

◆◆◆ The Floating Island – Ghibli Style Scene Rendering

Date: Sep 2025

Role: Solo

Engine: Unreal, Houdini, Blender, Substance Designer

Demo Video: <https://youtu.be/6rq0kAFhATg>

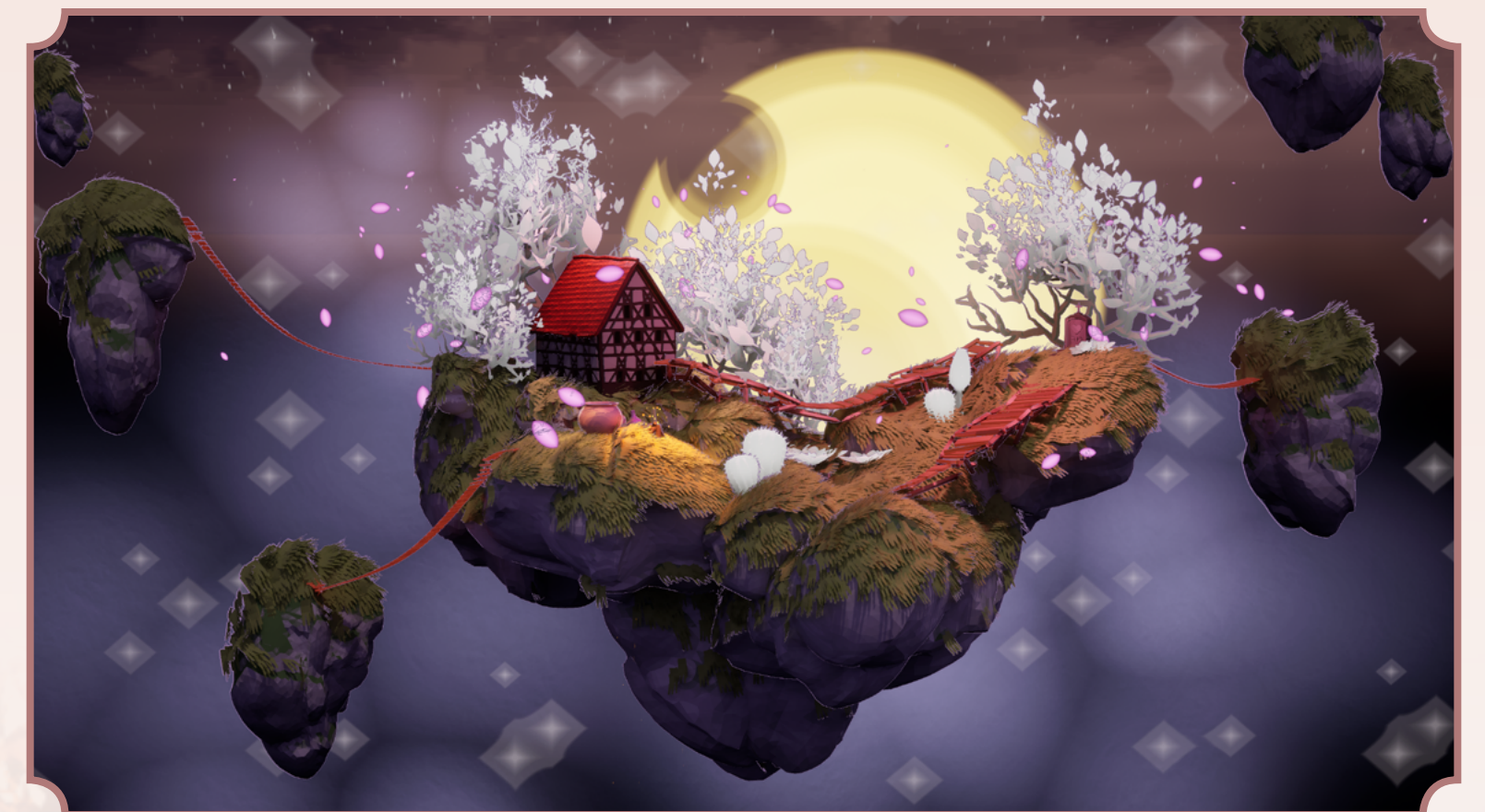
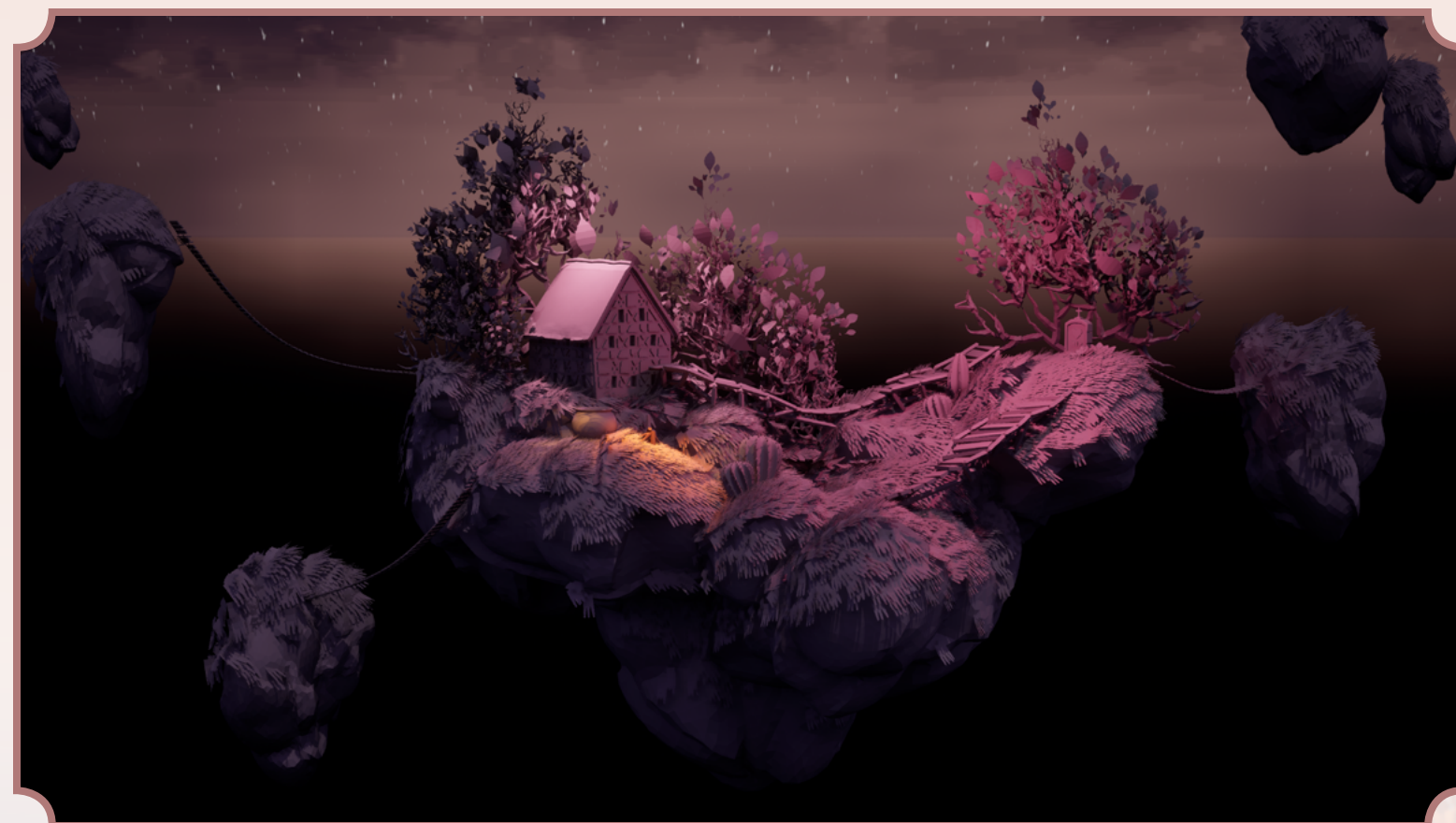
Introduction: This project builds a floating-island environment inspired by Studio Ghibli's painterly worlds. I reused procedural assets from my Houdini PCG set (rocks, vegetation, bridge, house, etc.) and a few props made using Blender, then developed a group of stylized materials and FXs in Unreal to support both a warm daytime look and a cooler, mysterious nighttime variation.



◆ Outcome

The final scene is presented in two lighting setups built around the same geometry and material library. In the day version, bright, saturated colors and soft shadows create a calm, sun-washed mood, with lighting that highlights the yellow-green grass and warm roof tones against a pale sky. In the night version, a cooler palette, localized warm light from the woodpile, and glowing potion bottles push the island toward a “witch’s house” atmosphere while preserving the overall composition.

Because the scene is driven by color palettes and parameterized materials rather than heavy post-processing, switching between day and night only requires adjusting global lighting, a few material values, and post-process settings instead of rebuilding assets. The combination of Houdini-generated geometry, reusable master materials (rock, grass, foliage, bottles, outlines, fog), and lightweight FX makes the scene easy to iterate on and provides a small stylized asset library that can be reused in future projects.

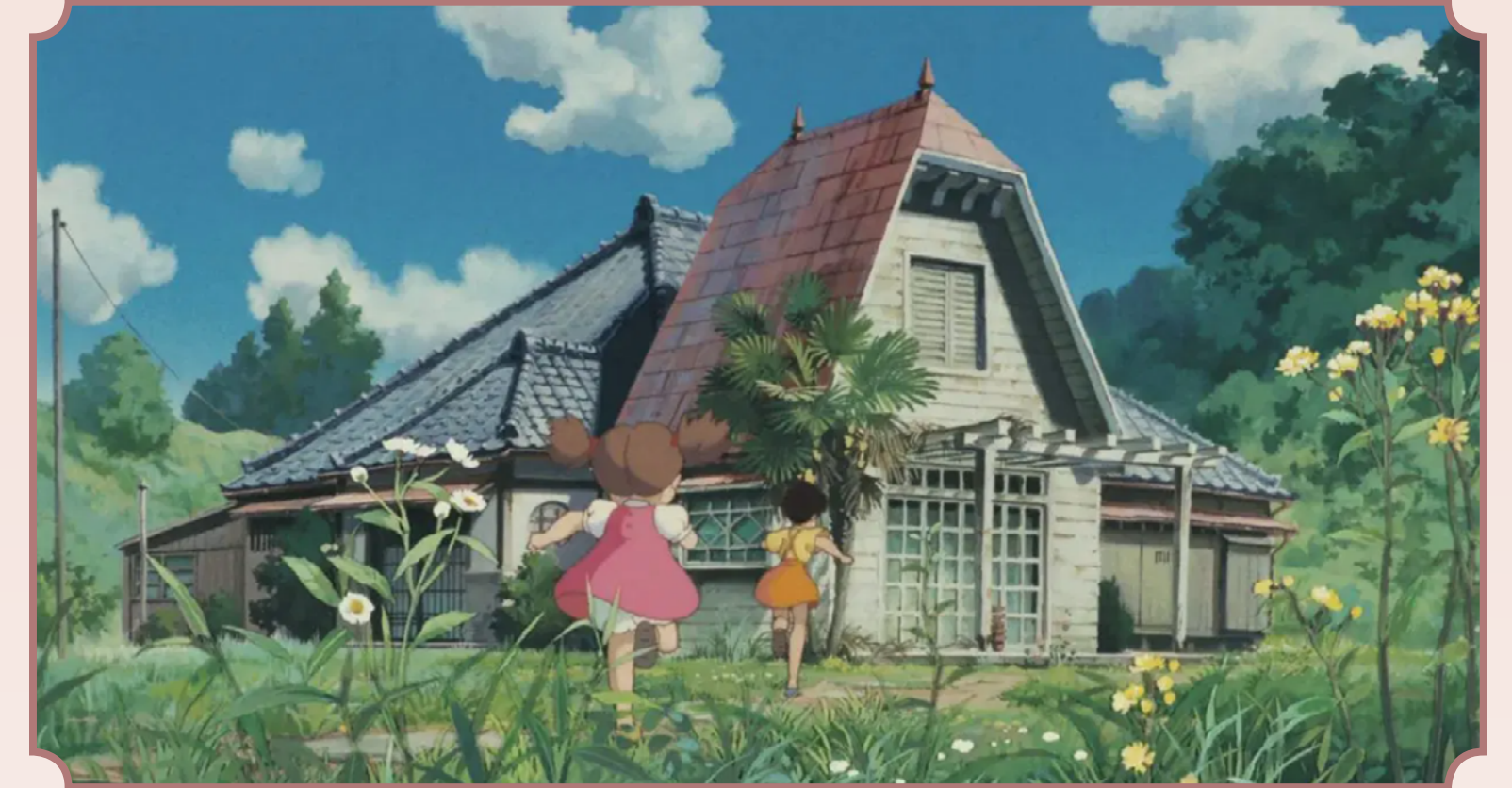
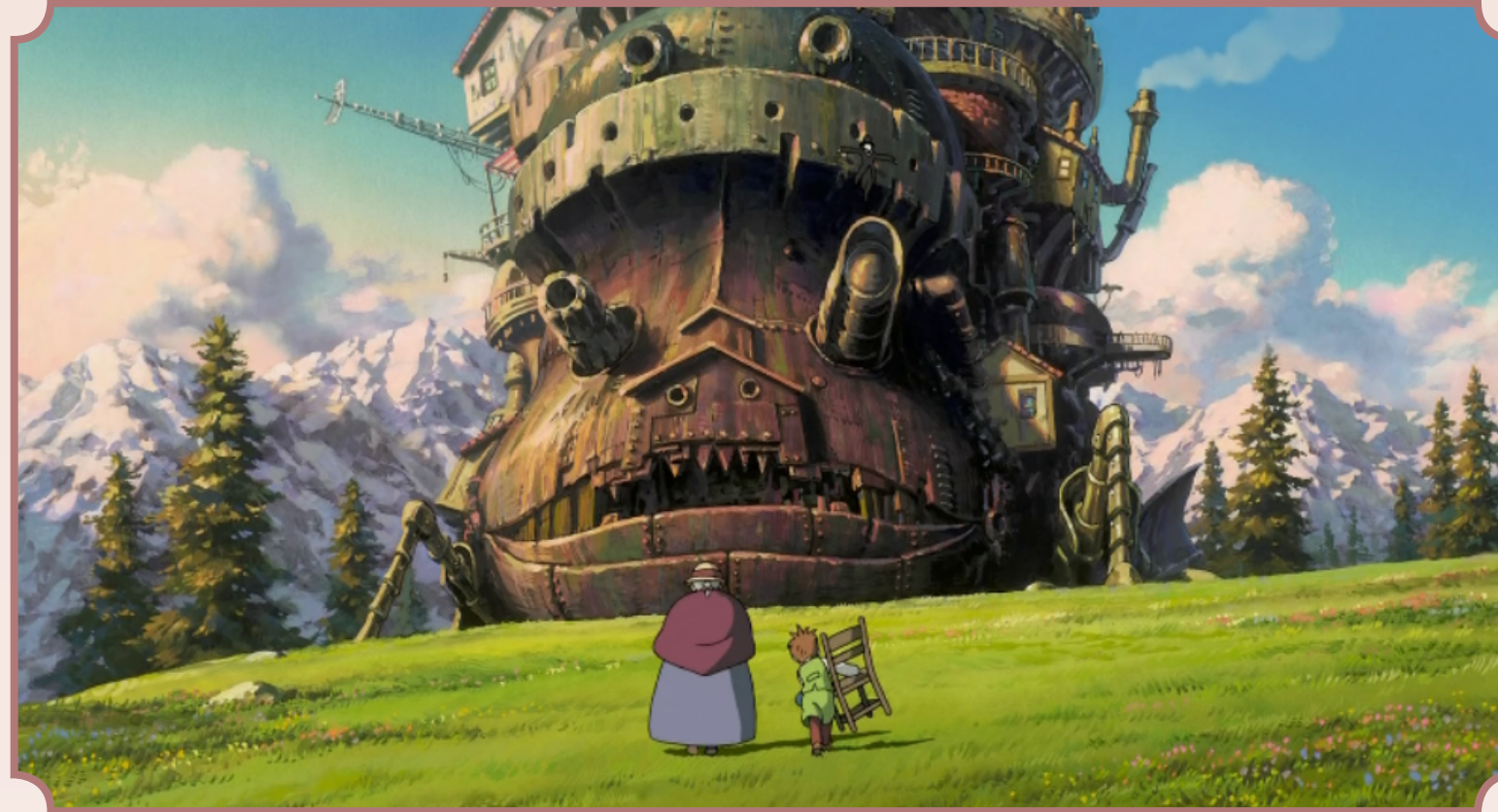
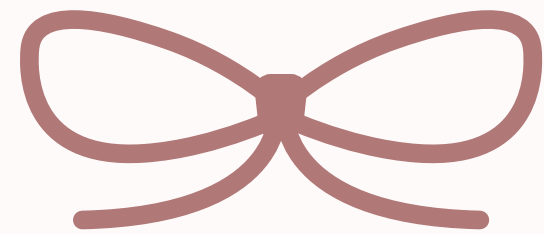


◆ Inspiration & Art Direction

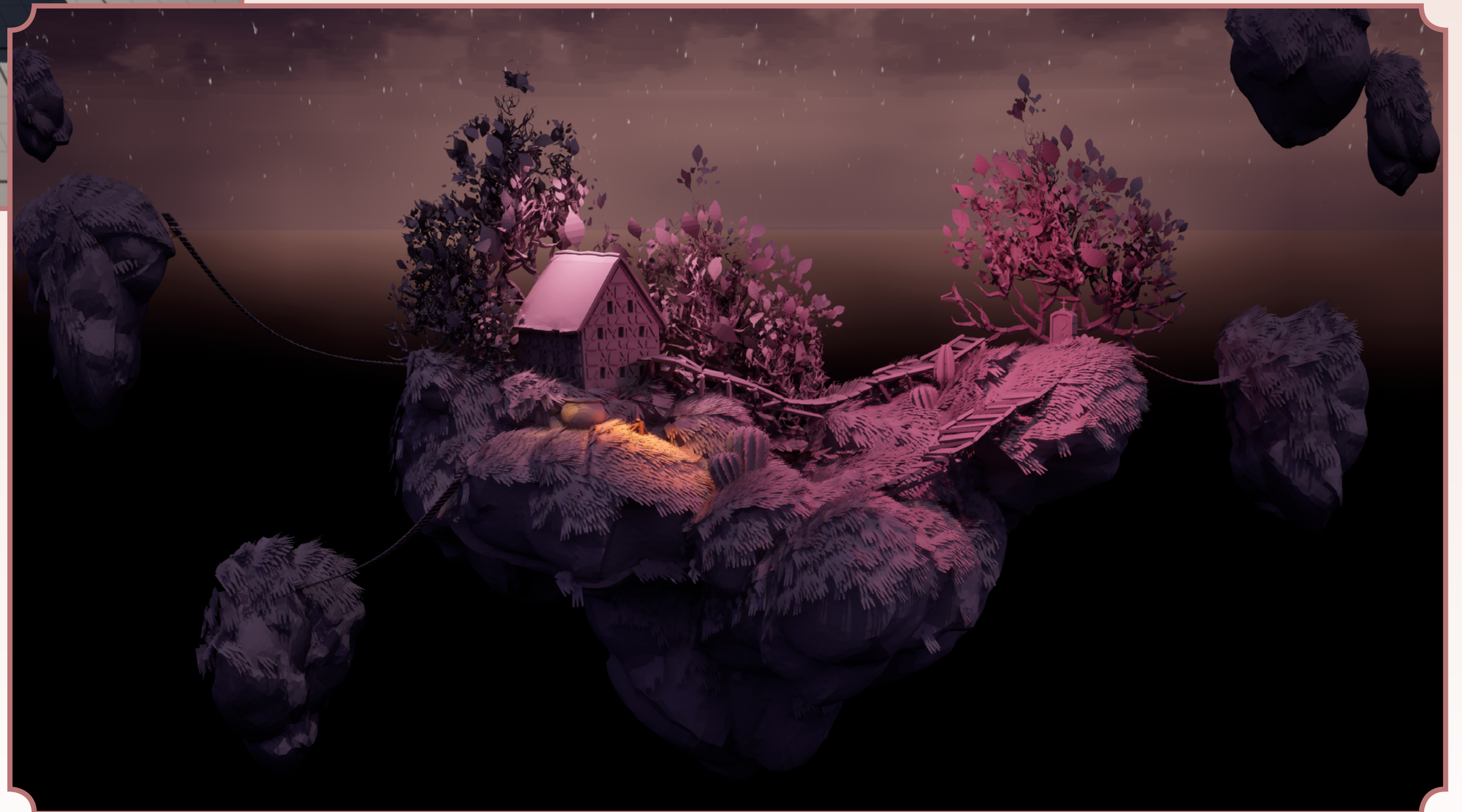
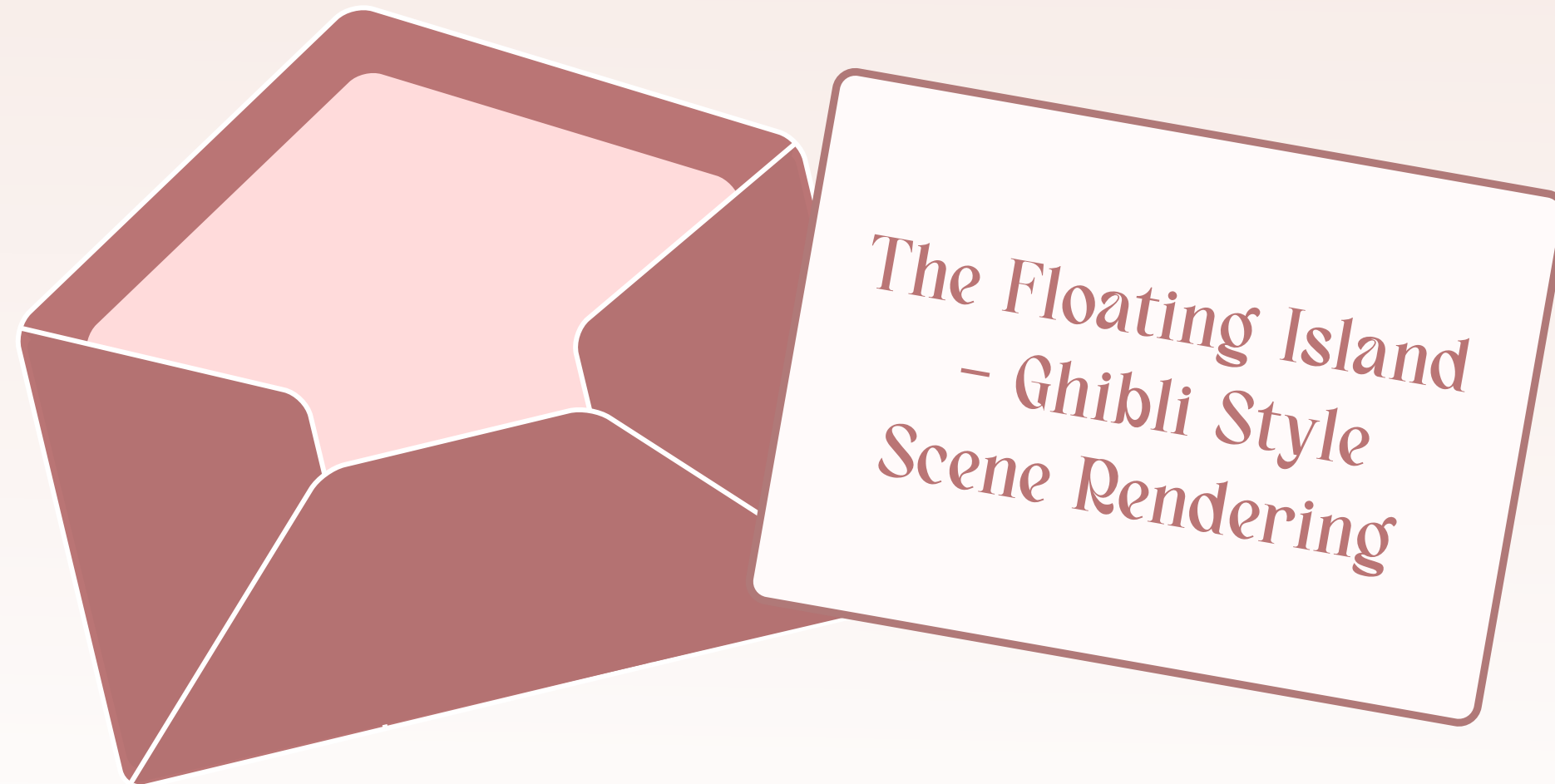
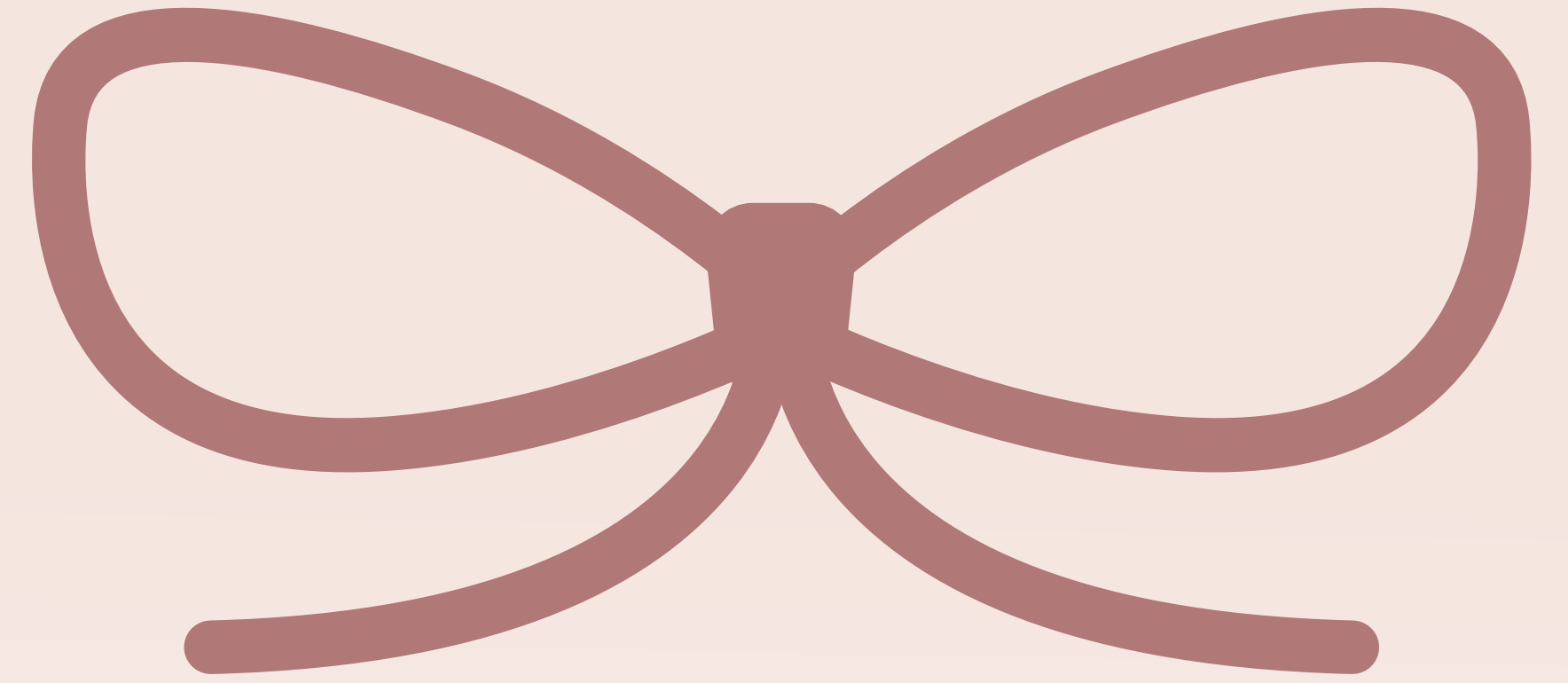
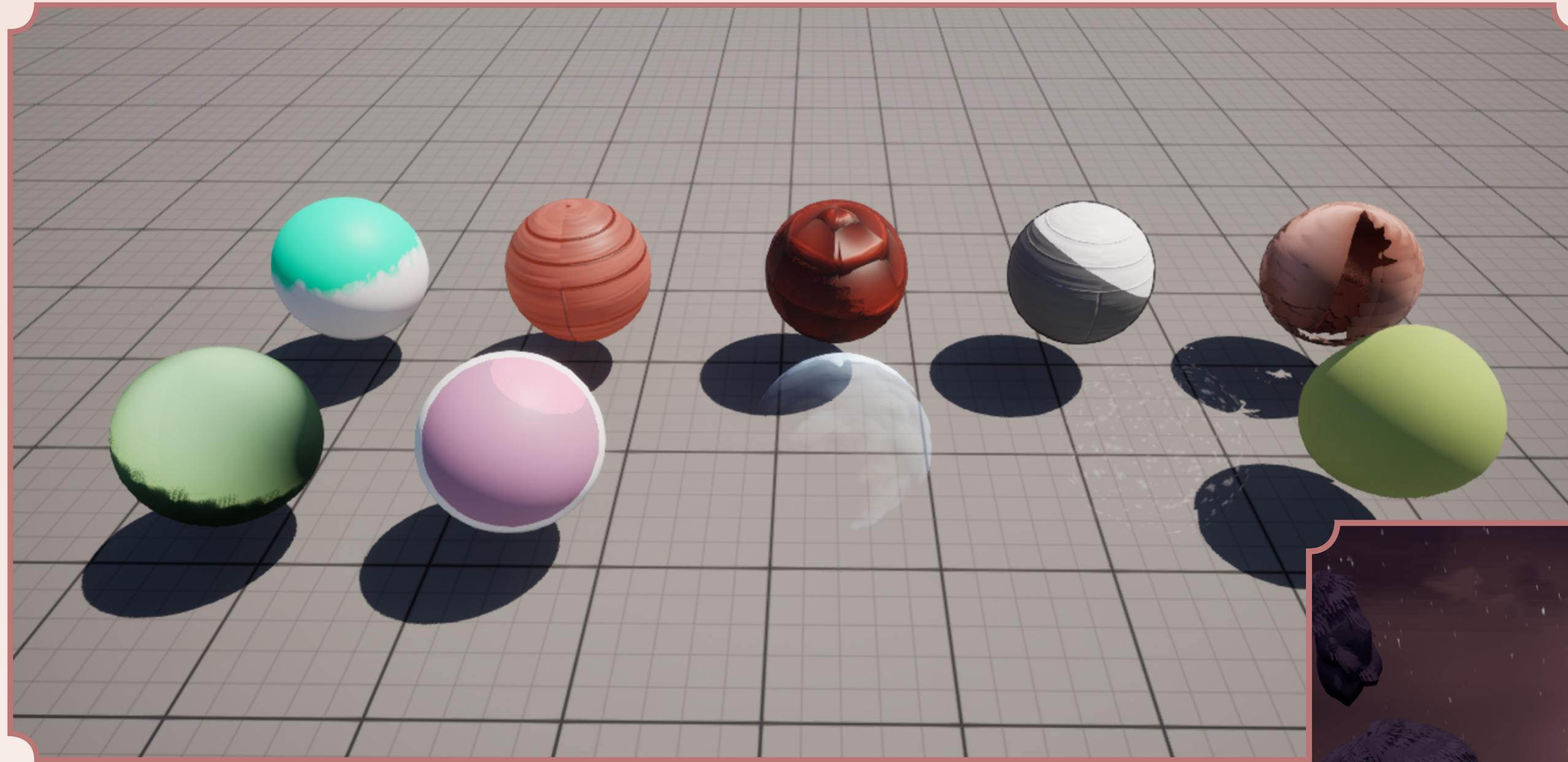
Inspired by Studio Ghibli, I wanted to create a floating island to mimic the painterly worlds and cozy, whimsical mood, and experiment with colors and tones to create a different feeling at another time of day.

Day: Bright, saturated palette; soft shadows; calming, sun-washed mood.

Night: Cooler palette; witch's house vignette with potion bottles and a cauldron; added woodpile for localized warm light contrast and story hints.



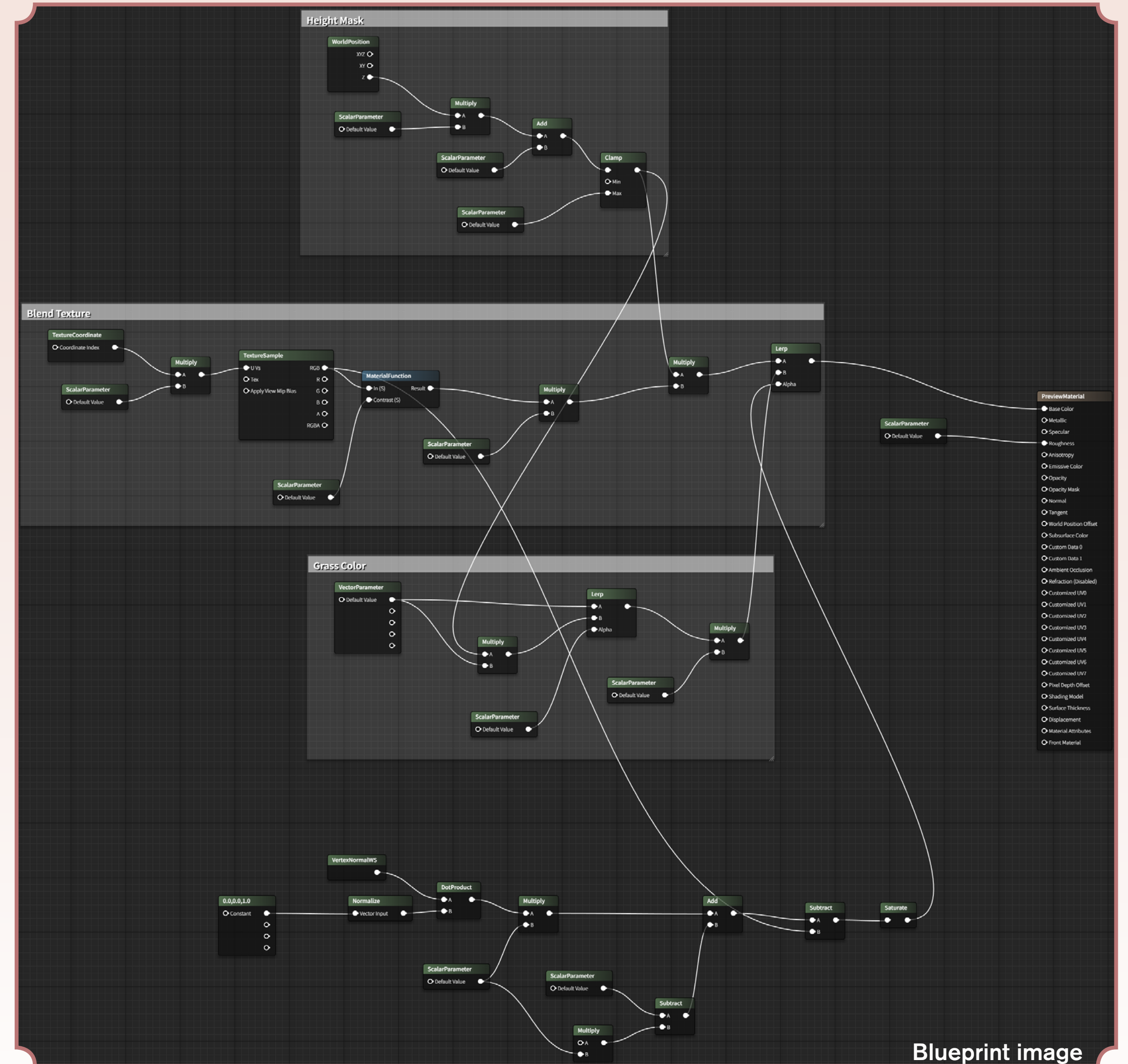
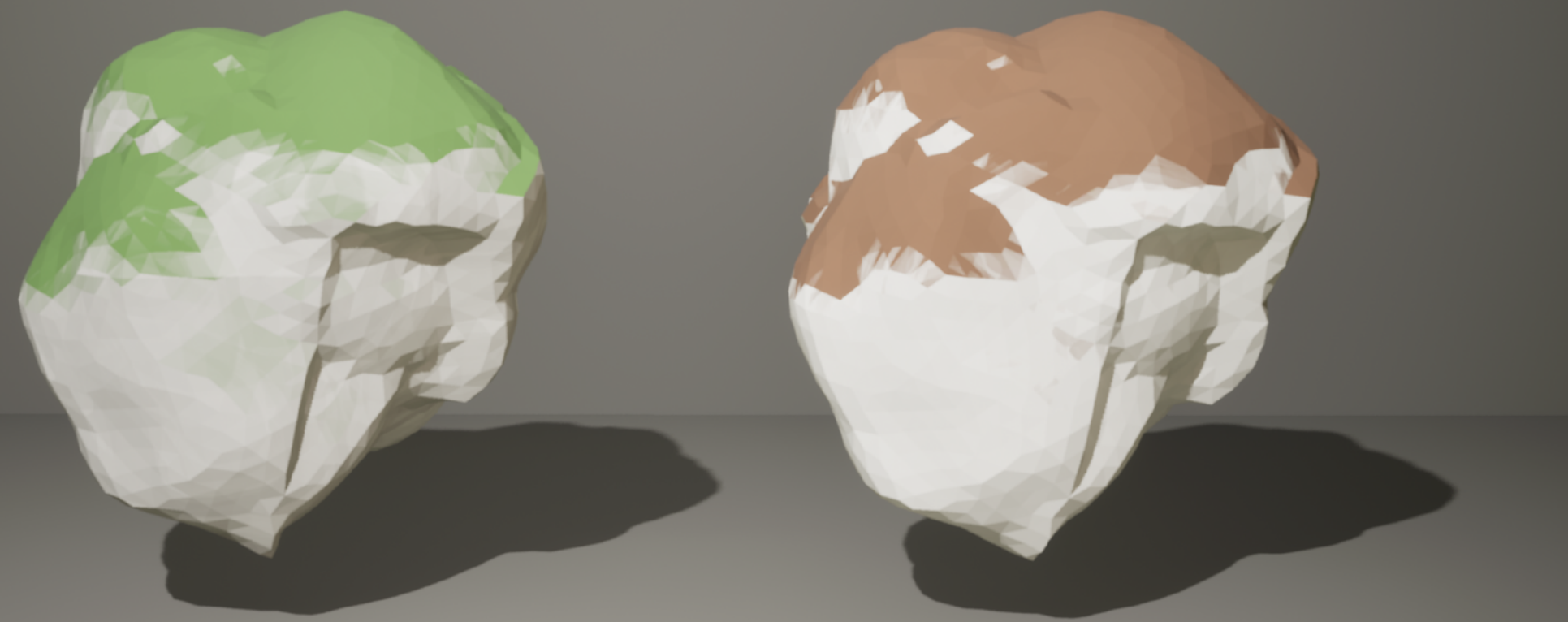
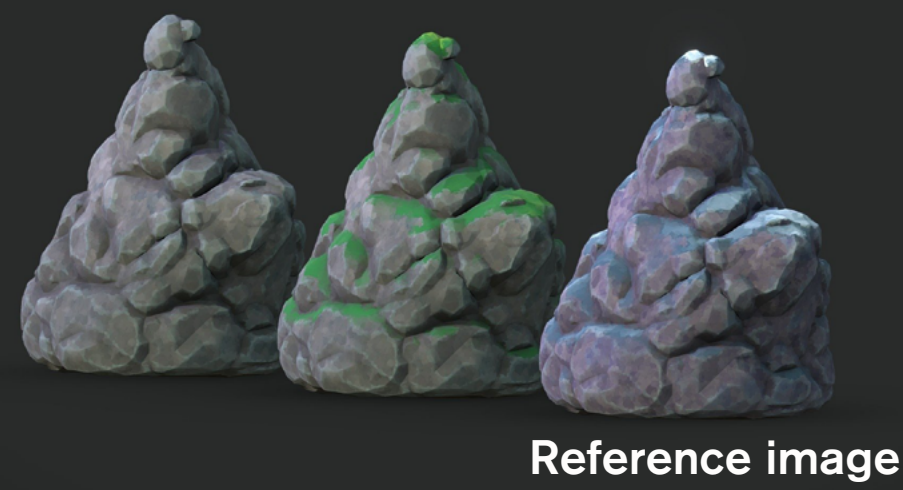
◆ Models & Materials



◆ Models & Materials

Rock/Dirt

The rock material supports all floating-island meshes generated in Houdini and aims to mimic a hand-painted, brush-stroke look rather than realistic stone. It provides a consistent base for both day and night palettes.

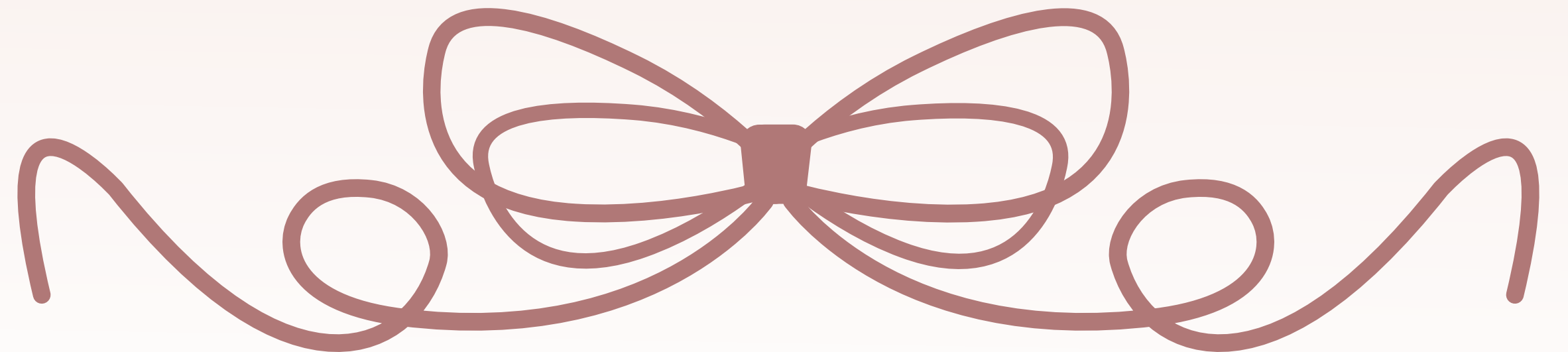
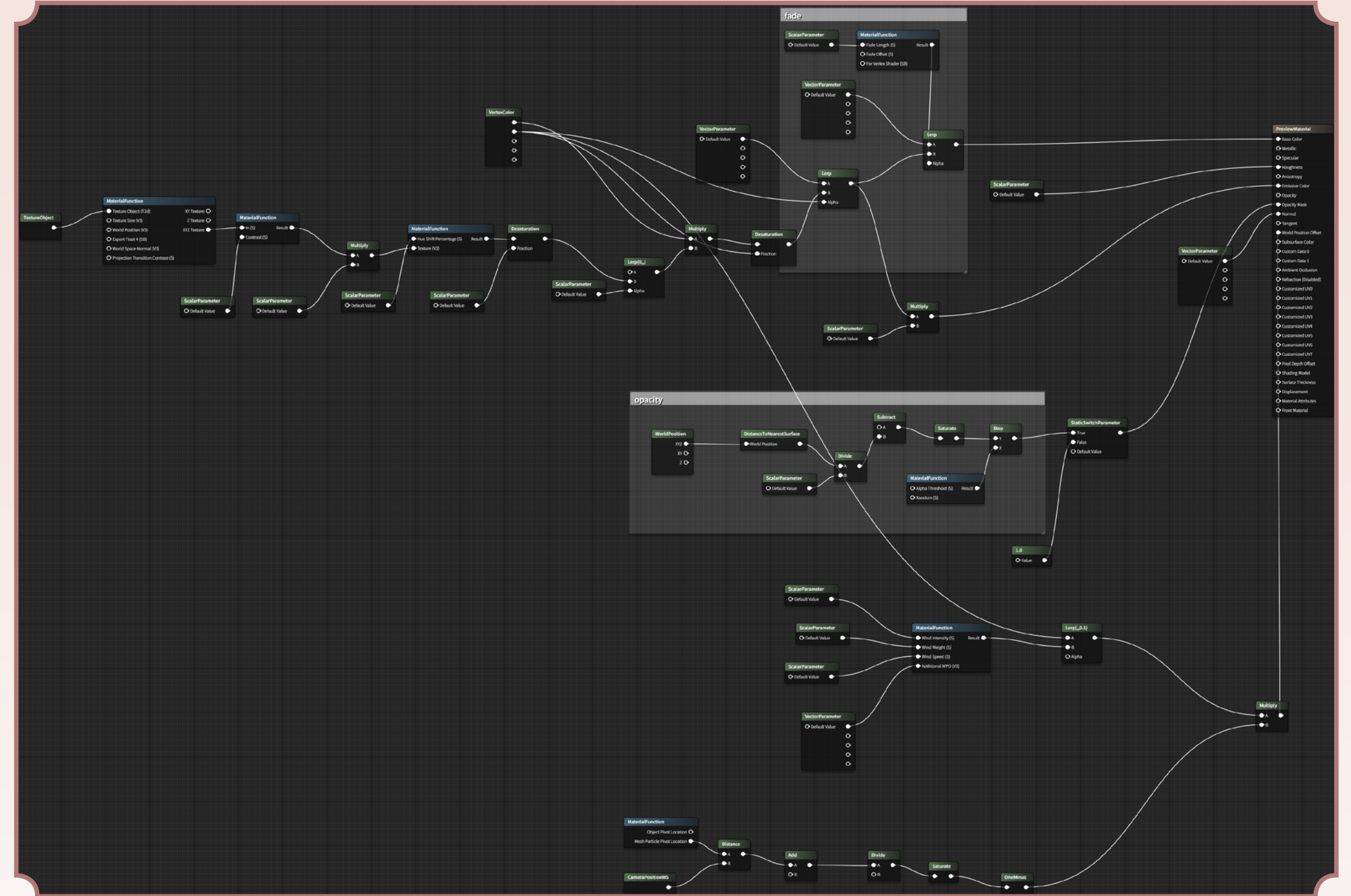
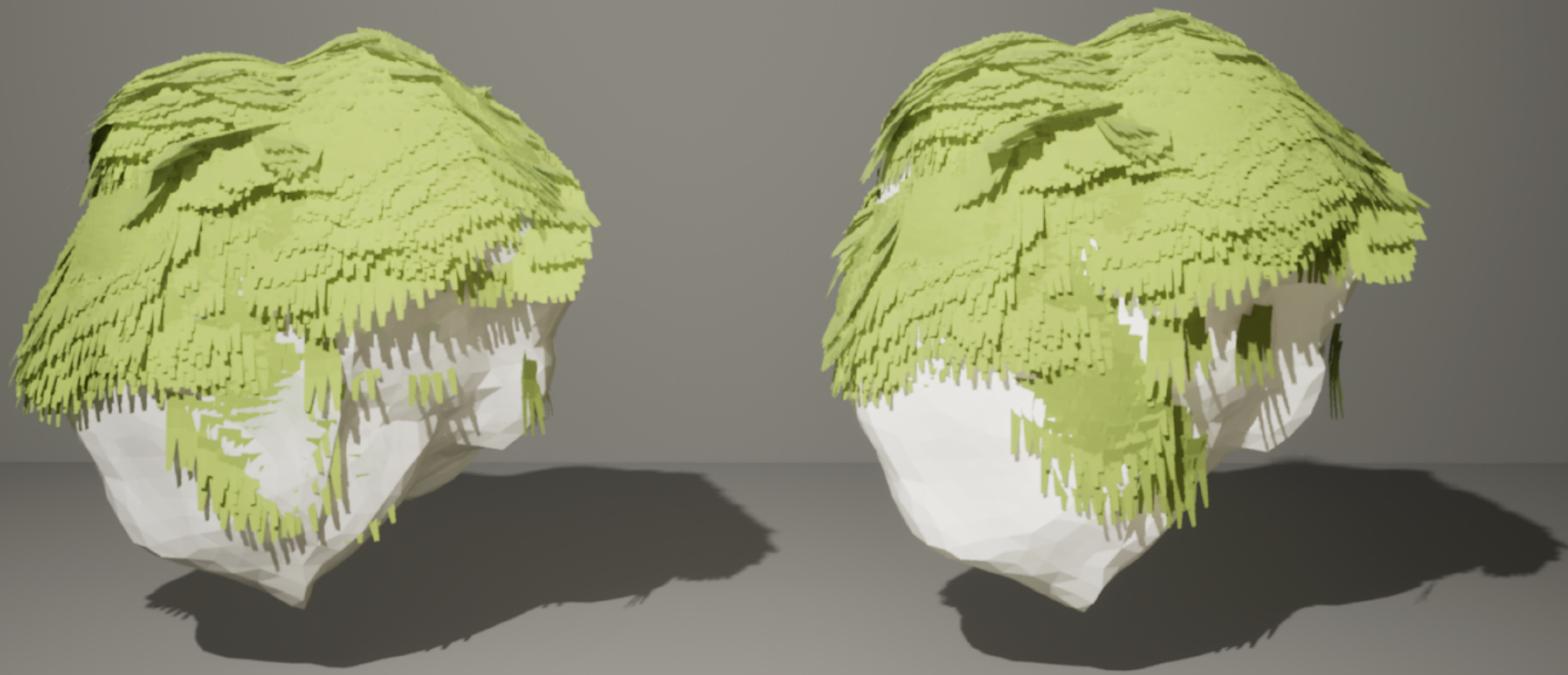


Models & Materials

Grass

The grass system adds a fluffy, stylized layer of grass on top of the terrain, with a soft fade at the edges so the island silhouette stays clean and readable.

Grass layer is generated using Houdini, it scatters simple grass meshes or cards across the top surfaces of the rocks, aligning them to the terrain normal. In Unreal, the grass material uses a height / mask branch to blend from solid grass color to transparency, creating the fade-out toward the edges. Additional scalar parameters control overall brightness and saturation so the same material can be reused for both day and night.

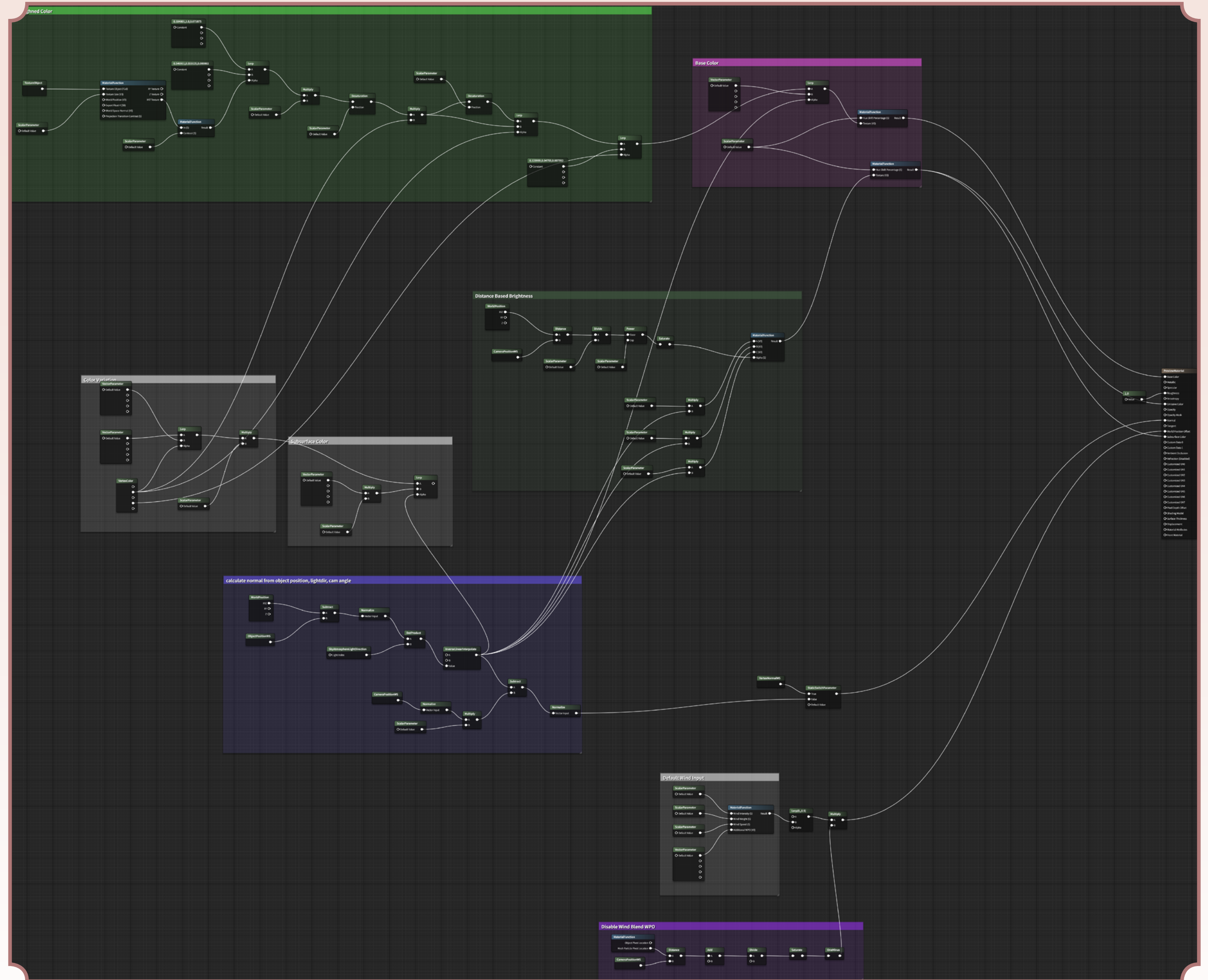


◆ Models & Materials

Tree Leaves/Flowers

I want the tree and flower clusters to feel light, fluffy, and slightly graphic, echoing Ghibli tree shapes.

In Unreal, I built a foliage material that takes the sky-light direction and vertex normals, uses their dot product to drive a simple two-tone ramp between light and shadow colors, and multiplies that with a basic leaf texture for breakup.

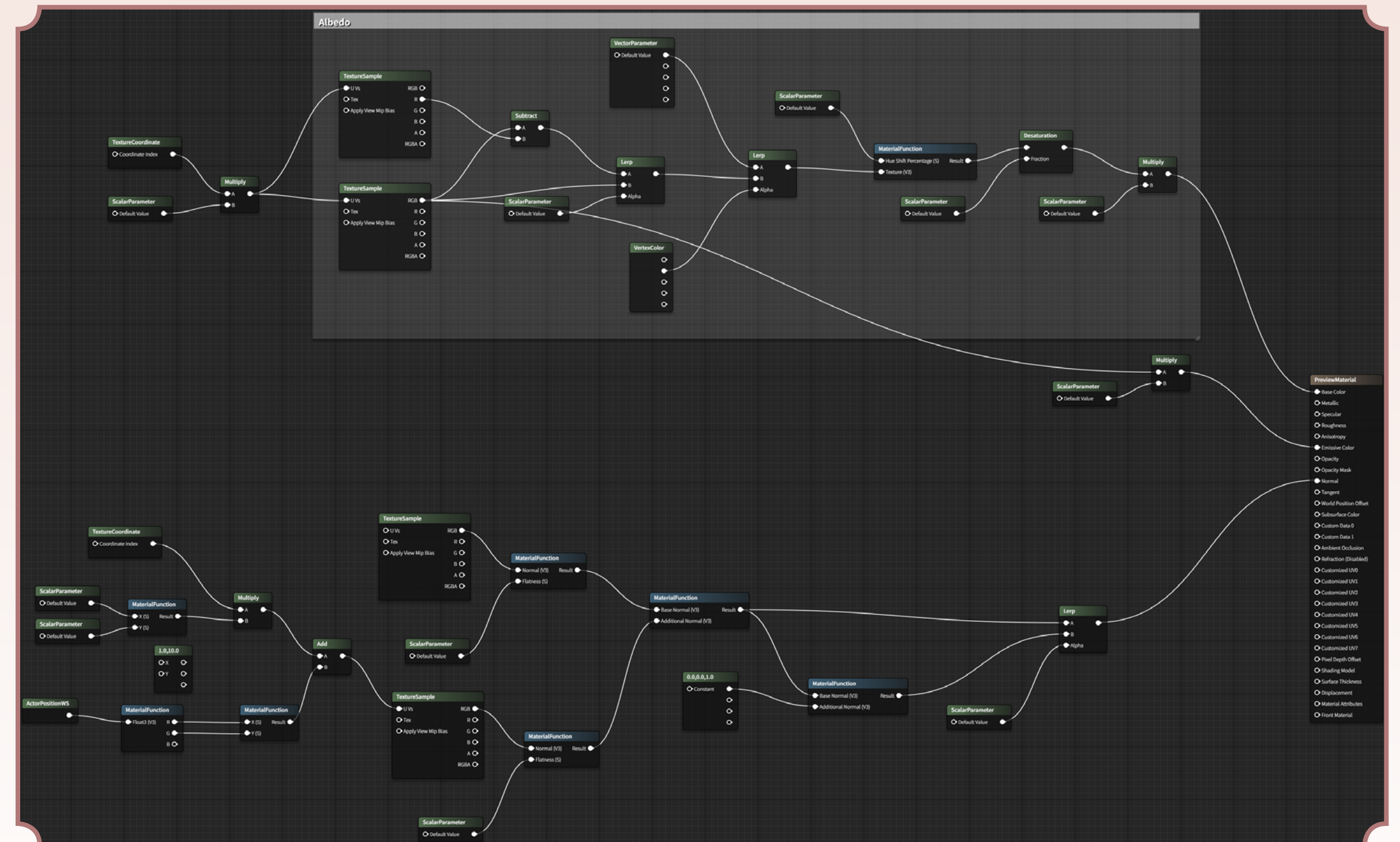
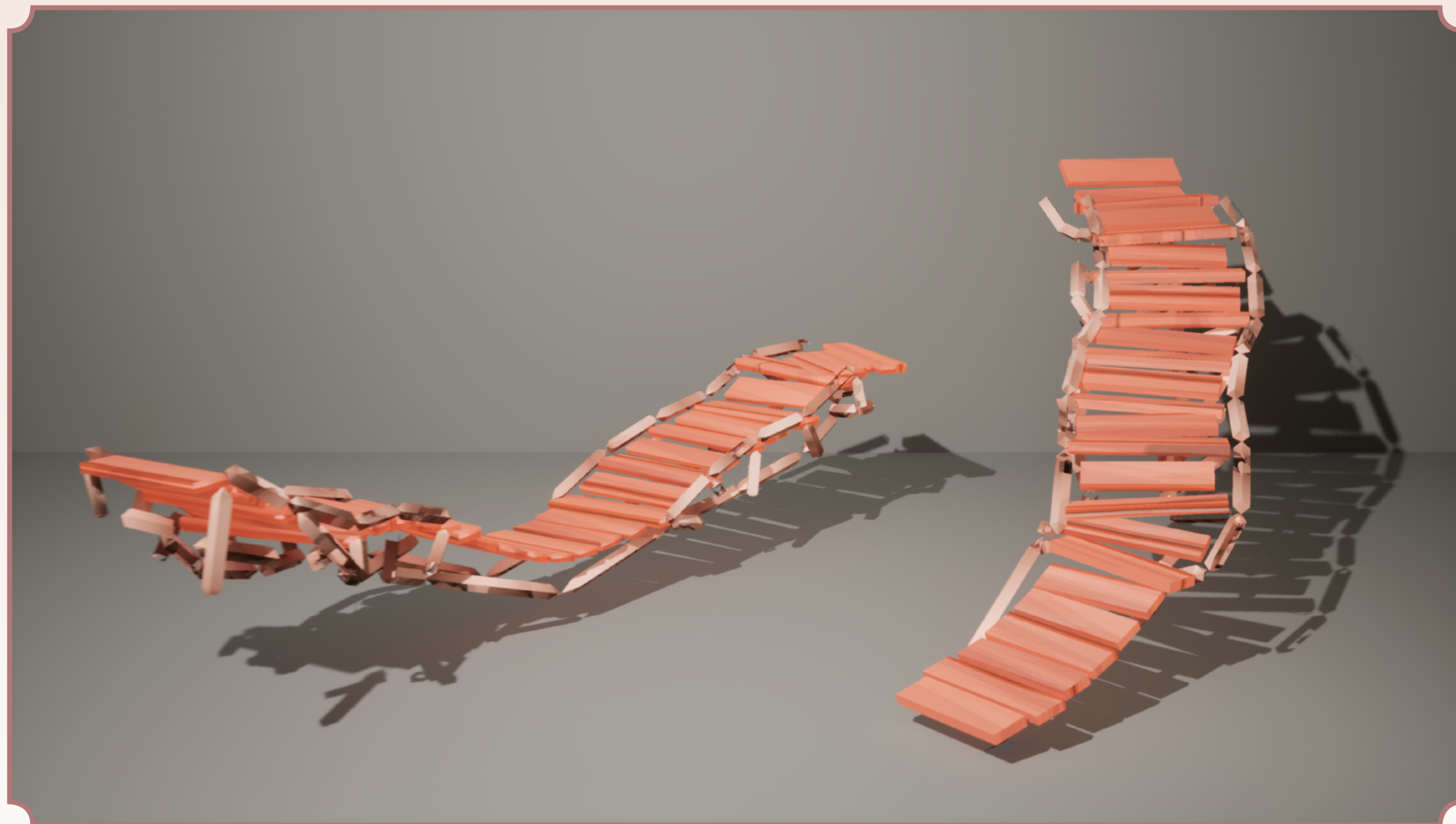


◆ Models & Materials

Bridge/Rope

The bridge and rope connect the main island to the surrounding rocks, guiding the viewer's eye.

Bridge geometry is procedurally generated in Houdini to follow the terrain profile and automatically adjust plank spacing and curvature. In Unreal, the bridge material blends a stylized wood color with simple shadow / highlight controls, so planks and ropes stay in the same palette as the house and rocks. The material graph exposes parameters for color tint and brightness, letting the bridge be warmed up for day or slightly desaturated for night while keeping the same underlying shader.

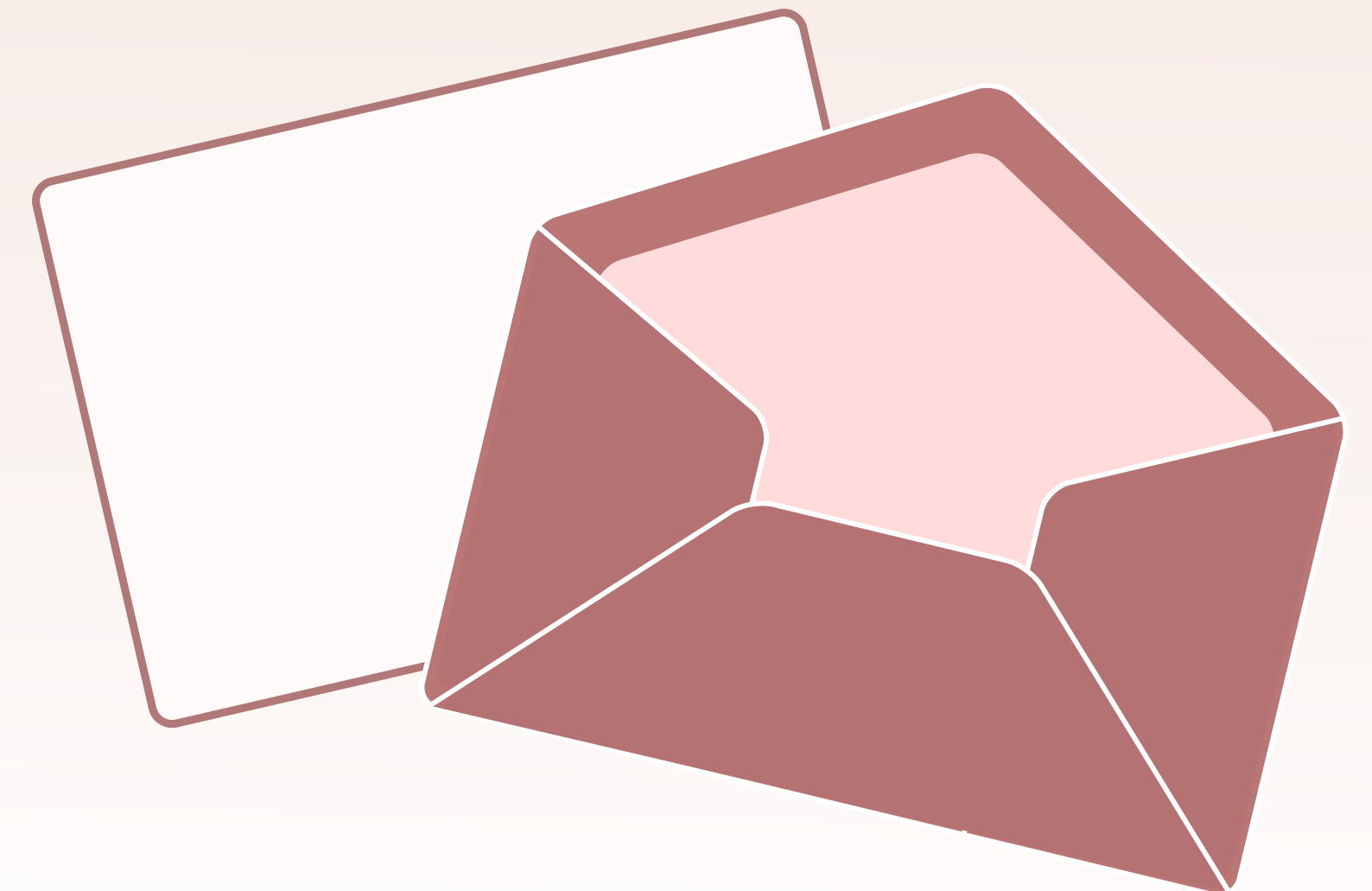
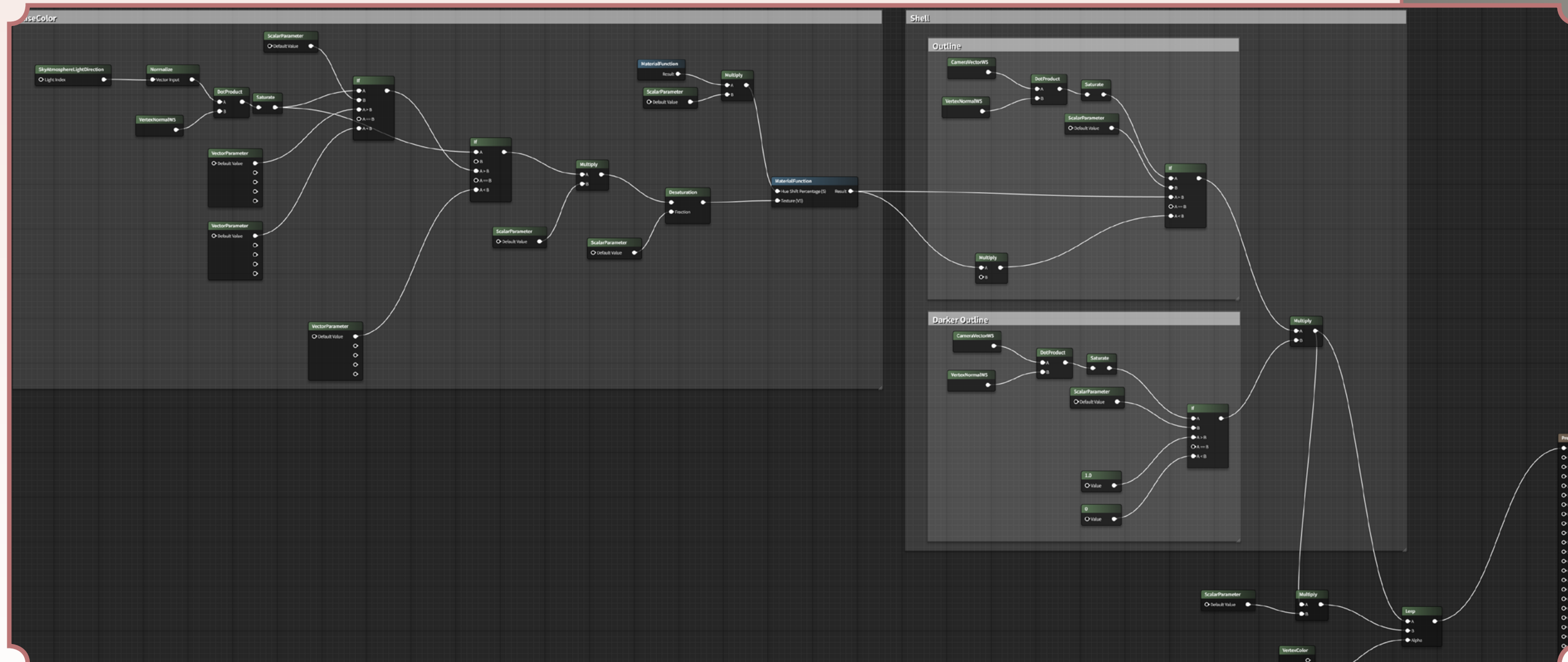


◆ Models & Materials

Bottles

The potion bottles and cauldron populate the witch's house area and should give a sense of variety without hand-authoring colors on every mesh. The goal is to have many similar bottles that automatically pick up different pastel hues.

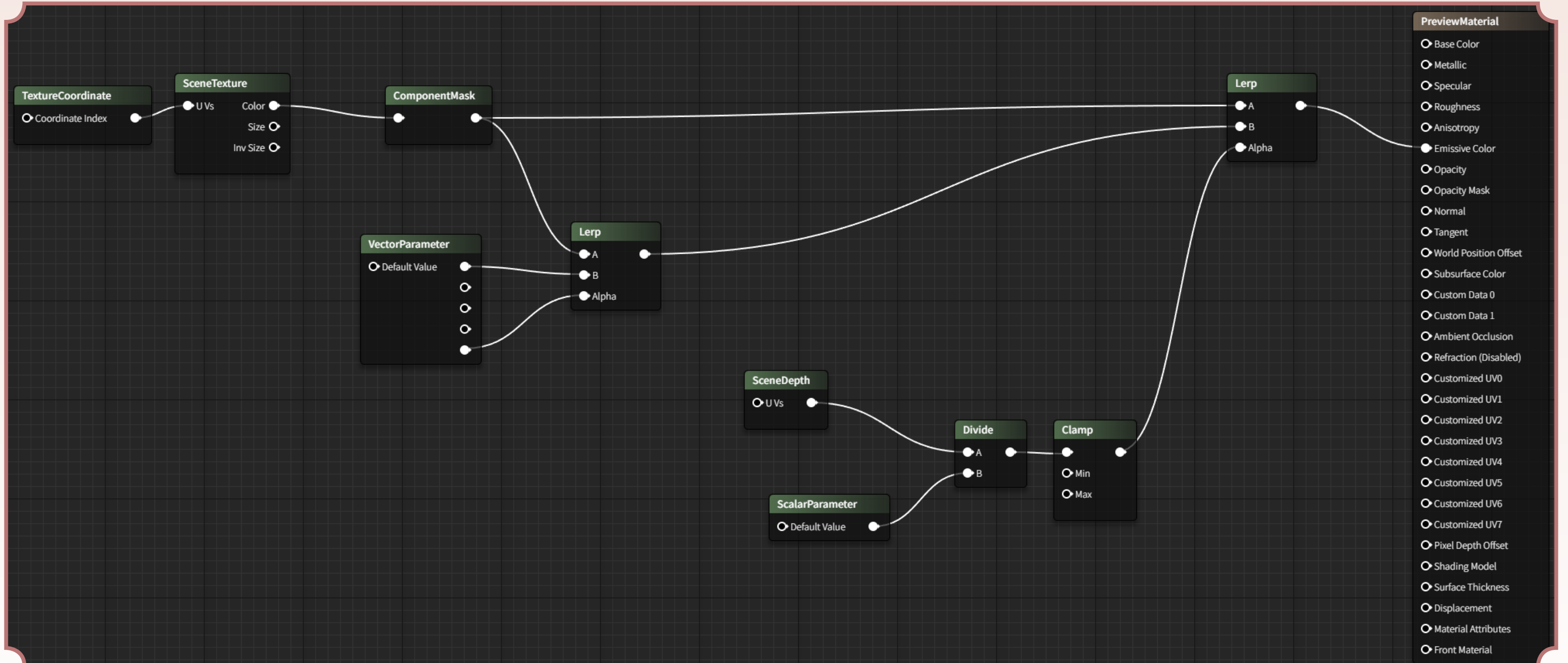
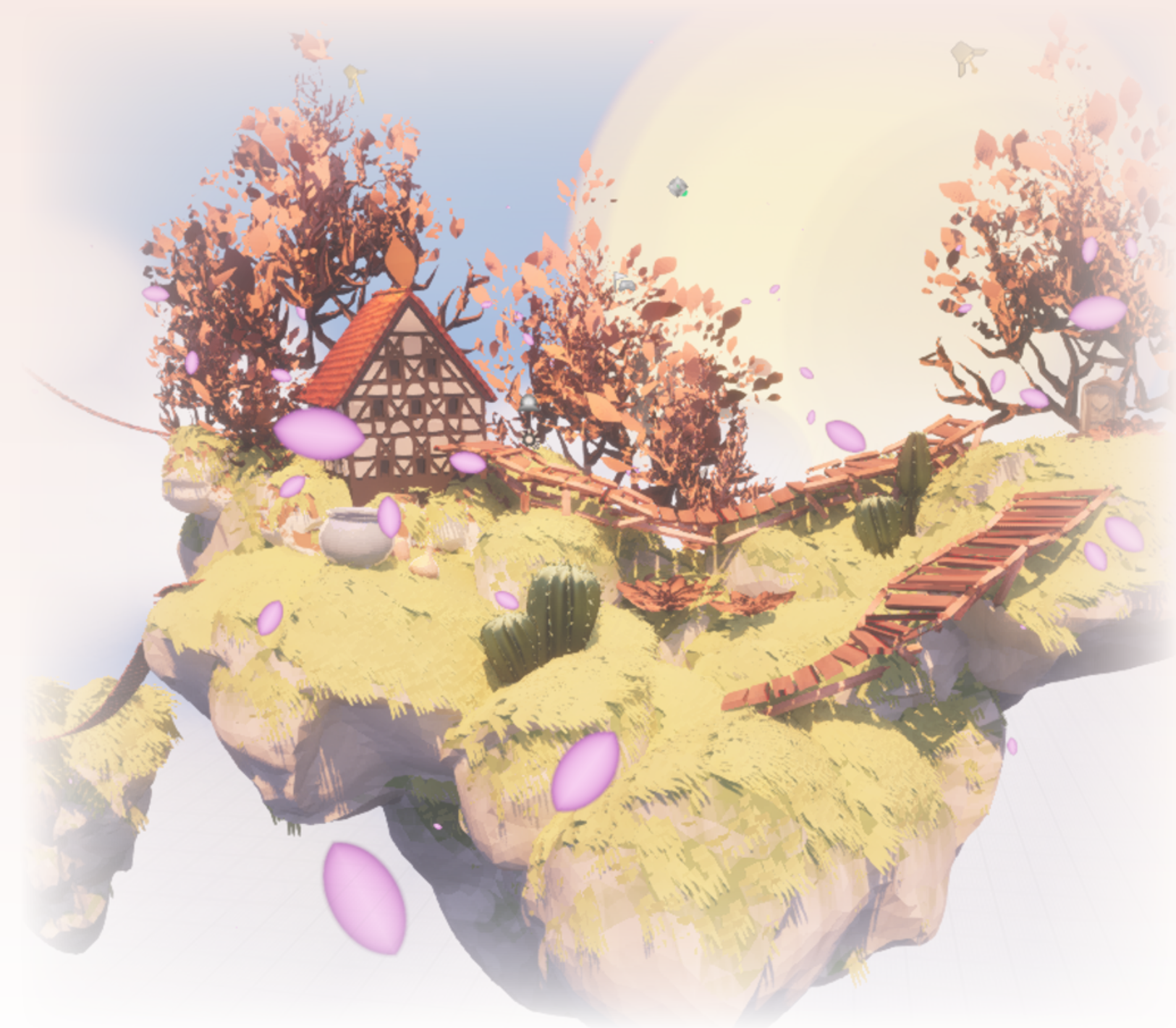
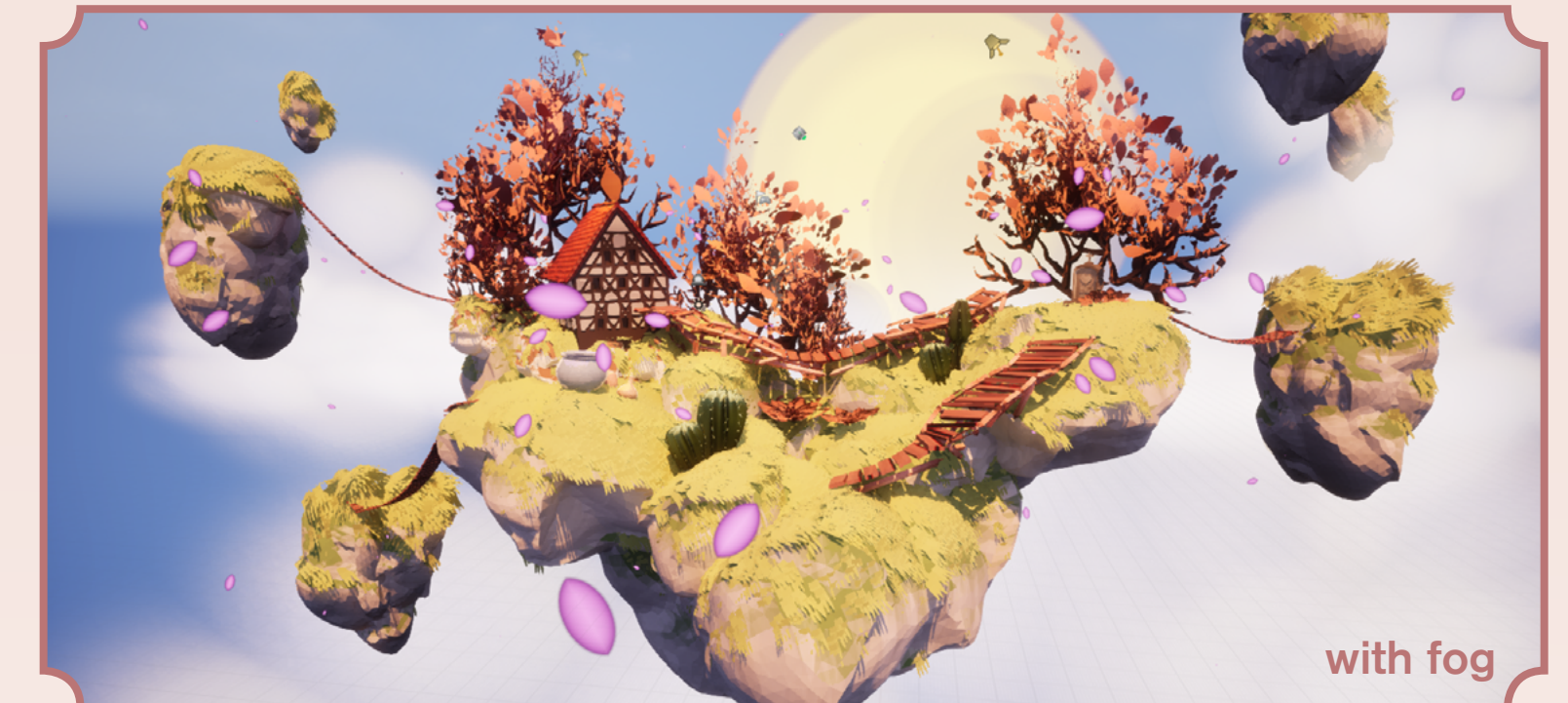
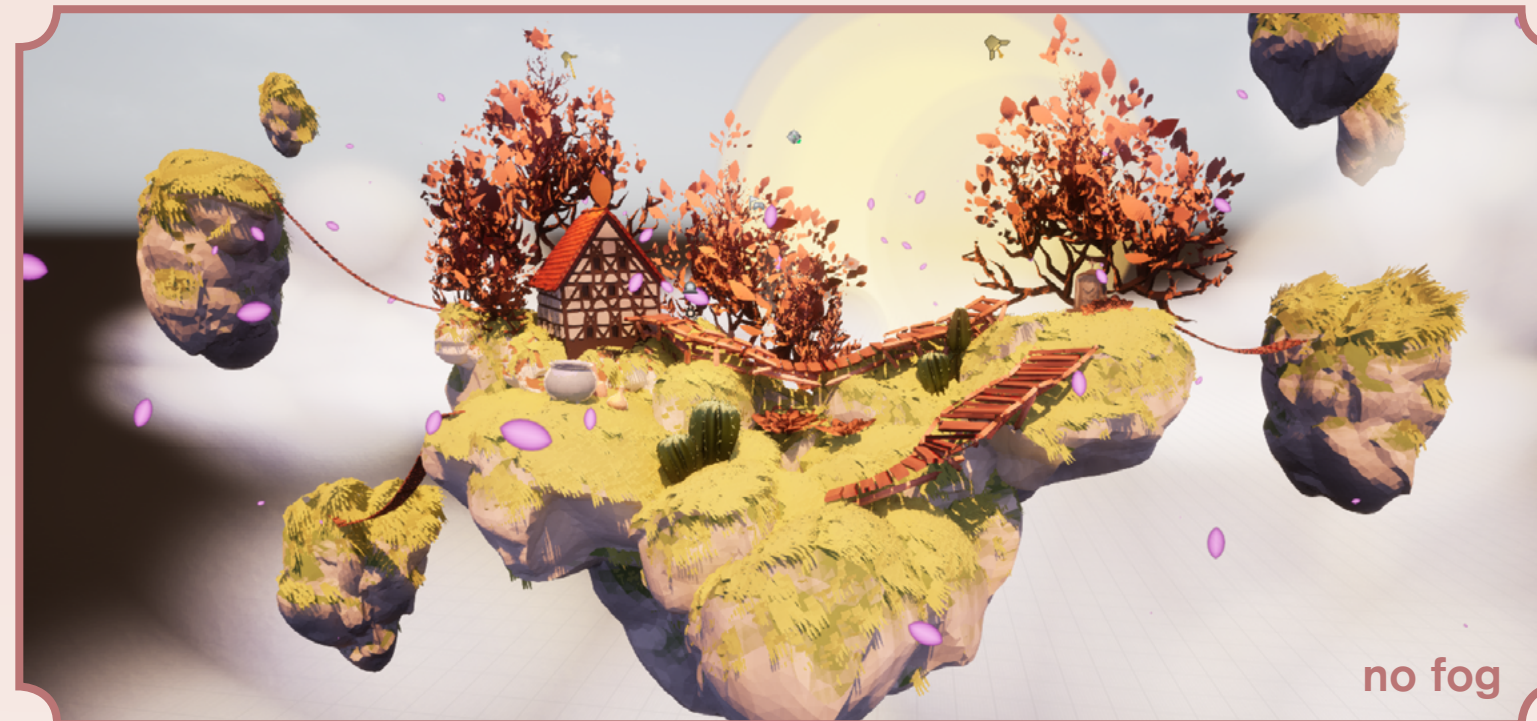
Bottle meshes are modeled in Blender and then shaded in Unreal using a glass-like material. The shader takes a base color and shifts its hue based on the object's world position, so bottles at different positions automatically receive different colors. A Fresnel or rim-lighting term is added to emphasize bottle edges, and translucency / roughness parameters keep the glass look soft and stylized.



◆ Models & Materials

Fog (post)

The fog softens the background, increases depth separation between foreground and distant islands, and pushes the atmosphere closer to the Ghibli reference—especially in the daytime version.



◆ Models & Materials

Billboard(stars/moon) (substance designer)

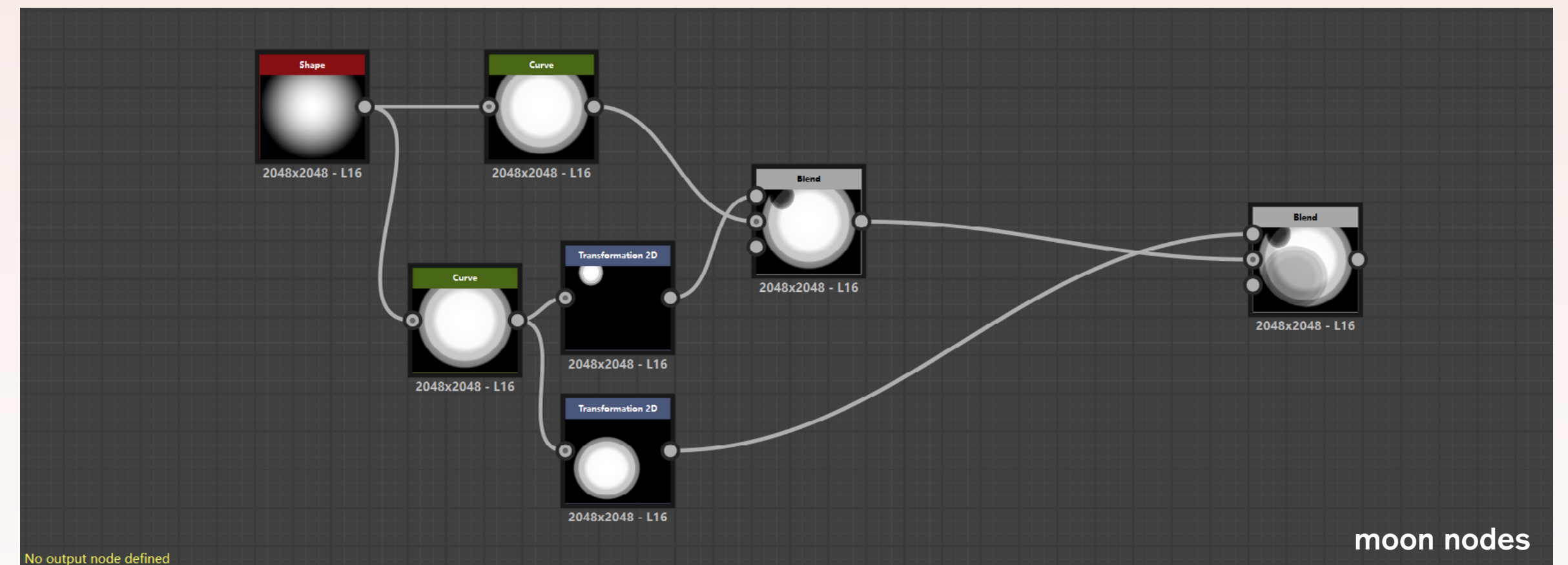
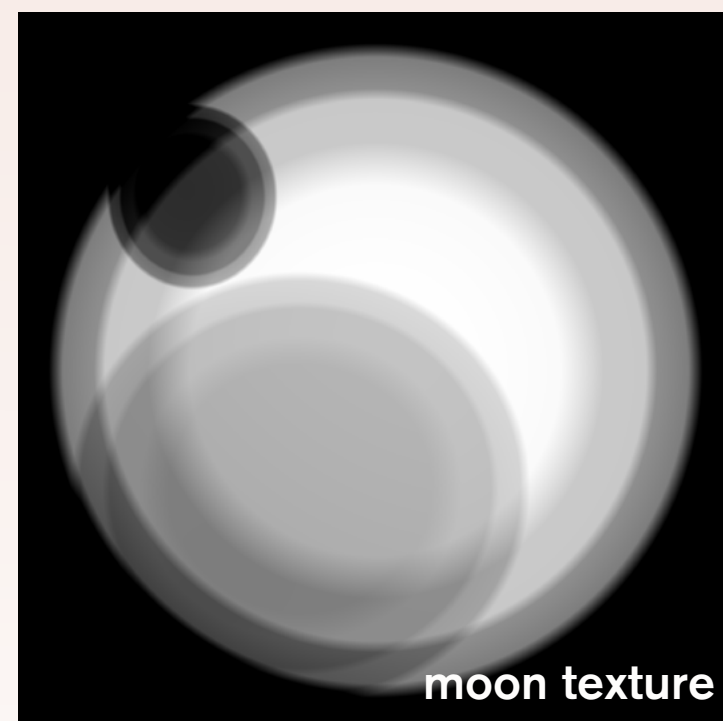
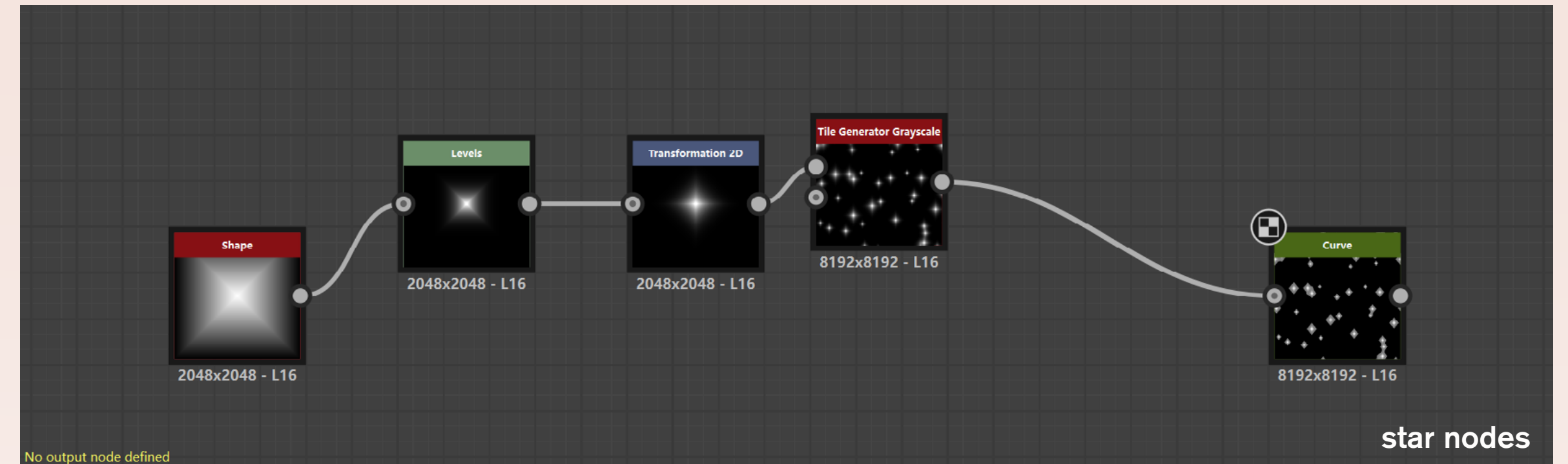
The billboard textures provide a stylized starry sky and large moon disc that sit behind the island and reinforce the dreamy, floating-world feeling.

In Substance Designer, I created separate star and moon textures.

The star texture is built from simple shape and noise nodes combined through a tile / scatter setup, then blurred and layered to get bright cores with softer halos.

The moon texture uses radial gradients and blended circular shapes to define the main disc, secondary rings, and a smaller moon, producing a clean grayscale mask.

These maps are then used in Unreal on unlit billboard materials, where color and intensity are controlled via parameters to match the current time-of-day look.



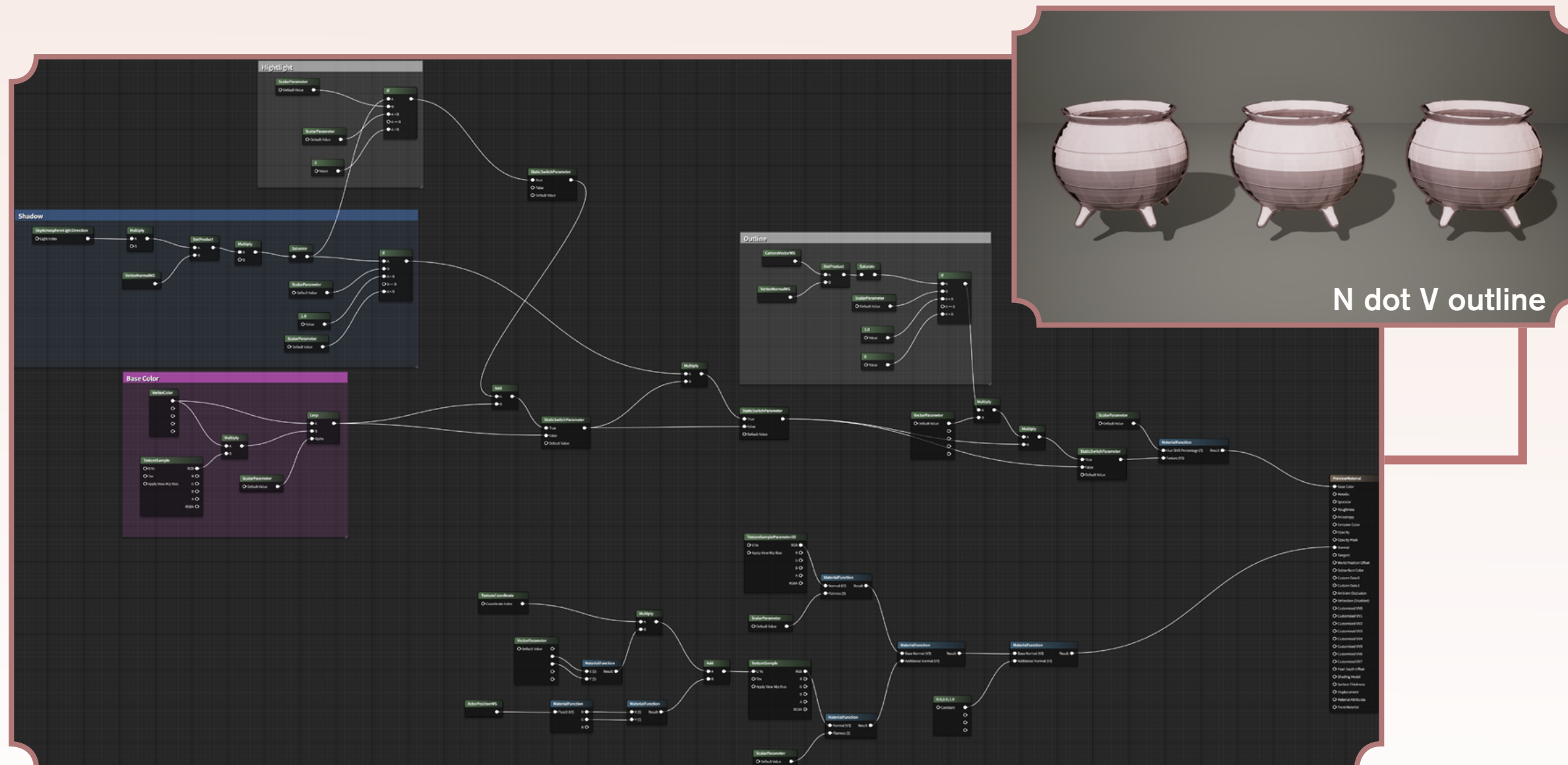
◆ Models & Materials

Outline

I wanted a stylized outline similar to the reference images, where forms are clearly separated by a soft ink line but still feel integrated with the lighting. I explored both N-dot-V and post-process approaches.

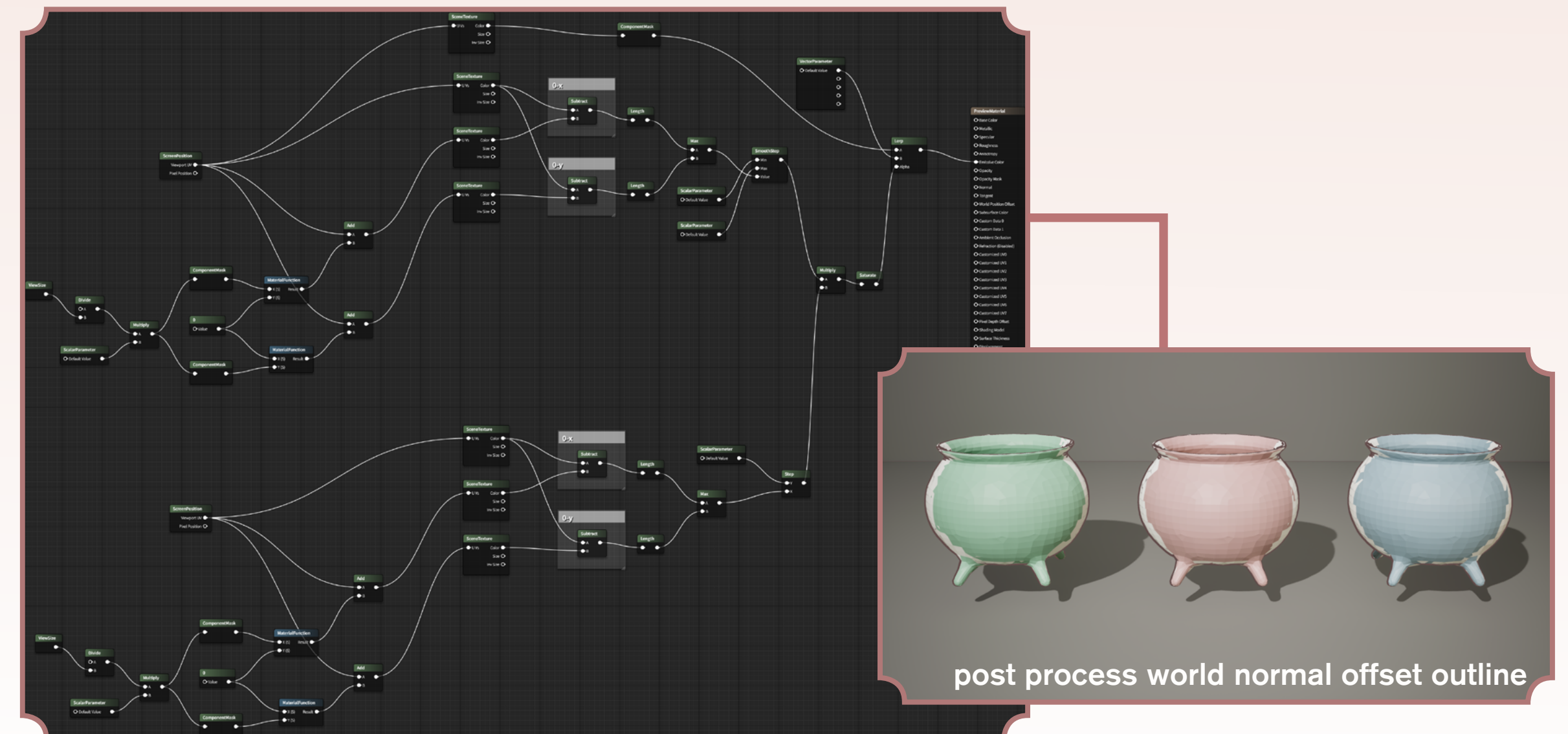
N dot V Outline

This approach computes an outline mask from the dot product between the camera vector and the vertex normal ($N \cdot V$). CameraVectorWS and VertexNormalWS are normalized, passed into DotProduct, saturated, and compared against a threshold with an If node. While this works, outline thickness varies across models and can look uneven on low-poly meshes.



World-Normal Post-Process Outline

To get more consistent lines, I use a post-process material that samples the world-normal buffer at slightly offset UVs and subtracts them to measure normal differences. The length of this difference vector, after scaling and thresholding, becomes an outline mask that is multiplied by a user-defined outline color and blended over the scene. Because it runs in screen space, this method keeps outline thickness uniform and closer to the clean, drawn look in the reference.



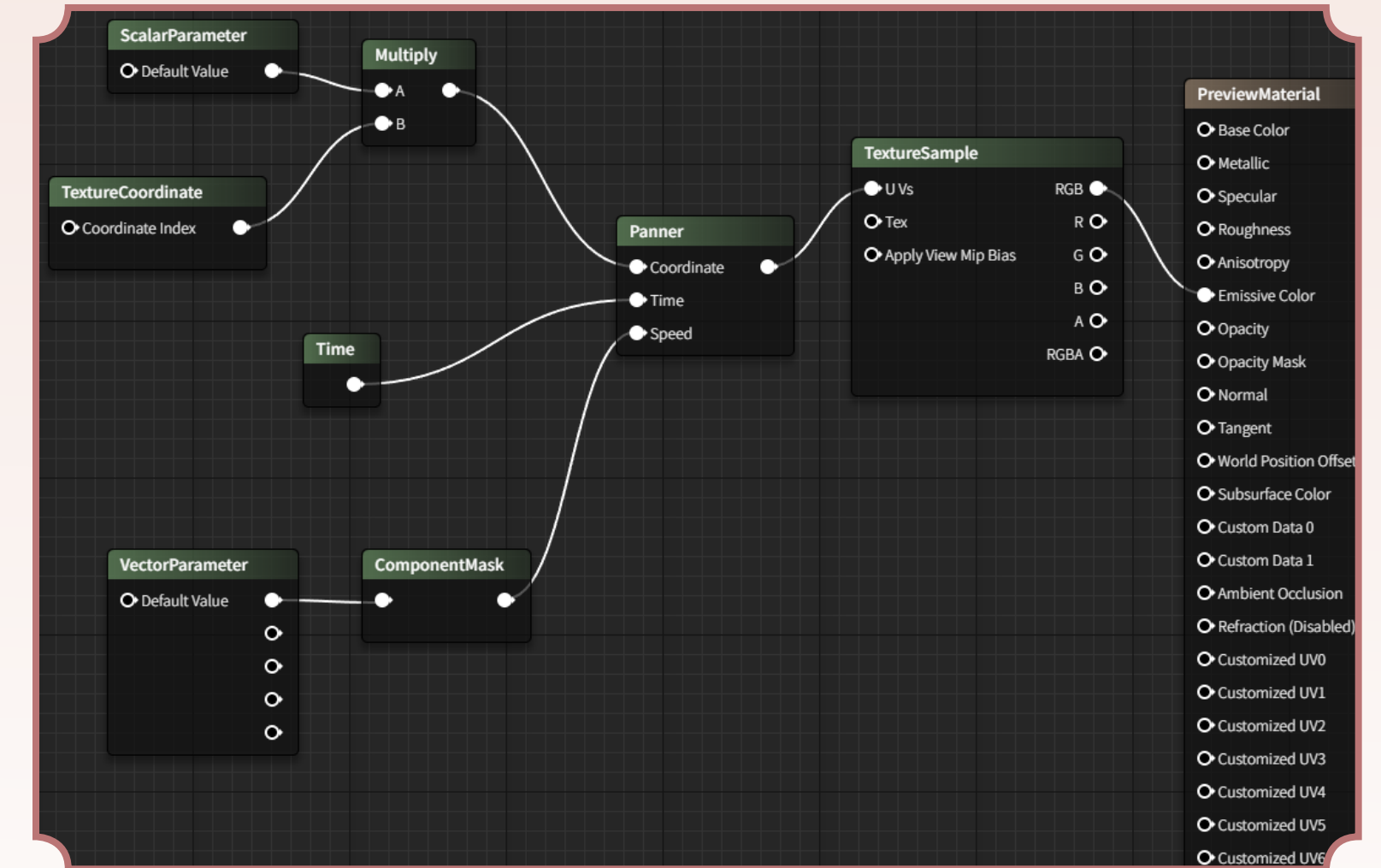
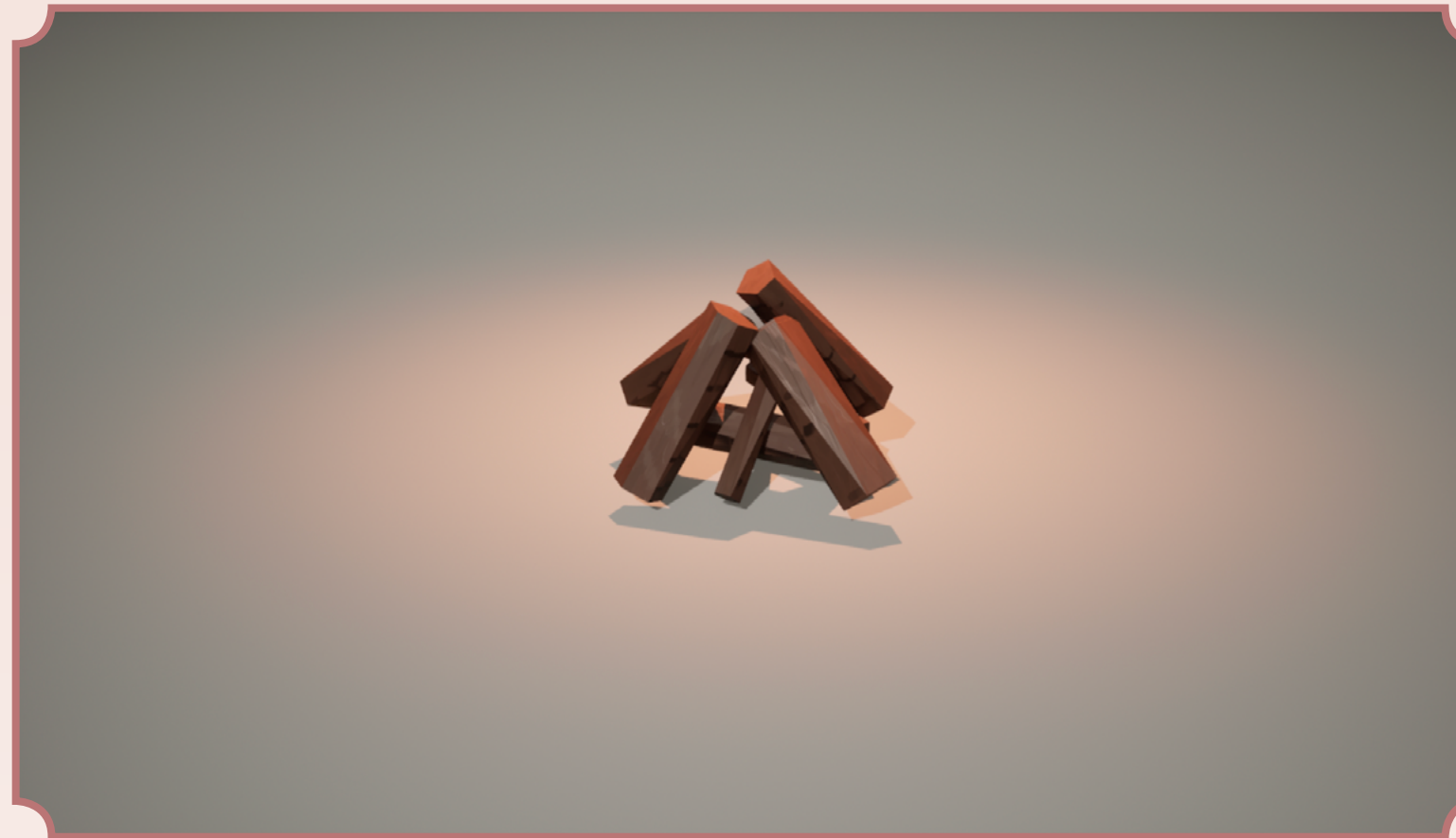
FX

Fire and Fire Effects

The campfire under the woodpile adds a secondary warm light source for the night scene and introduces subtle movement through fire particles and flickering light.

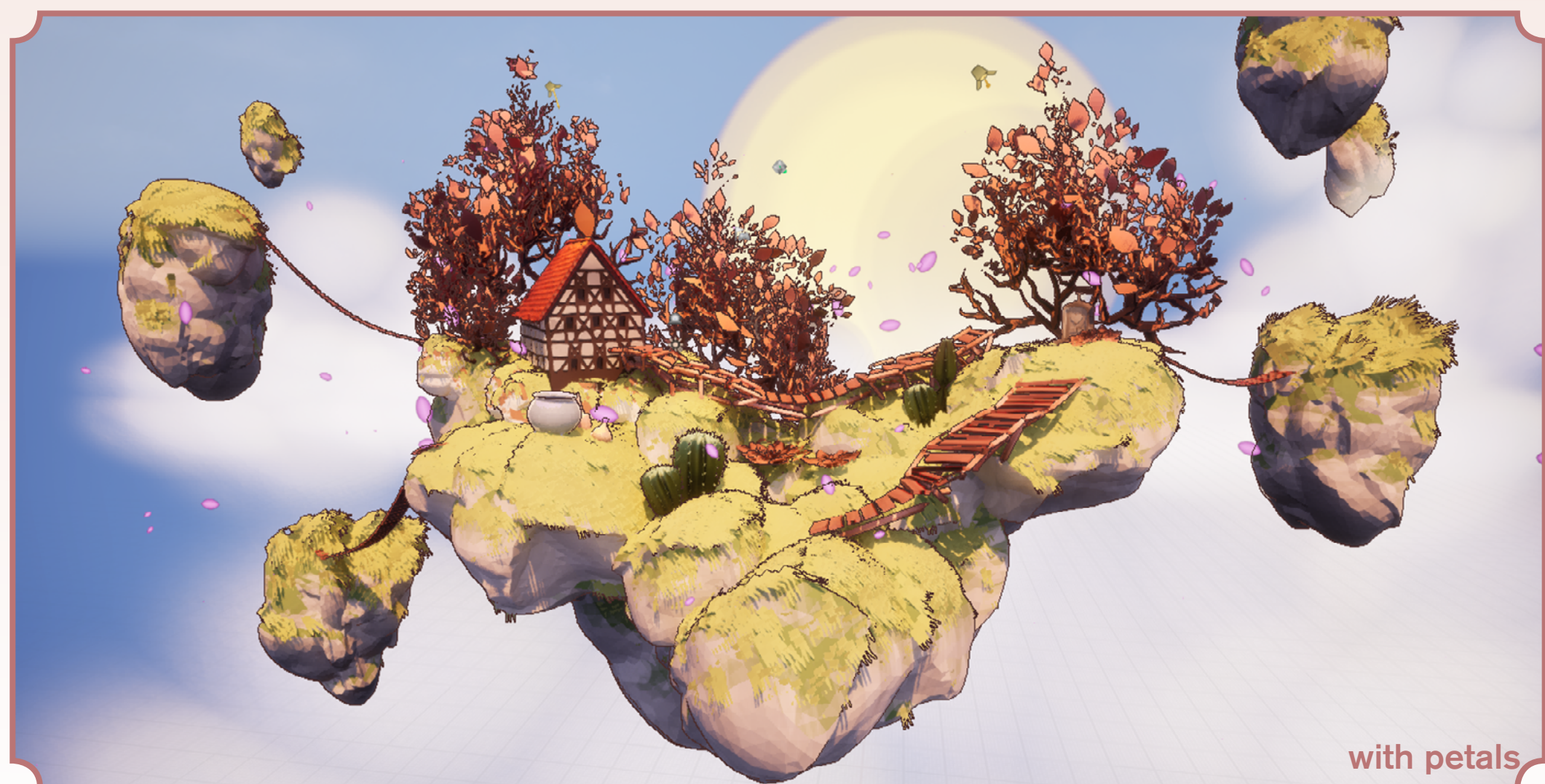
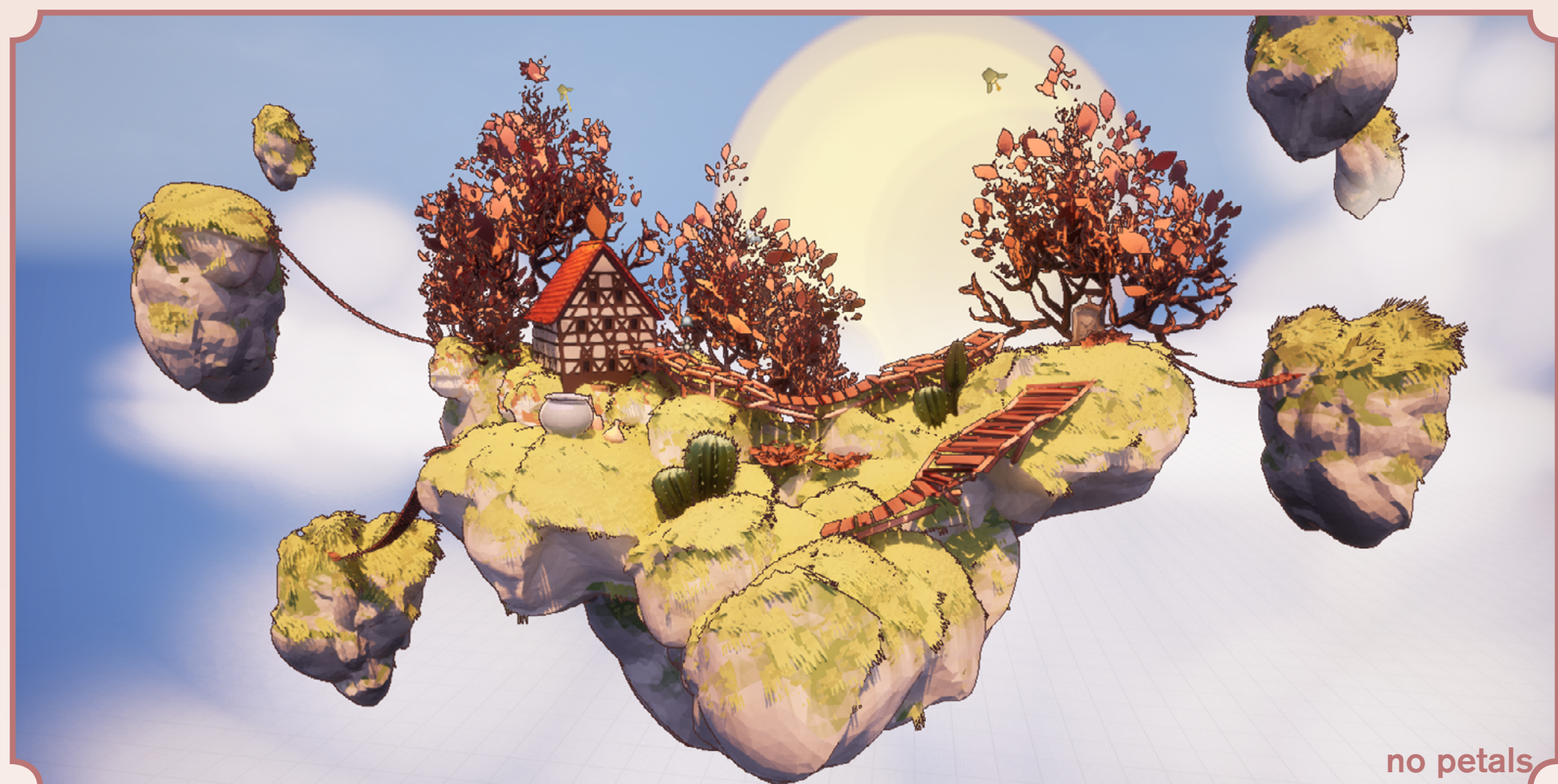
I used a Niagara system that spawns fire particles around the woodpile, using randomized size, lifetime, and velocity to create a natural flicker. The material for these particles is additive and uses a simple gradient / noise texture to shape each flame sprite over its lifetime.

To avoid a static point light, I also created a light-function material: a grayscale noise texture is panned over time using a Panner node driven by Time and a speed ScalarParameter. The sampled value modulates the light's intensity (through Emissive Color or light-function output), producing a convincing, controllable flicker without scripting.



FX

Falling Cherry Blossoms



In the daytime version the scene felt a bit empty, so I added falling cherry blossoms to bring more life.

The blossoms are implemented as a Niagara system that spawns petal meshes or cards above the island and lets them drift downward with some lateral “wind” noise. The petal material procedurally generates a soft petal shape from UVs. Combined with randomized rotation and velocity in Niagara, this creates a gentle, continuous shower of petals across the scene.

