Cycles: Dynamic Day-Night System" offers an unparalleled toolkit for simulating realistic day-night cycles in Unity. Through its advanced shader and comprehensive environmental settings, developers can craft breathtaking scenes that engage players and elevate the gaming experience. We encourage exploration and experimentation with the system to discover the full potential of your game's environments.