HexTile: Nature Pack

Thank you for purchasing the HexTile: Nature Pack! I hope this Unity asset pack is helpful in creating stylized, high quality hexagonal-themed environments. If you need any advice, bug reports, suggestions or comments, feel free to contact me at: alexakins1@gmail.com.

This asset pack is intended to be used with Unity's Universal Render Pipeline (URP) or Unity's Standard Pipeline, Unity 2019.4 upwards. However graphical features such as the included shaders are exclusive to URP. A HDRP version is planned for a later update. No additional plugins or assets are required, however a vertex-painting plugin such as Polybrush will be needed to paint additional layers such as Moss. Polybrush can be found in Unity's Package Manager.

In addition to the many hand sculpted assets included in this pack, the asset pack also contains a number of shaders which enable effects such as secondary moss/dirt layers, animated wind sway, water, fire and more. It also contains a number of scripts developed by William Akins (w.akins111@gmail.com) that enable you to easily snap together assets along a hexagonal grid.

Included in this PDF is a guide on using the included scripts and the options they provide, along with a breakdown of the various shaders included in the pack.

Credits:

Art & Shaders: **Alex Akins**

Website: https://alexakins.art/
Artstation Portfolio: https://www.artstation.com/alexakins
Email: alexakins1@gmail.com

Scripting: **William Akins**

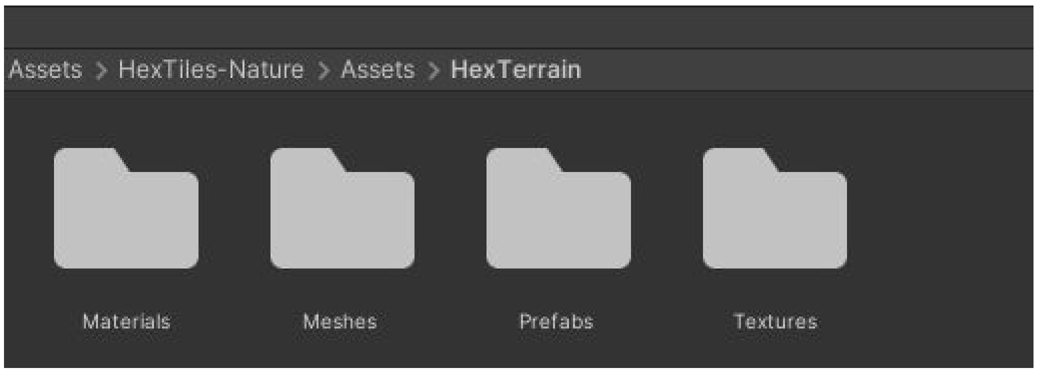
Email: w.akins111@gmail.com

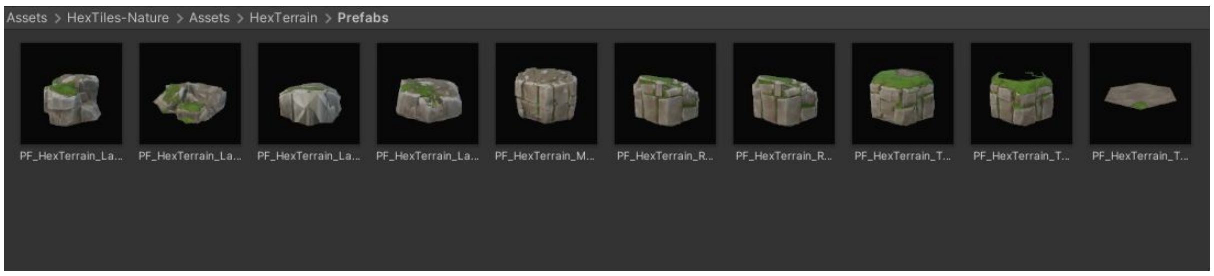
Table of Contents

Assets

To make use of the assets included in this pack, use the prefabs stored in *.../HexTiles-Nature/Assets/("AssetType")/Prefabs*

Materials, textures, and raw meshes for each asset can be found in the same root folder:





HexTerrain Prefabs folder

Using HexTile: Nature Pack in Unity's Standard Pipeline

Although this asset pack was built to take advantage of the new URP pipeline offered in Unity, and features such as Shader Graph which are exclusive to the new scriptable render pipelines, Standard versions of the assets can be found in the *Assets_Standard* folder, with the prefabs using Unity's standard shader instead of the custom URP shaders included in this pack.

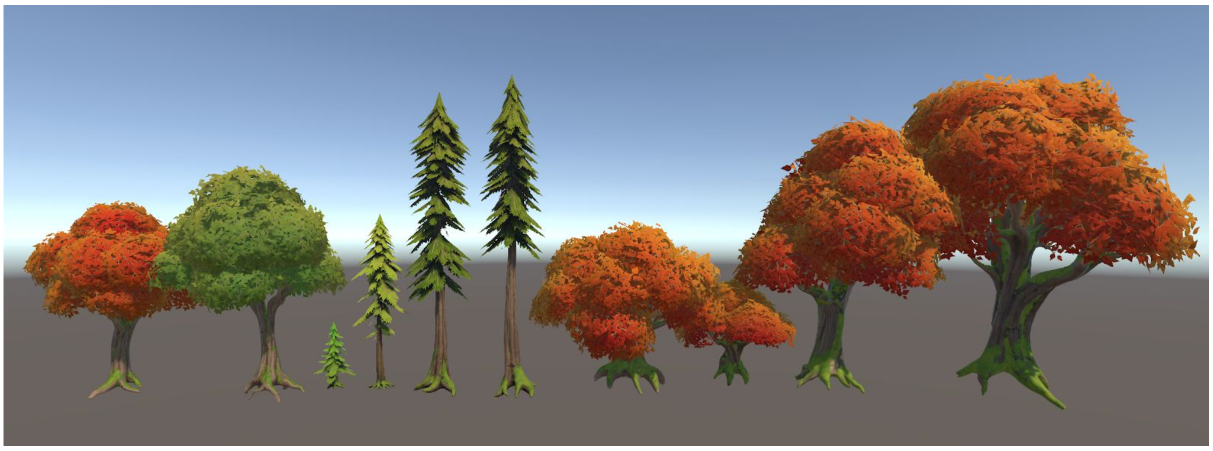
For space reasons textures and raw meshes are shared between the two asset versions. For organization purposes you may wish to install the asset pack in its entirety, then replace the *Prefabs* and *Materials* folders and their contents within each specific asset folder (*HexTerrain*, *HexGrass* etc) with the equivalent files from Asset_Standard.

Note that if the asset pack is used in Unity's Standard Pipeline, only the basic Unity Standard Shader will be able to be used, so extra features such as the moss layers or wind animation will not be available and other shaders such as the water and fire shaders won't work either.

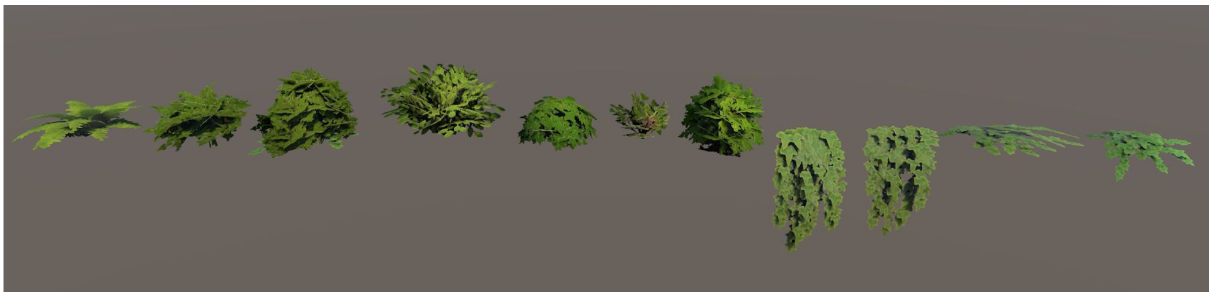
A version of the demo demo asset scene is available for Standard pipelines under *DemoScene_Assets_Standard.* A version of the regular demo level is planned for a later update.

Asset List

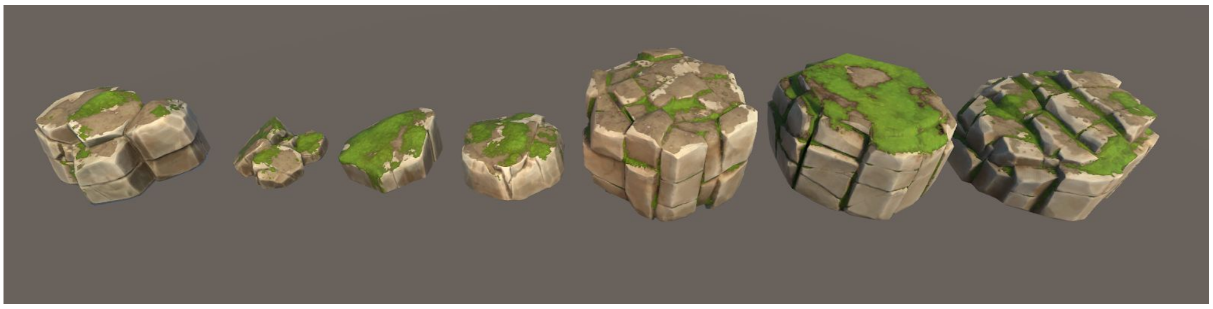
The following is a list of assets included in this pack, sorted by group, listing the directory where they be found:



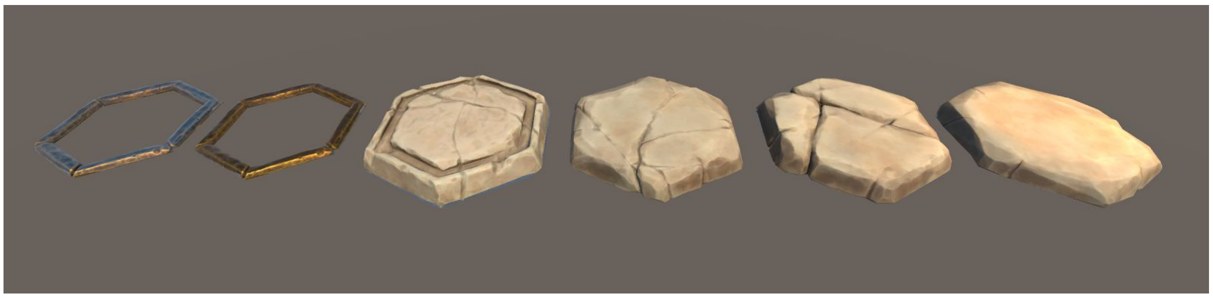
Trees (Regular, Conifer, Tall) **.../HexTiles-Nature/Assets/Trees/Prefabs**



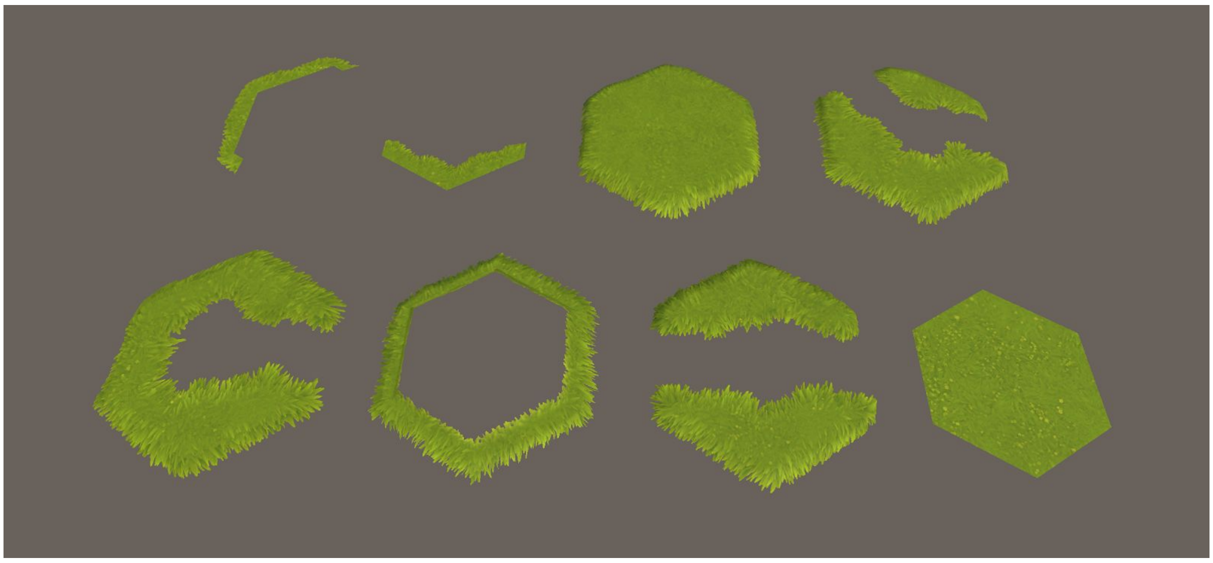
Foliage (Ferns, Bushes, Ivy) **.../HexTiles-Nature/Assets/Foliage/Prefabs**

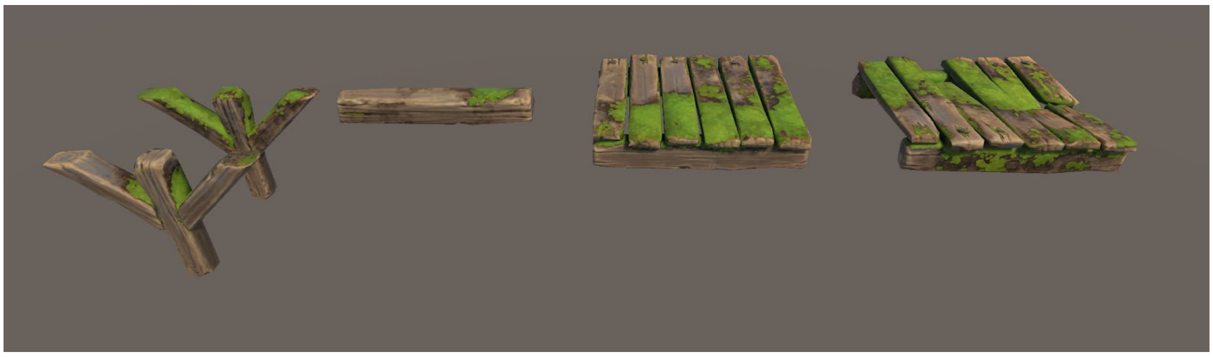


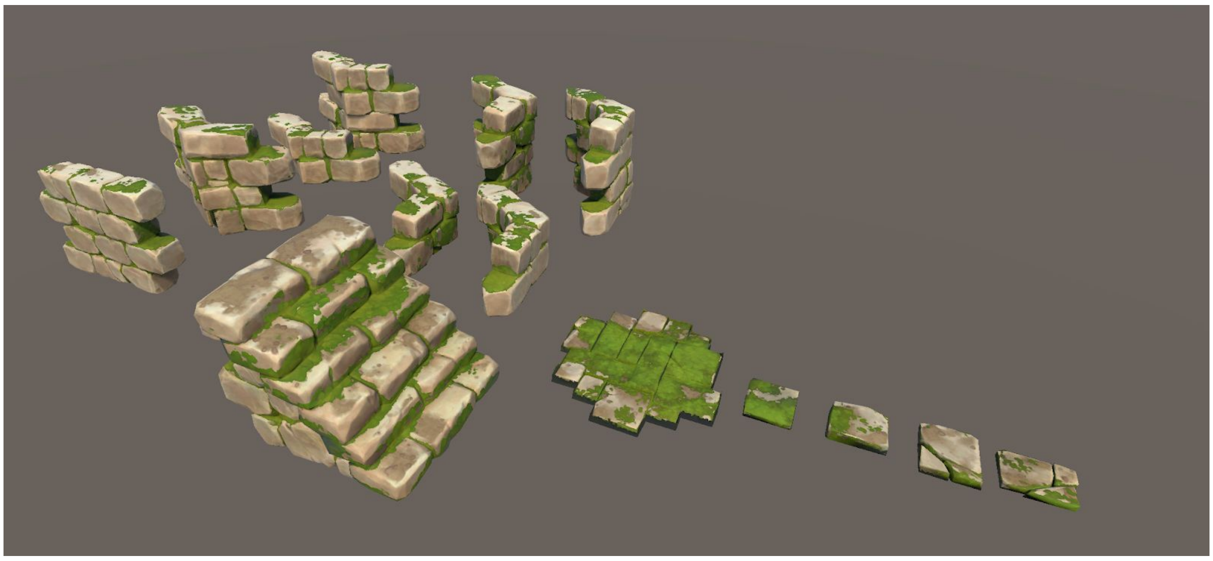
Hex Terrain (Large Rocks, Cliff Pieces, Ramp)
.../HexTiles-Nature/Assets/HexTerrain/Prefabs

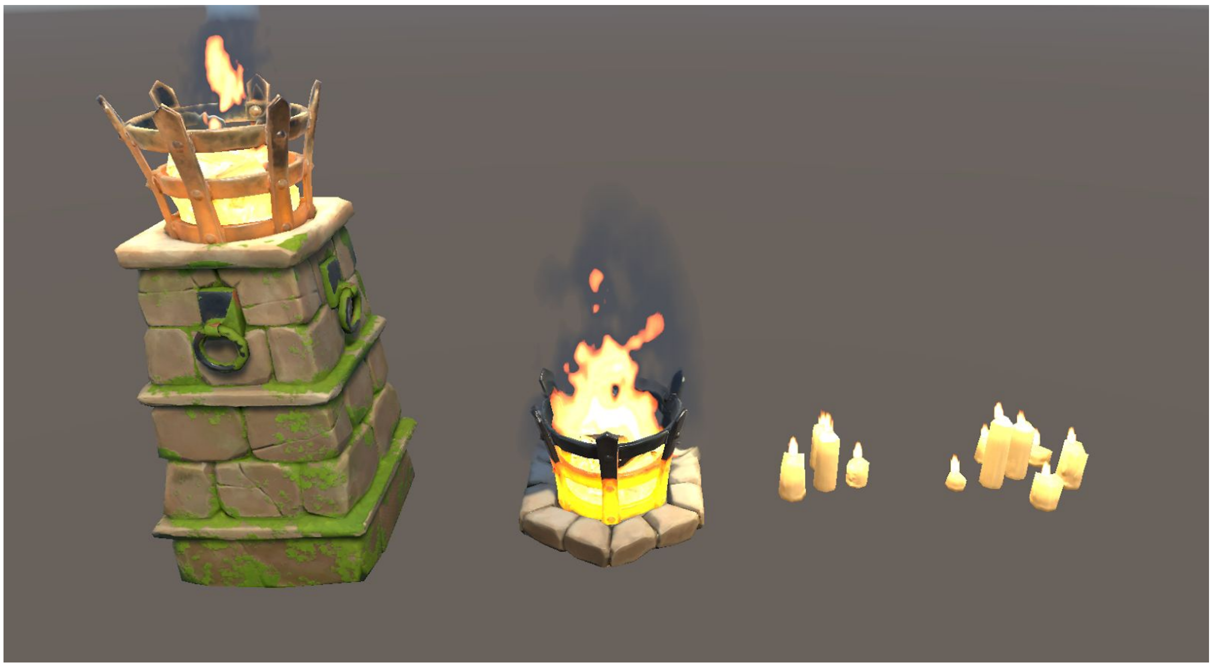


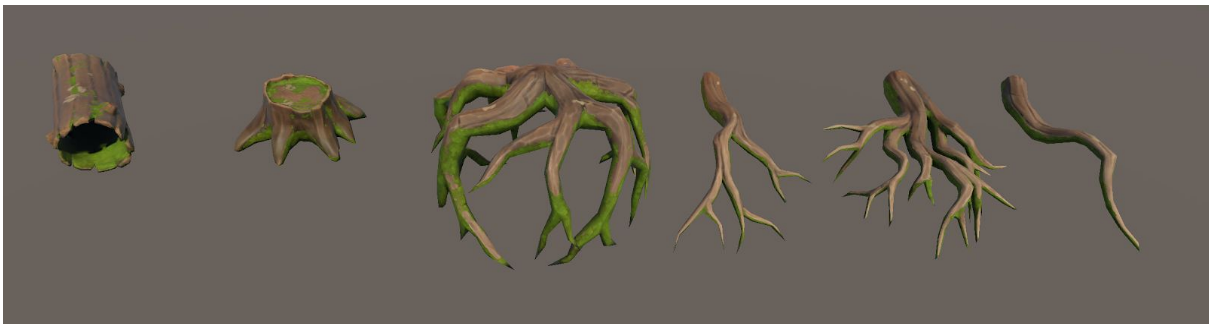
Hex Board & Metal Border **.../HexTiles-Nature/Assets/HexBoard/Prefabs**

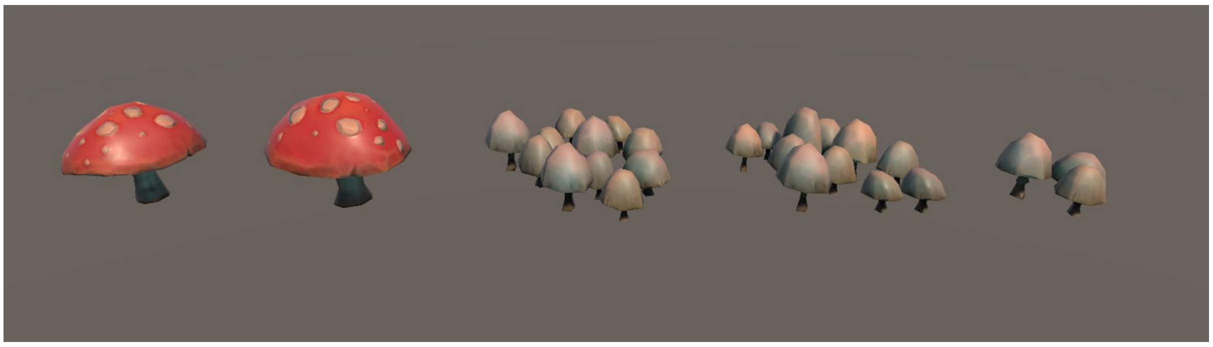

Hex Grass Sections **.../HexTiles-Nature/Assets/HexGrass/Prefabs**


Wooden Bridge Sections **.../HexTiles-Nature/Assets/WoodenBridge/Prefabs**

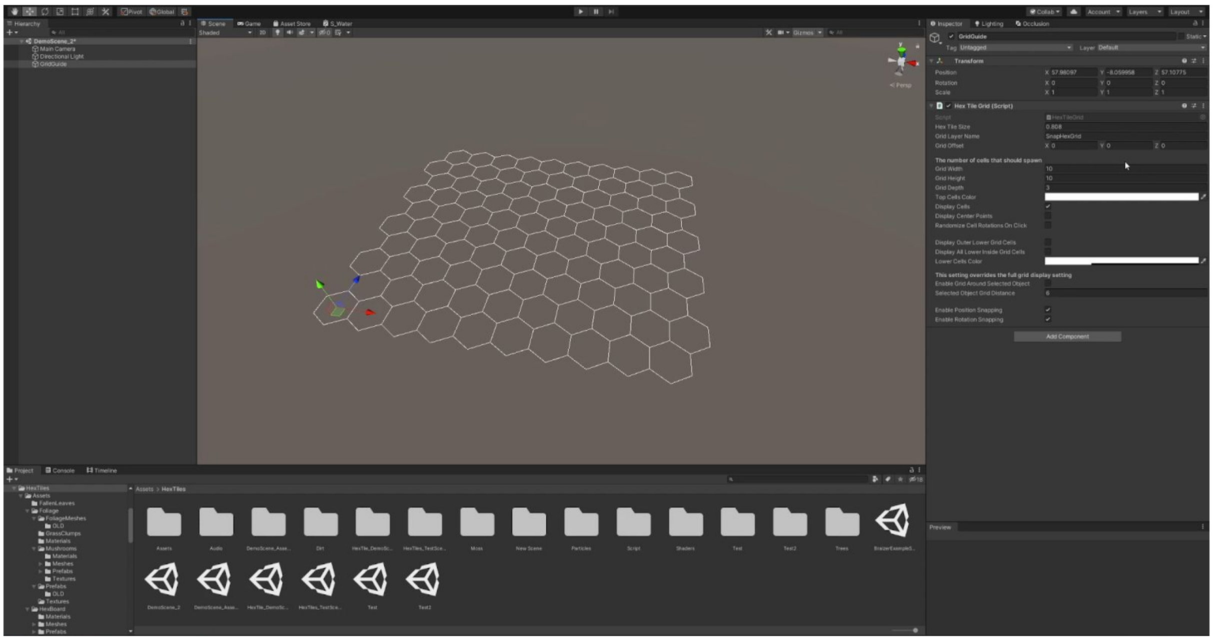

Stone Ruins **.../HexTiles-Nature/Assets/StoneRuins/StoneRuins**


Lighting Props (Braizer, Candles)
.../HexTiles-Nature/Assets/LightingProps/(Brazier/Candles)/Prefabs


Roots and misc tree assets **.../HexTiles-Nature/Assets/Trees/Prefabs/Roots**
(The fallen tree trunk and tree stump can be found in the regular trees folder):
.../HexTiles-Nature/Assets/Trees/Prefabs


Mushrooms: **.../HexTiles-Nature/Assets/Mushrooms/Prefabs**

Hex Tile Grid Script



```
GridGuide                          Static
Tag Untagged          Layer Default
Transform
Position        X 57.98097   Y -8.059958   Z 57.10775
Rotation        X 0          Y 0           Z 0
Scale           X 1          Y 1           Z 1
Hex Tile Grid (Script)
Script                            HexTileGrid
Hex Tile Size                     0.808
Grid Layer Name                   SnapHexGrid
Grid Offset     X 0          Y 0           Z 0
The number of cells that should spawn
Grid Width                        10
Grid Height                       10
Grid Depth                        3
Top Cells Color
Display Cells                     ✓
Display Center Points
Randomize Cell Rotations On Click

Display Outer Lower Grid Cells
Display All Lower Inside Grid Cells
Lower Cells Color
This setting overrides the full grid display setting
Enable Grid Around Selected Object
Selected Object Grid Distance     6

Enable Position Snapping          ✓
Enable Rotation Snapping          ✓
Add Component
```

The HexTile: Nature Pack comes with a hexagonal-grid snapping script to make it easy to snap environment assets together in editor to speed up level creation.

To set up the script, create a blank game object in the scene of your choice and attach the “Hex Tile Grid” script to it. You should see a hexagon grid appear in the center of the scene when Gizmos are enabled in the viewport.

The script relies on Unity’s layer system to determine which assets should be snappable. Prefab assets affected by the grid system should be on the “SnapHexGrid” layer. Because Unity stores layer information within the project settings rather then within scenes or assets, the Hex Tile Grid script will automatically create this layer when attached to a gameobject, if such a layer doesn’t already exist, as well as search for a list of specific assets included in the package which are intended to be snapped and automatically attach the layer to them.

The script comes with a number of customisation options:

<u>Setup options: (Should usually left as default)</u>

> **Hex Tile Size**: Size of individual hex cells, defaults to 0.808. Shouldn’t be modified if the pre-built assets are used otherwise the assets will no longer seamlessly snap together (unless such a thing is intended).
>
> **Grid Layer Name**: Name of layer used to identify snappable assets, default SnapHexGrid. Probably shouldn’t be changed unless required.
>
> **Grid Offset**: Offset grid in world space. Should only be done on startup otherwise snappable assets will likely become misaligned when updated.

Grid Options

Grid Width/Height/Depth: Change grid size along X/Y/Z axis. Higher values potentially lower performance.

Top Cells Color: Change color of grid overlay when visible.

Display Cells: Grid overlay visibility

Display Center Point: Center point of grid cell, Used for debug purposes.

Randomize Cell Rotations On Click: When enabled, the selected asset will randomly rotate in 60deg increments. Useful to add variation when using terrain assets, but remember to uncheck once finished to prevent unwanted rotations (Especially when using baked lighting!)

Display Outer Lower Grid Cells: Make the border cells for each grid layer visible, useful for seeing depth of grid without enabling visibility of all layers.

Display All Lower Inside Grid Cells: Display all grid cells for each layer. Can be performance intensive depending on size and depth of grid.

Lower Cells Color: Color of lower grid cell overlay

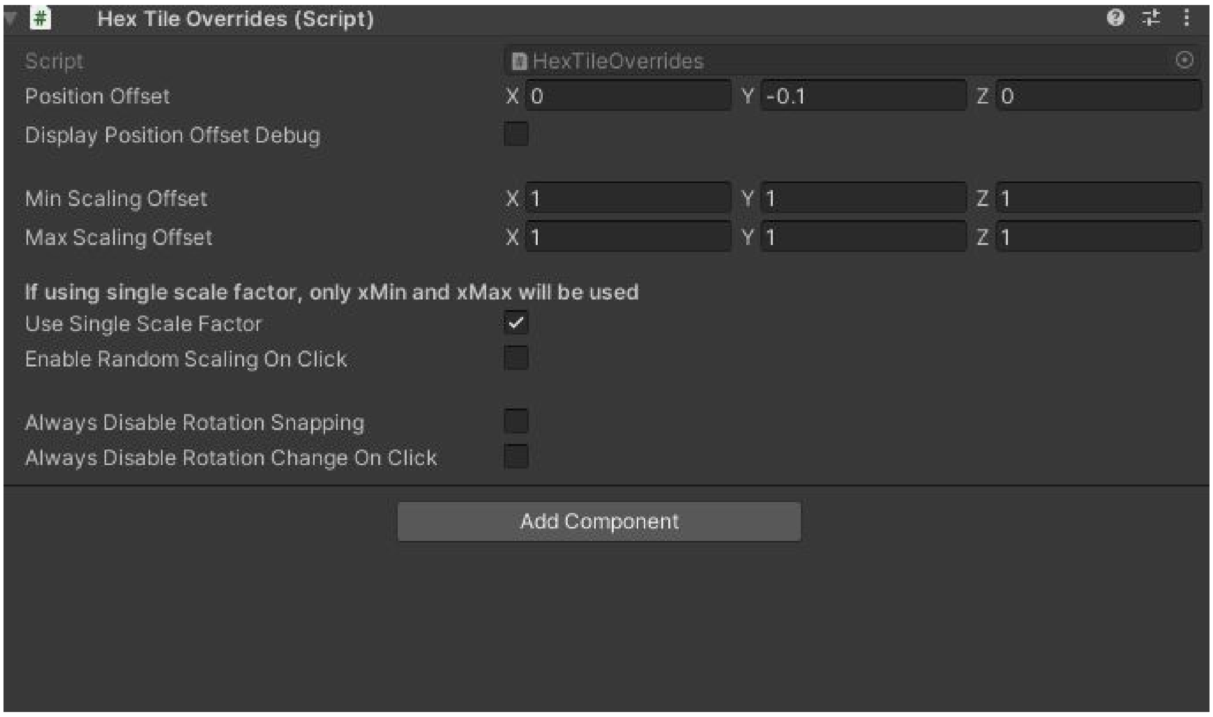
Enable Grid Around Selected Object: Creates a grid overlay around the selected object as an alternative to showing the entire scene grid.

Selected Object Grid Distance: Modify size of selected object grid overlay mentioned above.

Enable Position Snapping: Disable grid snapping entirely for position.

Enable Rotation Snapping: Disable rotation snapping between 60 degree increments.

Hex Tile Overrides Script



Hex Tile Overrides (Script)

Script	HexTileOverrides
Position Offset	X 0 Y -0.1 Z 0
Display Position Offset Debug	☐
Min Scaling Offset	X 1 Y 1 Z 1
Max Scaling Offset	X 1 Y 1 Z 1

If using single scale factor, only xMin and xMax will be used

Use Single Scale Factor	✓
Enable Random Scaling On Click	☐
Always Disable Rotation Snapping	☐
Always Disable Rotation Change On Click	☐

Alongside the Hex Tile Grid Script, an optional override script is available for individual prefabs to modify snapping on a per-object basis:

Position Offset: Offset object along X/Y/Z axis, displacing them from the grid. Useful for positioning objects in a certain location within the grid cell or to avoid unwanted intersections.

Display Position Offset Debug: Debug line showing offset amount from grid cell center.

Enable Random Scaling On Click: Similar to the **Randomize Cell Rotations On Click** option in the Hex Tile Grid script, this option randomizes the scaling of the selected asset on click - useful for certain assets such as trees if they are part of the snapping layer.

Min/Max Scaling Offset: If the randomized scaling is enabled, the min/max scaling factor.

Use Single Factor: Uniform randomized scaling.

Always Disable Rotation Snapping: Disable rotation snapping for this specific object.

Disable Rotation Change On Click: Overrides randomize rotation option for this asset, if **Randomize Cell Rotations On Click** is enabled in the Hex Tile Grid script.

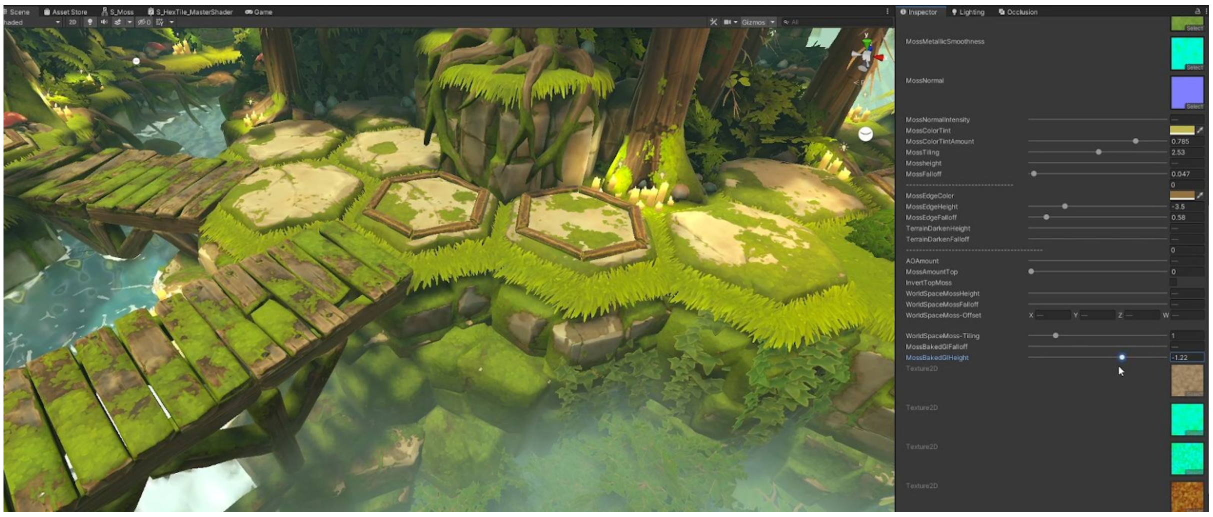
Custom Shaders

The HexTile: Nature Pack comes with a number of different shaders with complex effects such as moss growth, animated wind, water and fire. These shaders are created using Shader Graph, a node-based shader builder included in Unity URP and HDRP pipelines. As mentioned above, the shaders do not support Unity standard since the Shader Graph is unsupported on that pipeline, but versions of the assets using regular Unity standard shaders are available.

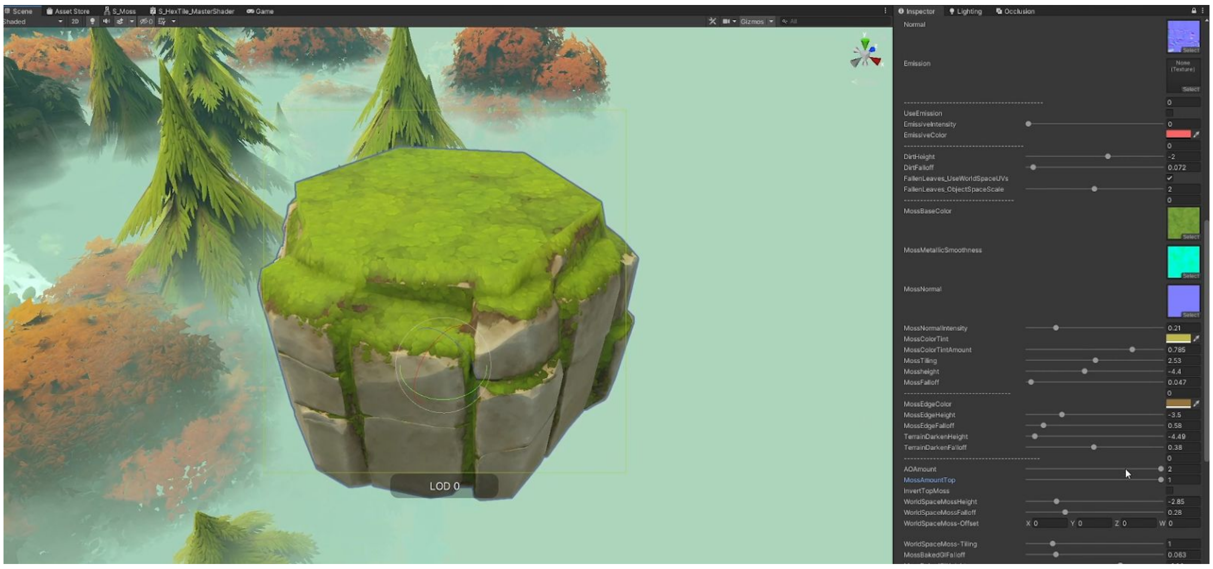
S_HexTile_MasterShader

(Used for Terrain, Hexboard pieces, Stone Ruins, Wooden Bridge, and & static tree assets (roots, fallen trees etc).

Primary shader used for most static objects. Features optional moss and dirt layers, the moss layer in particular “grows” within the cavities and above-facing sections of the asset, using the included AO maps and world-space normals. With baked lighting enabled, the moss will also grow within areas of the environment that are in shadow by using the baked lightmap texture as a mask. These features are designed to create a more natural, detailed environment as well as to break up repetition across instanced meshes.



MossMetallicSmoothness	
MossNormal	
MossNormalIntensity	—
MossColorTint	
MossColorTintAmount	0.785
MossTiling	2.53
Mossheight	—
MossFalloff	0.047
------------------------------	0
MossEdgeColor	
MossEdgeHeight	-3.5
MossEdgeFalloff	0.58
TerrainDarkenHeight	—
TerrainDarkenFalloff	—
--	0
AOAmount	—
MossAmountTop	0
InvertTopMoss	
WorldSpaceMossHeight	—
WorldSpaceMossFalloff	—
WorldSpaceMoss-Offset	X — Y — Z — W —
WorldSpaceMoss-Tiling	1
MossBakedGIFalloff	—
MossBakedGIHeight	-1.22
Texture2D	
Texture2D	
Texture2D	
Texture2D	



Normal	Select
Emission	None (Texture) Select
--	0
UseEmission	
EmissiveIntensity	0
EmissiveColor	🟥
------------------------------------	0
DirtHeight	-2
DirtFalloff	0.072
FallenLeaves_UseWorldSpaceUVs	✓
FallenLeaves_ObjectSpaceScale	2
---------------------------------	0
MossBaseColor	Select
MossMetallicSmoothness	Select
MossNormal	Select
MossNormalIntensity	0.21
MossColorTint	🟨
MossColorTintAmount	0.785
MossTiling	2.53
Mossheight	-4.4
MossFalloff	0.047
---------------------------------	0
MossEdgeColor	🟫
MossEdgeHeight	-3.5
MossEdgeFalloff	0.58
TerrainDarkenHeight	-4.49
TerrainDarkenFalloff	0.38
--	0
AOAmount	2
MossAmountTop	1
InvertTopMoss	
WorldSpaceMossHeight	-2.85
WorldSpaceMossFalloff	0.28
WorldSpaceMoss-Offset	X 0 Y 0 Z 0 W 0
WorldSpaceMoss-Tiling	1
MossBakedGIFalloff	0.063

<u>Shader Properties:</u>

ViewVertexColors: Disables textures and shows raw vertex color values. Useful for seeing where vertex paint has been applied and for debug purposes.

InvertVertexColors: Inverts usage of vertex colors so that assets with fully white colors have no vertex painted materials applied to them, and vice-versa. Useful for 3rd party assets where the mesh defaults to white rather than black vertex colors, as is a common default in 3D modelling packages. Alternatively, the meshes can be fixed by flooding the vertexes black using a plugin such as Polypaint.

BaseColor
MetallicSmoothness: (R) is Metallic, (G) is AO, (A) is Smoothness
Normal
Emission: RGB texture.

Unpacked Metallic/AO/Smoothness Maps: Enable if the texture maps for the asset are separate texture files rather than packed into the RGBA channels of a single MetallicSmoothness texture. Apply maps to the texture slots below this property.

No AO Map: Enable if the asset doesn't use an AO map. Important for the moss parameters as the shader will default to black (and therefore full moss coverage) if no AO map is provided otherwise.

EnableBaseColor_Tint
BaseColor_Tint
Tweak material colors

MetalnessIntensity
SmoothnessAmount
Tweak metallic/smoothness values.

UseEmission
EmissiveIntensity
EmissiveColor: Tint emissive value

<u>The following parameters are available when ***EnableDirtLayer*** is checked:</u>

DirtHeight: Increase visibility of dirt layer applied to the top of asset
DirtFalloff: Sharpness of dirt layer edge

FallenLeaves UseWorldSpaceUVs: Paint fallen leaves layer using (B) vertex channel.
FallenLeaves ObjectSpaceScale: If UseWorldSpaceUVs is unchecked, scales texture.

<u>The following parameters are available when ***EnableMossLayer*** is checked:</u>

MossBaseColor
MossMetallicSmoothness: (R) is Metallic, (G) is AO, (A) is Smoothness
MossNormal

MossNormalIntensity: Increase/reduce normal shading
MossColorTint: Tints Moss BaseColor
MossColorTintAmount
MossTiling: Tile UV
MossHeight: Visibility of moss layer
MossFalloff: Sharpness of moss layer edge

MossEdgeColor: Tint moss near layer edge
MossEdgeHeight: Increase distance from layer edge
MossEdgeFalloff: Tinted moss sharpness

MossCavityAmount: Increase moss within the object's cavities. Uses AO map (Green channel of object's MetallicSmoothness texture.
MossAmountTop: Increase moss along the top of the object using worldspace Y mask.
InvertTopMoss: Inverts Y mask so that moss grows along the bottom of the object instead.
WorldSpaceMossHeight: Visibility of worldspace moss. This option adds a layer of moss using noise with worldspace UVs, as a way to add variation to object instances.
WorldSpaceMossFalloff: Sharpness of worldspace moss layer.
WorldSpaceMossOffset: Offset position of noise texture used to generate this moss layer.
WorldSpaceMossTiling: Increase/decrease tiling (and therefore size) of noise texture.

EnableMossBakedGI: When enabled, moss will grow within the shaded areas of the environment after Global Illumination (static baked lighting) has been baked. If baked lighting isn't being used in your project, then this should be disabled otherwise the

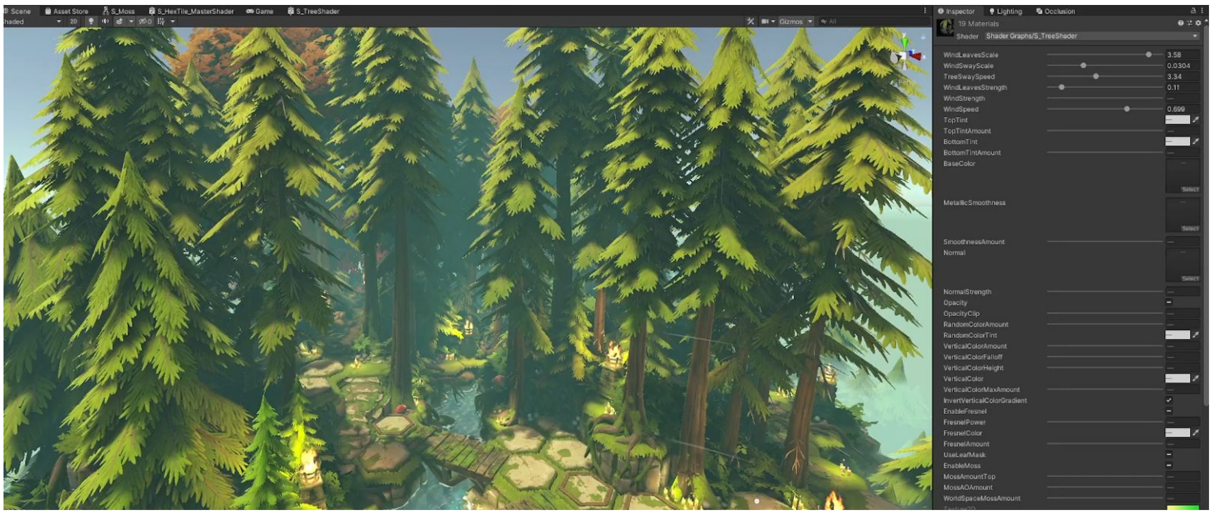
shader will default to applying the moss layer to the entire object if no lightmap textures can be found.
MossBakedGIHeight: Increase, decrease visibility of GI moss layer.
MossBakedGIFalloff: Sharpness of GI moss layer.

EnableMossLayer
EnableDirtLayer
Enable secondary material layers and the properties mentioned above.

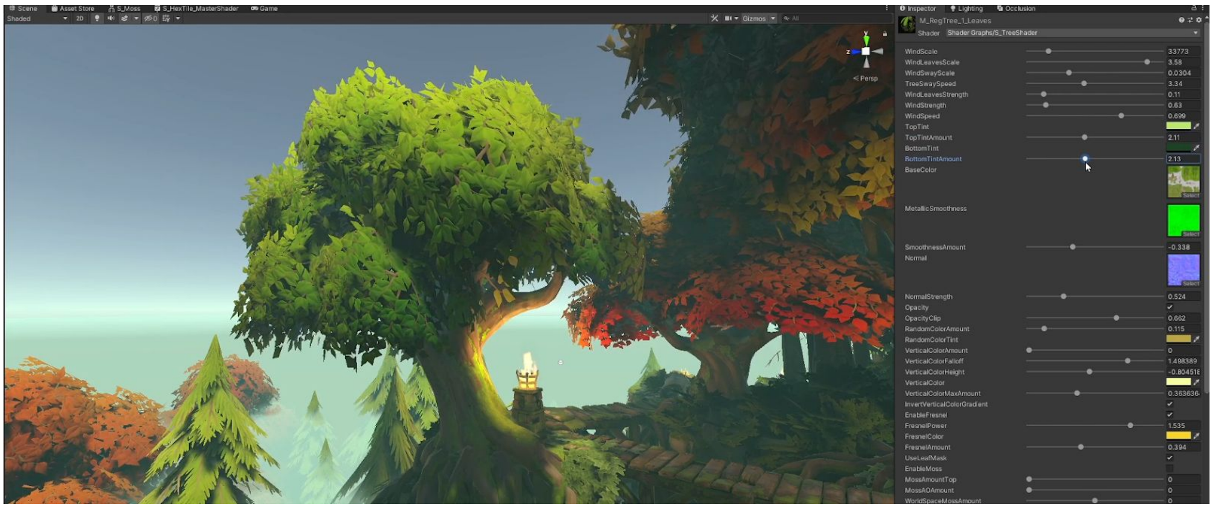
EnableAnimatedEmission
Enable animated noise masks for emission

S_TreeShader

Primary shader used for tree assets, both for leaves and tree trunk. Has a wind effect which takes advantage of painted vertex colors on each of the tree meshes - with the red channel controlling sway/object-scale wind, while blue channel controls blowing leaves.



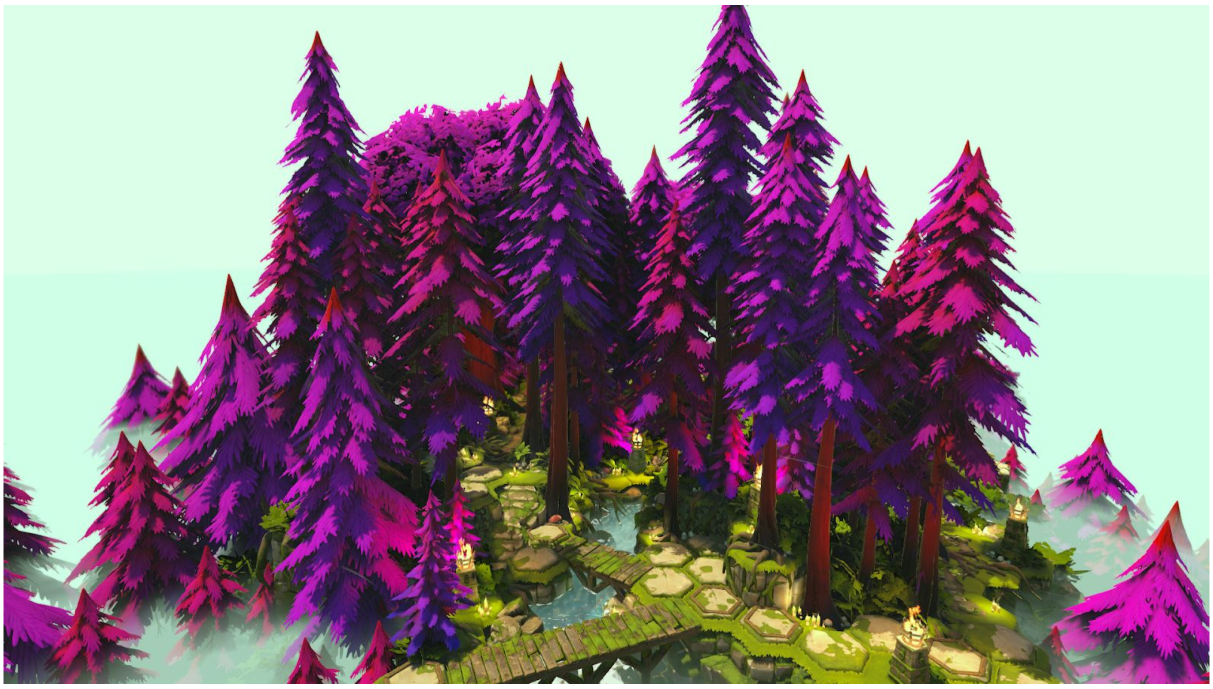
Shader	Shader Graphs/S_TreeShader
WindLeavesScale	3.58
WindSwayScale	0.0304
TreeSwaySpeed	3.34
WindLeavesStrength	0.11
WindStrength	—
WindSpeed	0.699
TopTint	—
TopTintAmount	—
BottomTint	—
BottomTintAmount	—
BaseColor	—
MetallicSmoothness	—
SmoothnessAmount	—
Normal	—
NormalStrength	—
Opacity	—
OpacityClip	—
RandomColorAmount	—
RandomColorTint	—
VerticalColorAmount	—
VerticalColorFalloff	—
VerticalColorHeight	—
VerticalColor	—
VerticalColorMaxAmount	—
InvertVerticalColorGradient	✓
EnableFresnel	—
FresnelPower	—
FresnelColor	—
FresnelAmount	—
UseLeafMask	—
EnableMoss	—
MossAmountTop	—
MossAOAmount	—
WorldSpaceMossAmount	—



M_RegTree_1_Leaves

Shader: Shader Graphs/S_TreeShader

WindScale	33773
WindLeavesScale	3.58
WindSwayScale	0.0304
TreeSwaySpeed	3.34
WindLeavesStrength	0.11
WindStrength	0.63
WindSpeed	0.699
TopTint	🟩
TopTintAmount	2.11
BottomTint	🟩
BottomTintAmount	2.13
BaseColor	Select
MetallicSmoothness	Select
SmoothnessAmount	-0.338
Normal	Select
NormalStrength	0.524
Opacity	✓
OpacityClip	0.662
RandomColorAmount	0.115
RandomColorTint	🟨
VerticalColorAmount	0
VerticalColorFalloff	1.498389
VerticalColorHeight	-0.804518
VerticalColor	🟨
VerticalColorMaxAmount	0.363636
InvertVerticalColorGradient	✓
EnableFresnel	✓
FresnelPower	1.535
FresnelColor	🟨
FresnelAmount	0.394
UseLeafMask	✓
EnableMoss	
MossAmountTop	0
MossAOAmount	0
WorldSpaceMossAmount	0


Tree assets with “ViewVertexColors” enabled. Red channel: Tree Sway, Blue Channel, Leaf Wind

<u>Shader Properties:</u>

ViewVertexColors: Replaces BaseColor with vertex colors, useful for debug purposes to see which areas of the mesh are affected by the wind effect.
WindLeavesScale: Scale wind noise texture for leafs
WindSwayScale: Scale wind noise texture for tree trunk
TreeSwaySpeed
WindLeavesStrength:Amount of distortion to leaves caused by wind
WindStrength: Global wind strength value
WindSpeed: Global wind speed value
(Make sure that all wind parameters are the same for both the leaf mesh and tree trunk mesh otherwise they will no longer be aligned)

TopTint
BottomTint
TopTintAmount

BottomTintAmount
Color tint modifiers for tree leaves.

BaseColor
MetallicSmoothness: (R) is Metallic, (G) is AO, (A) is Smoothness
Normal
SmoothnessAmount: Modify specularity value
NormalStrength: Increase/decrease normal map shading for leaves, lower values give a more flatter style.
Opacity: Check if material is used for leafs rather than tree trunk.
OpacityClip: Value at which Unity “clips” the texture to fully transparent. Lower values give the mesh a more “fuller” look, higher values allow more of the interior of the leaf mesh to be seen. Also affects real time shadows.

RandomColorAmount
RandomColorTint
Tints the leaves a random color based on mesh position in the world.

EnableFresnel
FresnelPower
FresnelColor
Adds a “fresnel” shading effect to the mesh, tinting the edges of the visible mesh to give it a more “fuzzy” appearance.

The following parameters are available when ***EnableMossLayer*** is checked:

MossAmount
MossAmountTop: Increase moss along the top of the object using worldspace Y mask.
MossCavityAmount: Increase moss within the object’s cavities. Uses AO map (Green channel of object’s MetallicSmoothness texture.
WorldSpaceMossAmount: Visibility of worldspace moss. This option adds a layer of moss using noise with worldspace UVs, as a way to add variation to object instances.

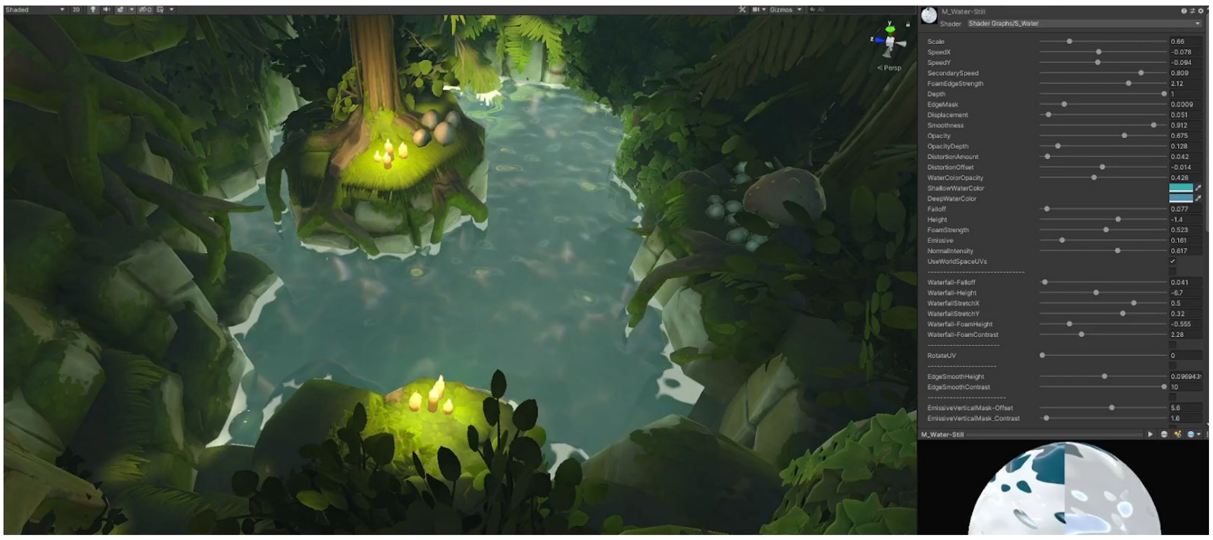
EnableMossLayer: Enable secondary moss material layer with properties controlled above.

UseLeafMask: For leaf materials, masks out the branch texture so that they’re not affected by color tinting or other shader effects

S_Water

The water shader is highly customisable with a number of different parameters to modify edge foam, refraction, color, as well as a casutics effect that appears projected onto underwater objects (though the effect is only visible when looking into the water from above, rather than below).

The water shader also works in conjunction with a script which modifies the “UVRotation” parameter in the shader to allow the flow of water to move with the rotation of the prefab. Note however this is still quite experimental and the script generates a new instance for each tile, so the water can no longer be modified by editing the core water material. For that reason, versions of the water prefabs with the script attached have “_UVRotate” affixed to their name alongside regular versions.



```
M_Water-Still
Shader  Shader Graphs/S_Water

Scale                               0.66
SpeedX                              -0.078
SpeedY                              -0.094
SecondarySpeed                      0.809
FoamEdgeStrength                    2.12
Depth                               1
EdgeMask                            0.0009
Displacement                        0.051
Smoothness                          0.912
Opacity                             0.675
OpacityDepth                        0.128
DistortionAmount                    0.042
DistortionOffset                    -0.014
WaterColorOpacity                   0.426
ShallowWaterColor                   🟦
DeepWaterColor                      🟦
Falloff                             0.077
Height                              -1.4
FoamStrength                        0.523
Emissive                            0.161
NormalIntensity                     0.617
UseWorldSpaceUVs                    ✓
------------------------------
Waterfall-Falloff                   0.041
Waterfall-Height                    -6.7
WaterfallStretchX                   0.5
WaterfallStretchY                   0.32
Waterfall-FoamHeight                -0.555
Waterfall-FoamContrast              2.28
----------------------
RotateUV                            0
----------------------
EdgeSmoothHeight                    0.0969431
EdgeSmoothContrast                  10
------------------------
EmissiveVerticalMask-Offset         5.6
EmissiveVerticalMask_Contrast       1.6
M_Water-Still
```

Make sure that “Opaque Texture” is enabled in “UniversalRP-HighQuality” asset for accurate water and fire rendering.

Shader Properties:

Scale
SpeedX
SpeedY
Control panning speed and scale of noise texture used for water.

SecondarySpeed: Multiplier of previous X/Y speed values for secondary noise texture.

FoamEdgeStrength: Intensity of edge foam, smaller values extend the foam outwards.

Displacement: Increase wave height in 3D space.

Smoothness: Specularity value

Opacity: Opacity of non-foam water.
OpacityDepth: Opacity gradient beginning from water surface.

DistortionAmount: Refraction of surface below water.
DistortionOffset: Offset distortion left and right. Used to correct offset created by increasing DistortionAmount value.

WaterColorOpacity
ShallowWaterColor
DeepWaterColor
Parameters for colorizing water.

FoamFalloff: Sharpness of foam edge.
FoamHeight: Height at which objects below the surface affect the amount of foam produced.
FoamStrength:Intensity of edge foam, increasing the value extends the foam outwards.

Emissive: Adds an emissive value to water, saturating color values. Useful for certain art styles or for depicting other types of liquid.
NormalIntensity: Normal map intensity. Decreases visibility of waves in conjunction with Smoothness value.

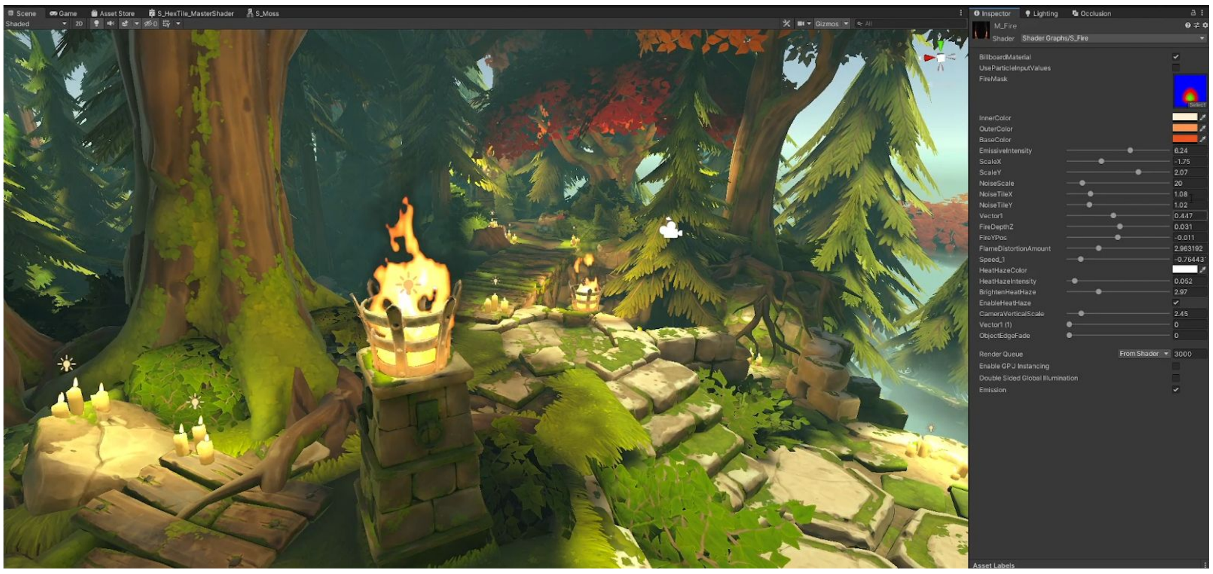
UseWorldSpaceUVs: Defaults to world-space to remove seams between the hextile meshes. You may wish to disable and revert to object-space UVs if the material is being used on a custom water mesh, such as a river mesh where the UVs conform to the curves of the river.

Waterfall-Falloff: Sharpness of waterfall foam
Waterfall-Height: Density of foam
Waterfall-StretchX/Y: Stretches texture along X/Y Axis based on verticality.
Waterfall-FoamHeight: Extra density parameter
Water-FoamContrast: Extra contrast parameter

RotateUV: Rotate direction of material (and therefore flow direction)

CausticsScale: Size of caustics texture
CausticsMaskHeight: Max height of caustics projection onto underwater objects.
CausticsIntensity: visibility of caustics layer
CausticsSurfaceMaskHeight: Visibility of caustics near water surface.

S_Fire

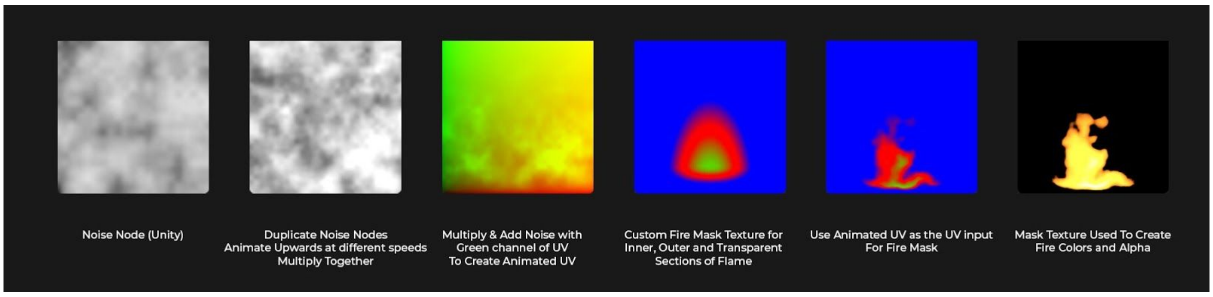


M_Fire
Shader: Shader Graphs/S_Fire

BillboardMaterial	✓
UseParticleInputValues	
FireMask	
InnerColor	⬜
OuterColor	🟧
BaseColor	🟥
EmissiveIntensity	6.24
ScaleX	-1.75
ScaleY	2.07
NoiseScale	20
NoiseTileX	1.08
NoiseTileY	1.02
Vector1	0.447
FireDepthZ	0.031
FireYPos	-0.011
FlameDistortionAmount	2.963192
Speed_1	-0.76443
HeatHazeColor	⬜
HeatHazeIntensity	0.052
BrightenHeatHaze	2.97
EnableHeatHaze	✓
CameraVerticalScale	2.45
Vector1 (1)	0
ObjectEdgeFade	0
Render Queue	From Shader 3000
Enable GPU Instancing	
Double Sided Global Illumination	
Emission	✓

Make sure that “Opaque Texture” is enabled in “UniversalRP-HighQuality” asset for accurate water and fire rendering.

The fire shader works by distorting the UVs of a RGB mask texture with Shader Graph's built-in noise functions. A short breakdown of the effect can be seen here:


Noise Node (Unity)
Duplicate Noise Nodes Animate Upwards at different speeds Multiply Together
Multiply & Add Noise with Green channel of UV To Create Animated UV
Custom Fire Mask Texture for Inner, Outer and Transparent Sections of Flame
Use Animated UV as the UV input For Fire Mask
Mask Texture Used To Create Fire Colors and Alpha

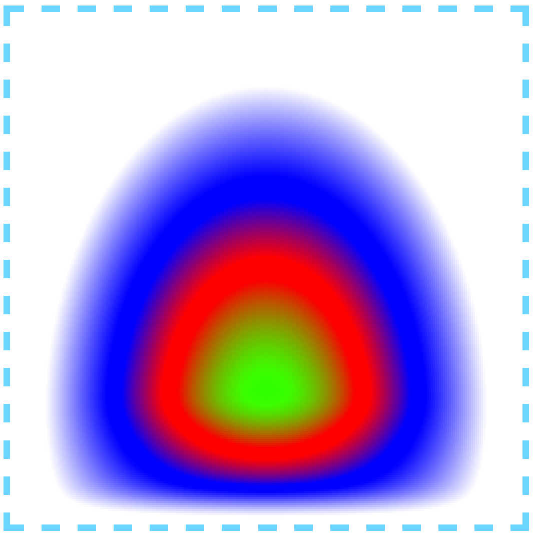
The fire shader can be customized in a number of different ways for use in different assets - for example the same shader used for the braziers is also used for the candles, just with certain parameters scaled down to give the appearance of a candle flame.

The effect was inspired by the work done in the game RiME by Tequila Works, information of which can be found in this talk: https://youtu.be/ExD_p3hsV80?t=202 and on this blog post: https://cyangamedev.wordpress.com/2020/08/04/fire-shader-breakdown/

Shader Properties:

BillboardMaterial: When enabled the mesh will always face the camera.

FireMask: RGBA mask texture which corresponds to the inner, outer and haze sections of the flame. Can be replaced to create certain styles or different flame shapes - for example a "flat" version of the mask is included which creates a more stylized version of the effect, as well as a horizontally tiling version which could be used for a long trail of fire, for example.
ViewRGBA-FireMask: Debug view for mask


Default RGBA mask

InnerColor
MidColor
OuterColor
Color individual sections of flame based on fire mask.

EmissiveIntensity

Speed: Fire animation speed
ScaleX/Y: Scale fire effect independent of mesh

NoiseScale: Distortion noise scale
NoiseTileX/Y

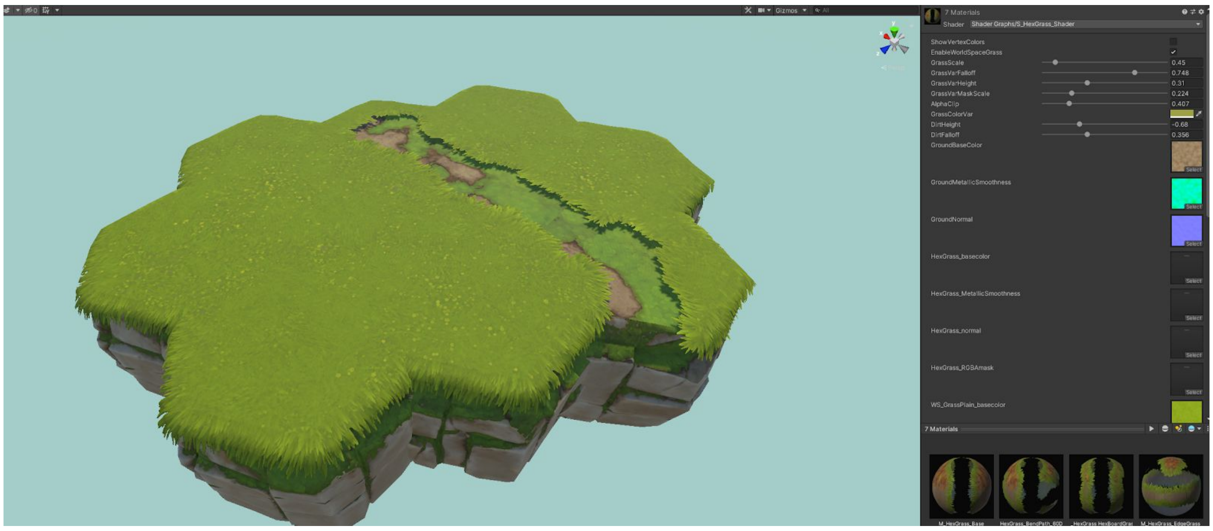
FireDepthZ: Offset fire effect inwards/outwards
FireYPos: Offset effect on Y axis

FlameDistortionAmount: Intensity of noise distortion

EnableHeatHaze: Enable background heat distortion using depth buffer
HeatHazeColor: Colorize haze effect
HeatHazeIntensity: Intensity of distortion
HeatHazeBrightness

CameraVerticalScale: Resizes flame on Y axis based on camera rotation, reducing the size of the flame when the camera is looking vertically downwards.

S_HexGrass_Shader



7 Materials
Shader: Shader Graphs/S_HexGrass_Shader

ShowVertexColors	☐
EnableWorldSpaceGrass	✓
GrassScale	0.45
GrassVarFalloff	0.748
GrassVarHeight	0.31
GrassVarMaskScale	0.224
AlphaClip	0.407
GrassColorVar	
DirtHeight	-0.68
DirtFalloff	0.356
GroundBaseColor	
GroundMetallicSmoothness	
GroundNormal	
HexGrass_basecolor	
HexGrass_MetallicSmoothness	
HexGrass_normal	
HexGrass_RGBAmask	
WS_GrassPlain_basecolor	

The HexGrass shader is used for the HexGrass tile assets, and has a number of properties to control color tint, worldspace UV grass and the wind effect.

Shader Properties:

ShowVertexColors: Replace BaseColor with vertex colors. Useful for seeing areas of the mesh affected by wind (Blue channel), areas where dirt (Green) or fallen leaves (Red) have been painted, and other debug purposes.

WindAmount: Wind multiplier
WindSpeed: Speed of wind noise texture

GrassScale: Scale factor for worldspace UV grass

The following parameters are available when ***EnableWorldSpaceGrass*** is checked:

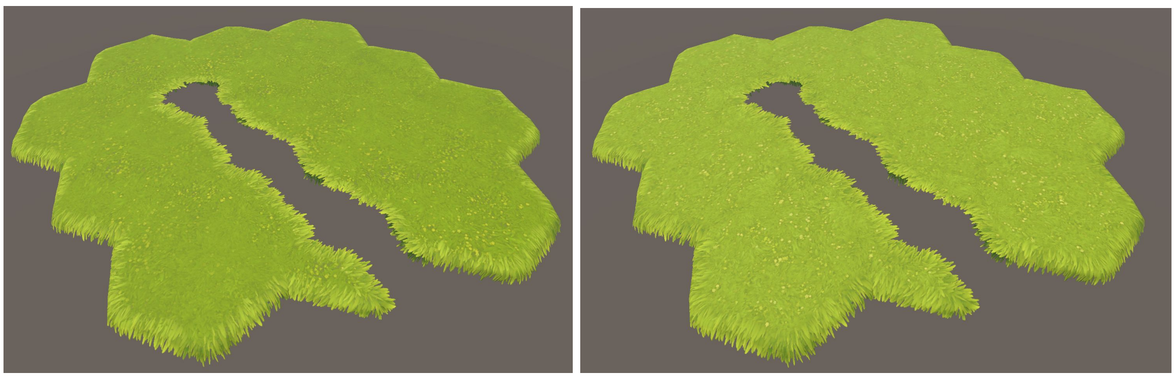
GrassVarFalloff
GrassVarHeight
Worldspace grass consists of two different sets of grass textures blended together using a noise mask to further breakup repetition. These parameters affect visibility of the secondary layer and sharpness.

GrassVarMaskScale: Scale of noise mask used to blend each grass texture set.

DirtHeight
DirtFalloff
If dirt has been painted onto the mesh using the green vertex channel, controls visibility and sharpness of mask.

AlphaClip: Effects cutoff point of alpha texture.

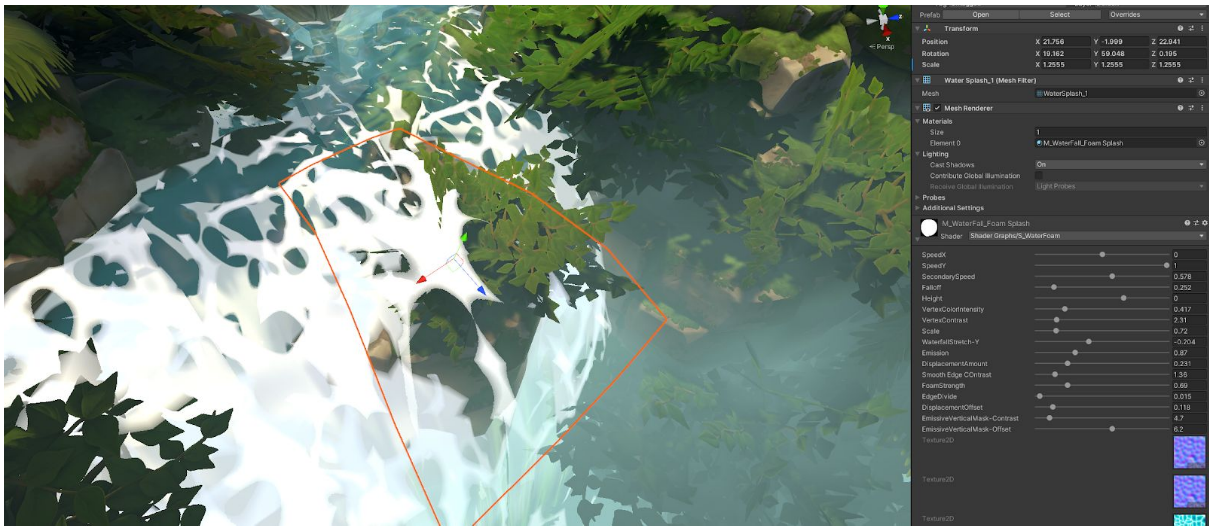
EnableWorldSpaceGrass: Enable secondary grass layer which tiles in world space. This reduces seams across meshes while also allowing for dirt and fallen leaves to be painted using vertex colors. Disable for a more performance-friendly shader that should suffice in most cases, enable for scenes with larger open fields where repetition is more noticeable.


EnableWorldSpaceGrass: enabled (left), disabled (right)

Other Shaders

S_WaterFoam

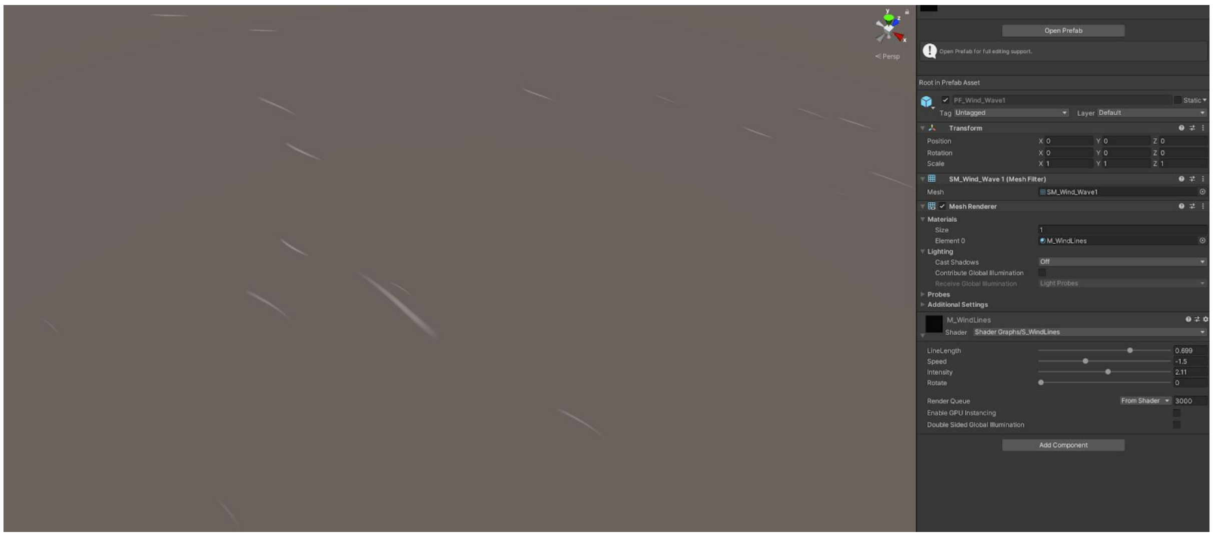
Simpler water shader designed for water splash cards and foam, has much of the same parameters as the standard S_Water shader.



```
Transform
Position                          X 21.756   Y -1.999   Z 22.941
Rotation                          X 19.162   Y 59.048   Z 0.195
Scale                             X 1.2555   Y 1.2555   Z 1.2555
Water Splash_1 (Mesh Filter)
Mesh                              WaterSplash_1
Mesh Renderer
Materials
  Size                            1
  Element 0                       M_WaterFall_Foam Splash
Lighting
  Cast Shadows                    On
  Contribute Global Illumination
  Receive Global Illumination     Light Probes
Probes
Additional Settings
M_WaterFall_Foam Splash
Shader  Shader Graphs/S_WaterFoam
SpeedX                            0
SpeedY                            1
SecondarySpeed                    0.578
Falloff                           0.252
Height                            0
VertexColorIntensity              0.417
VertexContrast                    2.31
Scale                             0.72
WaterfallStretch-Y                -0.204
Emission                          0.87
DisplacementAmount                0.231
Smooth Edge COntrast              1.36
FoamStrength                      0.69
EdgeDivide                        0.015
DisplacementOffset                0.118
EmissiveVerticalMask-Contrast     4.7
EmissiveVerticalMask-Offset       6.2
Texture2D
Texture2D
Texture2D
```

S_WindLines

Shader for animated stylized wind lines, prefabs can be found in:
Assets/VFX/Wind/Prefabs



```
Open Prefab for full editing support.
Root in Prefab Asset
PF_Wind_Wave1                                   Static
Tag Untagged                  Layer Default
Transform
Position                          X 0   Y 0   Z 0
Rotation                          X 0   Y 0   Z 0
Scale                             X 1   Y 1   Z 1
SM_Wind_Wave 1 (Mesh Filter)
Mesh                              SM_Wind_Wave1
Mesh Renderer
Materials
  Size                            1
  Element 0                       M_WindLines
Lighting
  Cast Shadows                    Off
  Contribute Global Illumination
  Receive Global Illumination     Light Probes
Probes
Additional Settings
M_WindLines
Shader  Shader Graphs/S_WindLines
LineLength                        0.699
Speed                             -1.5
Intensity                         2.11
Rotate                            0
Render Queue                      From Shader  3000
Enable GPU Instancing
Double Sided Global Illumination
```

S_EmissiveProps

Simple shader specifically for emissive materials. Has much of the functionality as the emissive options in **S_HexTile_MasterShader**