Documentation

The package includes the following:

1) A demo scene with demo materials and models

2) PainterlyLit, PainterlyLitSimple and PainterlyLitTransparent (example Shader Graphs)

3) PainterlyNormalsSubgraph (a subgraph for your shaders)

4) Painterly.hlsl (an HLSL file containing custom functions)

The shader uses simple noise (Shader Graph node) and 3D Voronoi noise in fragment shader, which can be computationally expensive. For optimal performance, I would recommend using it only for characters, props, and other small objects.

For larger objects, consider baking the normals in Blender (or other 3D software) using the same setup (copy the subgraph logic and replace my custom function with Voronoi noise).

You can check my shader graph setups to create your own shaders with the subgraph I provided. You just need to connect the Object Space Normal to NormalOS.

The output parameters are:

a. Normal Object Space

b. Normal Tangent Space (connect this to your surface input)

c. Color (random per-normal color)

d. Color Grayscale (red channel of Color output; can be used as a random value)

e. Distance (distance to the Voronoi noise edge)

How to use the shader

Prepare the 3D Model

1) Import your model into Unity.

2) Enable "Calculate normals" and apply a smoothing angle (or pre-smooth the normals in Blender/other 3D software).

Configure the Material

1) Create a new material and assign the PainterlyLit shader.

2) BaseColor: Apply your albedo texture with smoothness in the alpha channel and adjust the Tint (BaseColor parameter). If you are using transparent shader, your albedo texture should contain transparency in alpha channel and smoothness in Metallic + AO texture alpha channel.

3) Smoothness: Adjust to remap the alpha channel's smoothness values.

4) Metallic & AO: Provide a texture with Ambient Occlusion (red channel) and Metallic (green channel). Set the Metallic parameter to 1 (if you need).

If you prefer a different texture format, you can modify the Shader Graph's surface inputs accordingly.

5) If you are using PainterlyLit or PainterlyLitTransparent you can add a PainterlyMask texture which controls the strength of effects (Normals in red channel and Color Variation in green channel).

Adjust Normal Parameters

1) Stroke Randomness: Controls Voronoi texture cell

alignment.

2) Stroke Scale: Adjusts Voronoi noise scale.

3) UseBlendingNoise: Smoothes Voronoi cells (optional).

Important:

Voronoi noise (the first two parameters under "Normals") is UV-independent, but blending noise relies on UV coordinates — ensure your mesh is unwrapped (doesn’t matter how) if using this feature.

Color Variation (Optional)

1) Set the "Color Variation" value to more than 0 for additional color adjustments.

2) Adjust the colors of variation.