

Vista

User Guide

2023.1.0

PROLOG

Vista is an advanced toolset for procedural terrain and scene editing. Harness the power of GPU calculation, combine its robust graph and biome system, Vista has the following characteristics:

- Non-linear: You can combine nodes in any order to define the terrains, and output multiple data (height, color, vegetation, etc.) at once.
- Non-destructive: Your terrains and scenes are generated using rules defined in the graph, modifying a single property of the graph will give you a whole new terrain. If you want to, you can reverse the previous result without losing any of your work.
- Parametric: You can control nearly every aspect of your terrains just by playing around with simple numbers, the terrains will adapt without sweating. That way you can have infinite variations in a single graph.
- Local: Your terrains are mixed by a set of “local biomes” manually placed in the scene. This way it’s easier for you to control the scene layout and it fits better to your game’s story. (*The core asset only allows 1 biome. To be able to place many biomes in the scene, you need to install the [Big World](#) module*)

Vista is an open and modular system that supports any heightfield terrain system. Currently there are 2 terrain systems that are supported natively: Unity Terrain and [Polaris Terrain](#). You can add support for your own terrain solution as well as extend the generation pipeline via scripting.

In this documentation, we provide you a very [quick guide](#) to help you to get started with the tool. After that, there will be detailed information about every part for you to dive deeper.

For system requirements, please see [this page](#).

For the release log of the main asset, please see [this page](#). For the release log of extension modules, please find in their corresponding section.

If you have any question or inquiry, please contact us via pinwheel.customer@gmail.com or join our [Discord](#) server.

Note

Images in this document are taken from the development machine for demonstration purposes, with all modules available at the time installed. They may look a bit different from the version you're using, the modules you have, please contact us if you wonder which module a button comes from.

SYSTEM REQUIREMENTS

- **Unity 2020.3 LTS and up.**
- **Compute shaders support.** The asset is heavily built upon compute shaders and compute buffers. This is supported on most machines if you use the asset in the editor only. [Learn more](#).
- **Instancing support.** Some parts of the asset require GPU Instancing to work. [Learn more](#).

VERSIONS

General update guide

A new version may contain major changes in structure or scripts that can seriously break your project, so it's best to practice the following process to safely update Vista:

- Always make a backup of the project, using version control software or copy the whole project to another drive.
- Locate and delete all asset files, including the core asset & all modules if any, except the terrain graphs (and other user-created contents) you are working on.
- If there is any script error pops in, open Project Settings > Player > Scripting Symbols and remove the VISTA keyword (and other related keywords if any).
- Open the Package Manager window, update Vista and all modules you have to the newest version.
- Import Vista to your project.
- Import all modules you have, in any order.
- Check if terrain graphs (and other user-created contents) are re-serialized and things work as normal.

Update from V2022

There are many breaking changes when updating from V2022 to V2023, this guide is specifically for someone who already had Vista (version 2022) in their library.

- Due to pricing model change, the version 2022 will be deprecated from the store. We introduce 2 new packages: [Vista 2023](#) (free) and [Vista - Big World Module](#) (paid). **Users of the old Vista 2022 will be able to get the Big World Module for free.**
- First, go to the Asset Store and add the 2 packages above to your asset library. You should see the Big World Module price set as FREE. If not, please contact us.
- Make a backup and open your Unity project, remove all files related to the Vista asset and its modules, except your terrain graph and other user created content.

- Open the Package Manager, download and import Vista 2023 and Big World Module, respectively, then update and install other modules you have.
- Open your terrain graph, at this moment all asset references (terrain layers, tree templates, detail templates, etc.) will be lost, you have to manually re-assign them.
- Check if things work as normal.

Logs

V2023.1.0

IMPORTANT NOTE

- This version contains major changes, please follow the [Update guide](#) to avoid breaking your project.
- This version also changes the way it serializes the graph asset. After upgrading, all asset references in the graph (terrain layer, tree templates, detail templates, object templates, etc.) will be lost. You have to manually re-assign them.

NEW FEATURES

- 3D Viewport: new button for setting default texture display mode when selecting a node.
- 3D/2D Viewport: able to visualize textures with gradients.
- Point Grid Node: adding “Per 100 Meters” option for better tree density when scaling the biome.
- Terrain Sub Graph Node: able to override seed value, editing the sub graph in the context of its parent graph (take input from parent graph)
- New nodes: Gradient Map (requires Big World module)
- Node searcher: displaying a preview of the selected node for better browsing.
- Graph Input: able to provide texture/position inputs to the graph via the LPB interface.
- New asset type: Position Container, a way to provide position input to the graph using editor UI.
- Troubleshooting Mode: a way to quickly find problems in the graph flow, identify the nodes that failed to execute which raise exceptions or crash.
- LPB: New utility button, Snap To Selected Tiles.

IMPROVEMENTS

- Scene View: Display an indicator in the Scene View Overlay when the VM is running.
- Scene View: Display an indicator at the biome location when it is generating data.
- Improving biome blending.
- Remap Node: allow using a min max slider for setting the range.
- Thin Out, Spread, Water Flow, Instance Output Node: better result with deterministic simple noise.
- 3D Viewport: smoother movement without delay when the user first uses the WASD for navigation.
- Asset references in node: better asset serialization, cleaner git.
- Better detail density transition when the density is low or there are many variants.
- Overall API improvements.
- Overall code cleanup & optimization.

FIXES

- Spline Extract Node: fix shader error on Metal (atomic operation on texture).
- Curve node: minor shader fix (getSign(0))
- Subgraph: Fix subgraph connection lost on entering play mode.
- Graph editor: Fix graph editor disappears when entering/exiting play mode and between Unity sessions.
- Fixing normal calculation bugs in shader.
- Fixing memory leaks caused by materials.
- Minor bug fixing.

QUICK START

Installing dependencies

Vista requires the following packages to work:

- Searcher (com.unity.searcher)
- Editor Coroutines (com.unity.editorcoroutines)

These packages will be added to your project automatically. If not, please check the Package Manager.

Creating your terrains

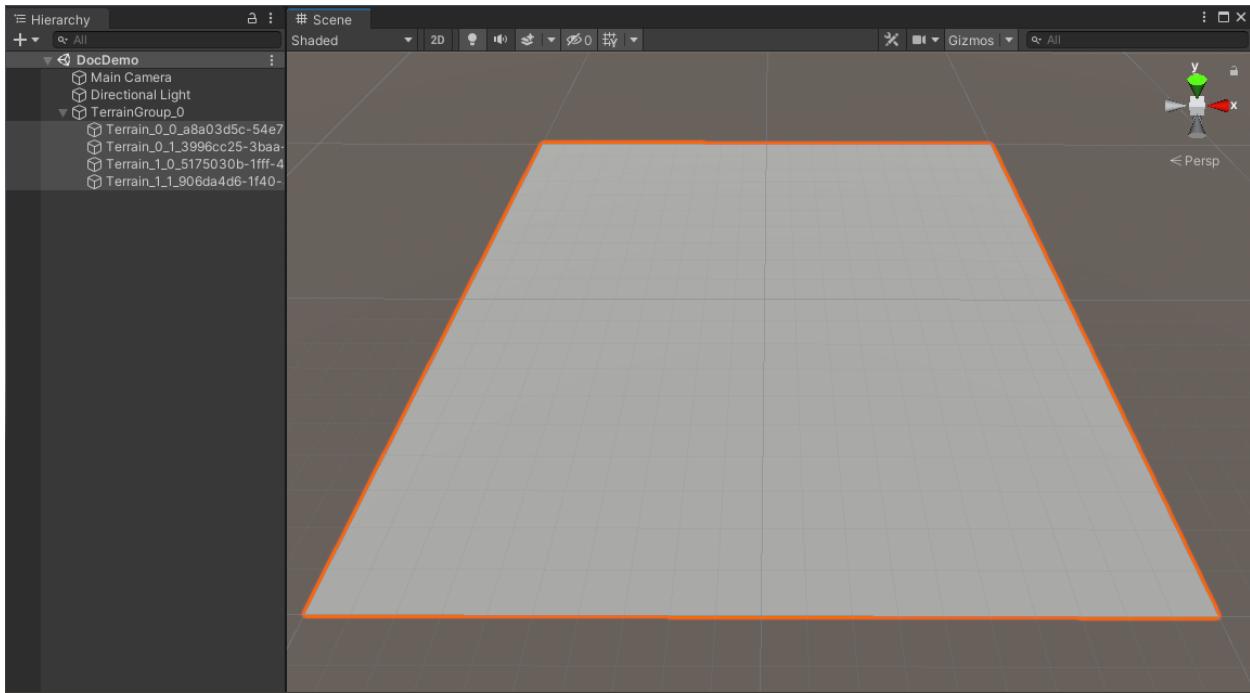
You can create and manage your terrain grid using anything you want.

For Unity Terrain, we suggest the [Terrain Tools](#) package (com.unity.terrain-tools).

For Polaris Terrain, we suggest the [Wizard](#) and the [Group Tool](#).

After using the tools above, you will have some terrains in the scene.

For example, here we have a 2x2 terrain grid, each terrain has the dimension of 500x500x500 meters:

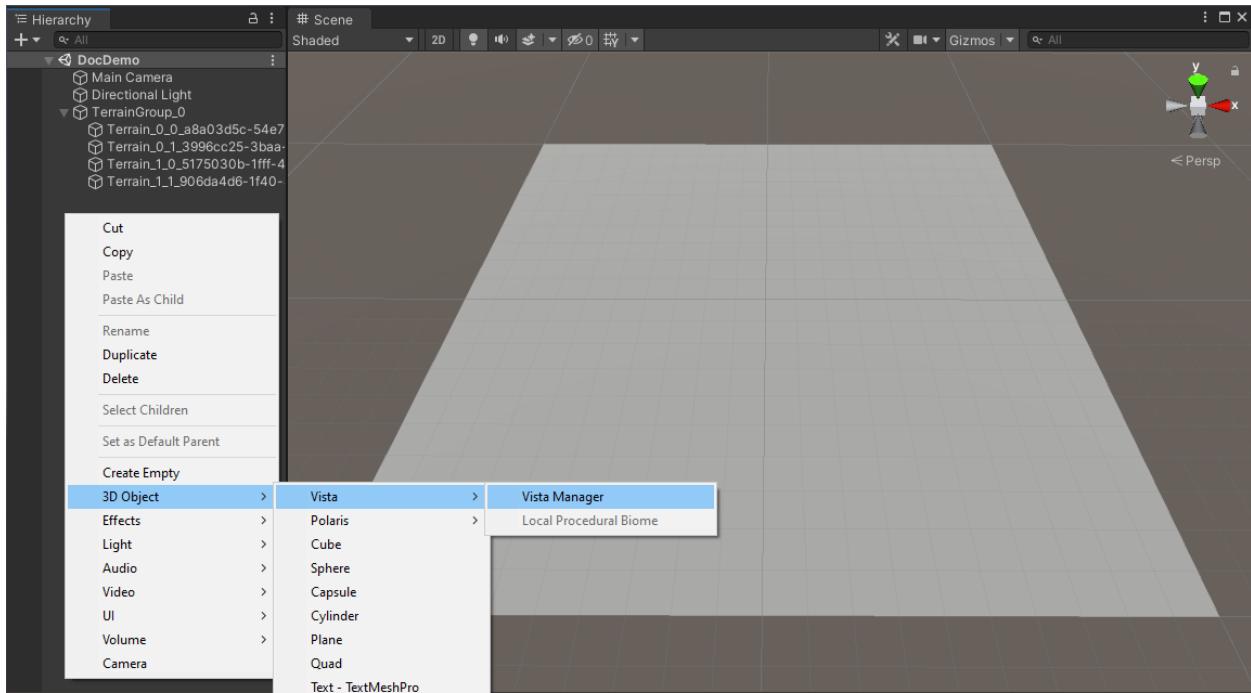


You can also use the tools above to set terrain properties, just ignore the following as they are managed by Vista:

- Height Map Resolution.
- Texture Resolution.
- Detail Density Resolution.

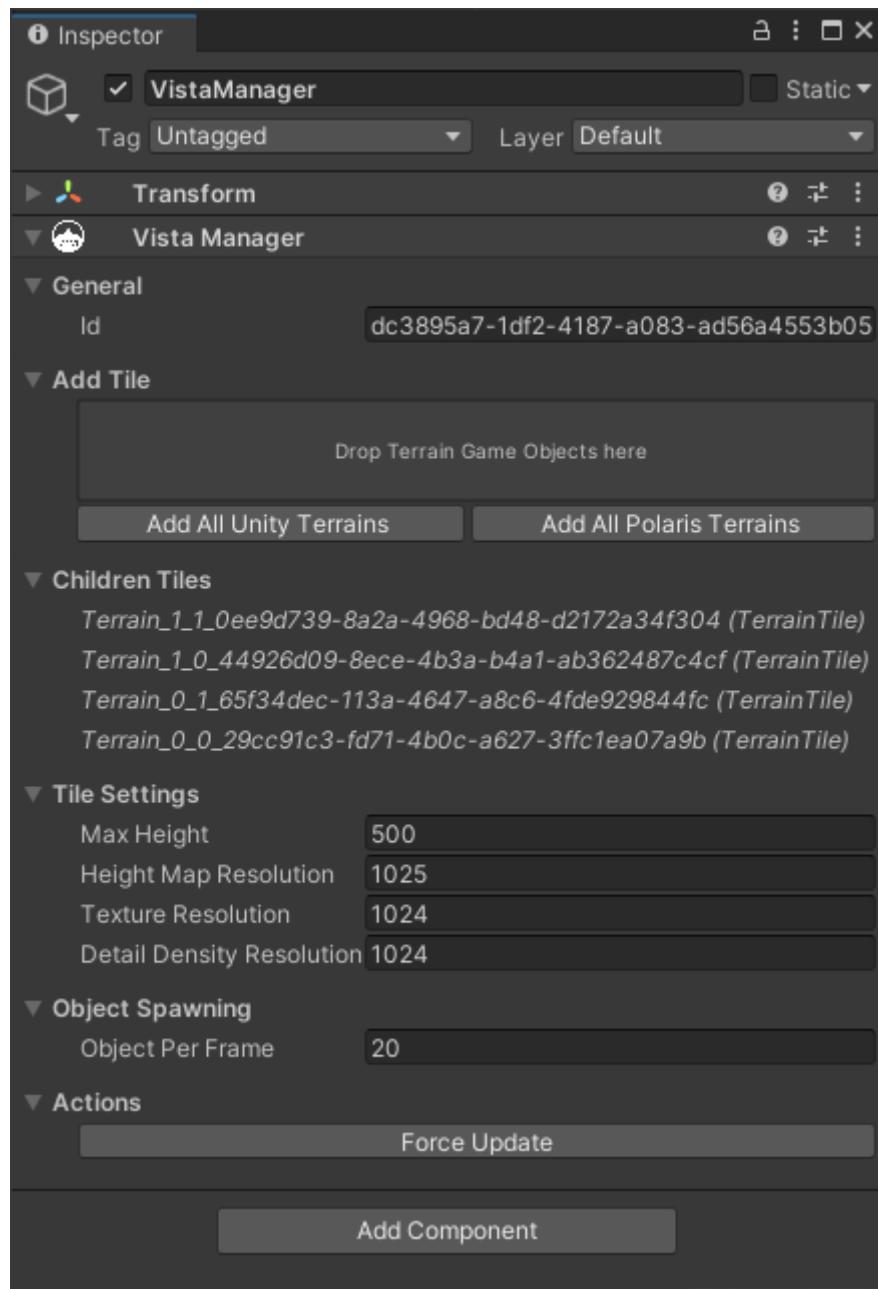
Creating the Vista Manager

Vista Manager is the component responsible for the generation pipeline. To create an instance, right click on the Hierarchy and select **3D Object>Vista>Vista Manager**.

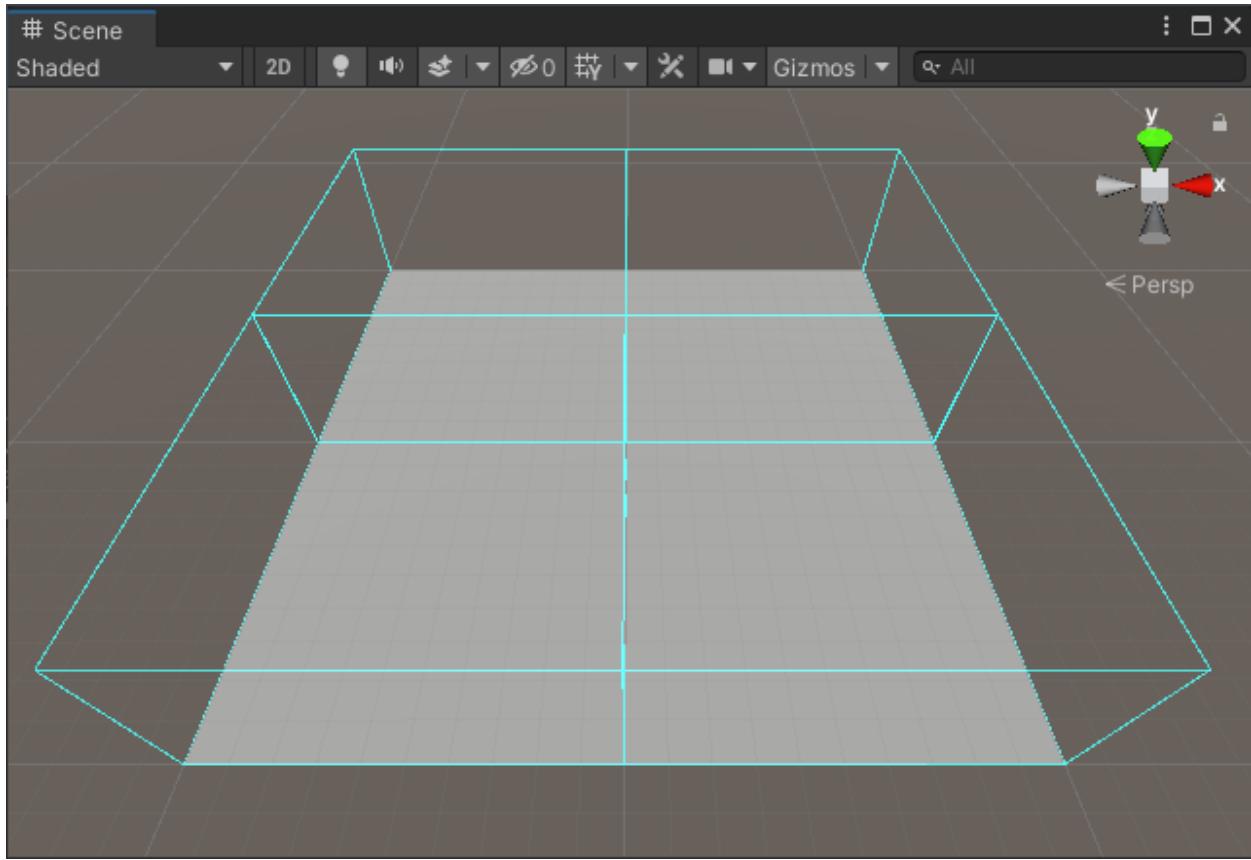


The Vista Manager holds a list of “tiles” which are under its control. A tile is simply a terrain instance of any terrain system, it doesn’t need to be a child game object of the Vista Manager, but can be anywhere in the Hierarchy.

To assign tiles to the Vista Manager, drag the terrain game objects from the Hierarchy and drop them to the “Add Tile” box in the Vista Manager Inspector, or simply click the “Add All Unity/Polaris Terrains” button.



Children tiles will be shown in the Inspector, and highlighted with cyan boxes in the Scene View.



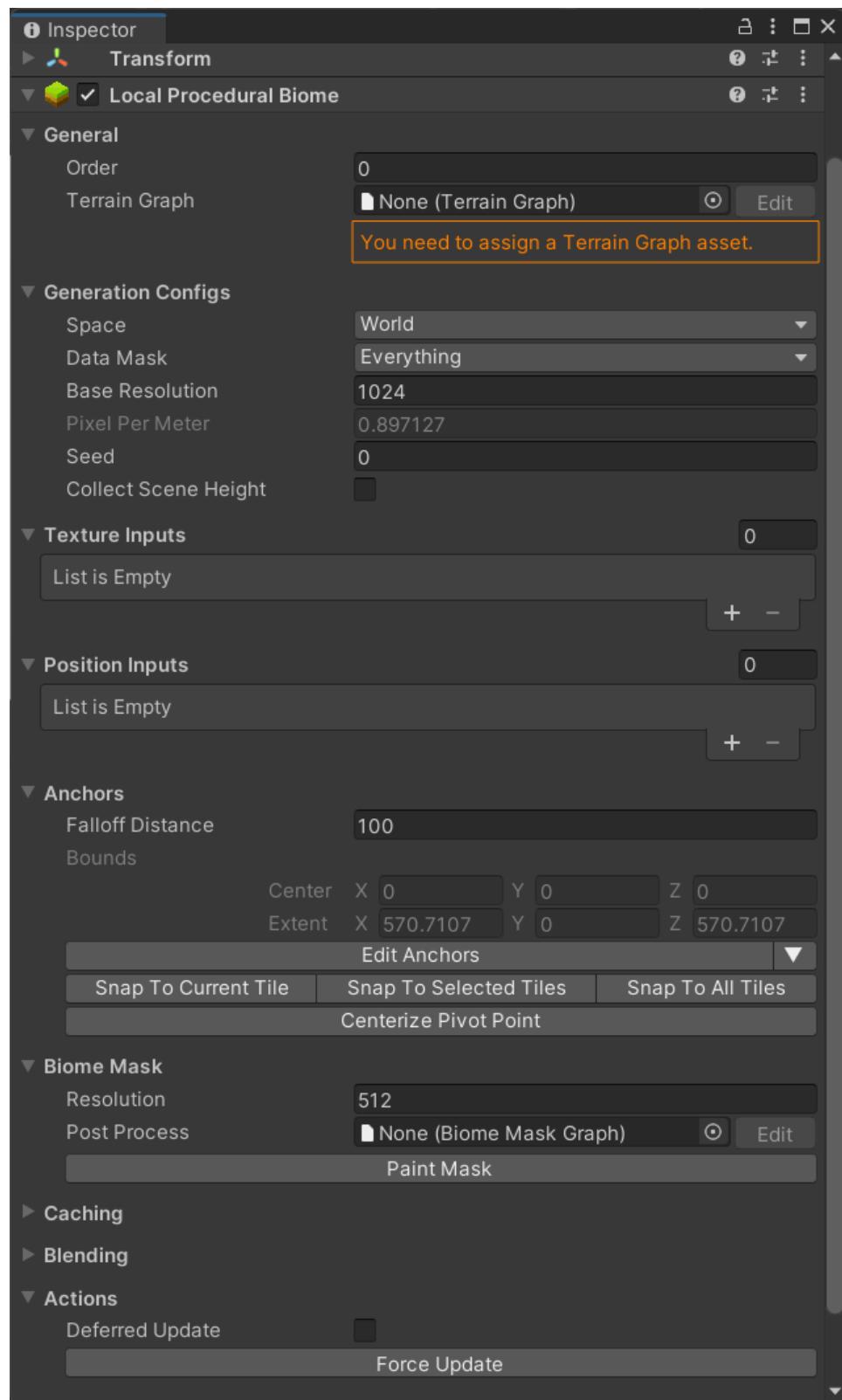
Next, go to the Tile Settings section and set the value to your desired values. We will use 512 for this example since each terrain has the size of 500 meters in width and length.

Creating your first biome

In terrain editing, a biome is a large area in the scene characterized by its geometry (mountain, plateau, river delta, etc.), its textures (grassy, rocky, sandy, snowy, etc.) and its vegetation species.

Usually when a Vista Manager instance is created, it will also create a default biome for you. If not, right click on the Vista Manager, then select **3D Object>Vista>Local Procedural Biome**.

Importance: A biome MUST be a child game object of a Vista Manager to take effect.



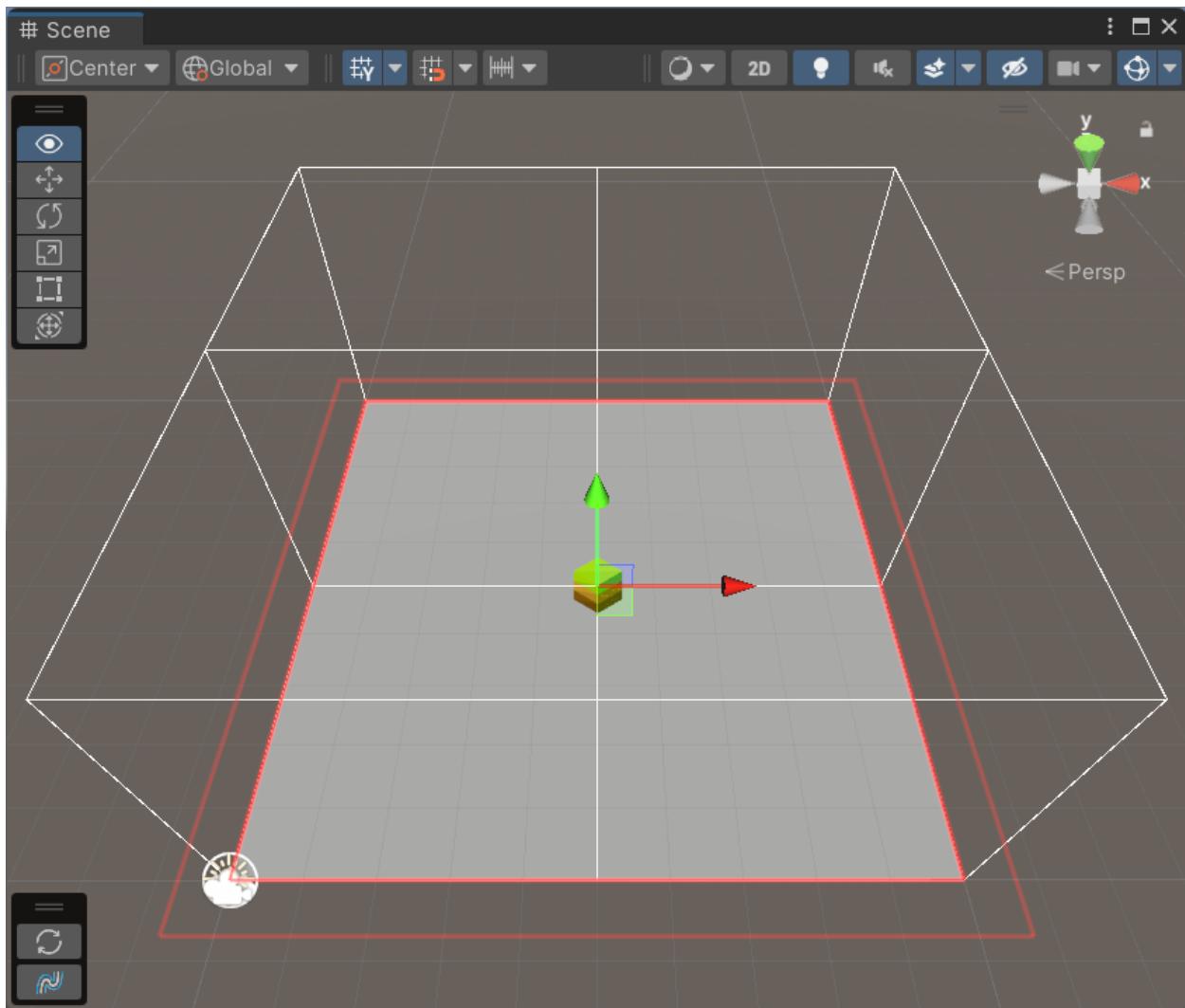
Since the terrain grid covers an area of 1000x1000 meters in width and length, we will set Base Resolution to 1024 to maintain a ~1 meter-per-pixel ratio (You can set any value you want depending on the quality needs).

Note: The core asset will limit texture resolution at 1K, which works perfectly fine at 1000 x 1000 to 2000 x 2000 meters. If you want to have up to 4K maps, consider getting the [Big World](#) module.

With the biome selected, you will see:

- A red polygon indicates the biome's boundaries.
- White boxes indicate the tiles affected by this biome.

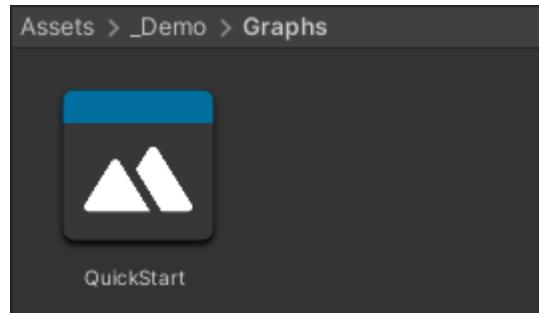
You can click on the Edit Anchors button in the Inspector to start editing the biome boundaries, but for now just use the **Snap To All Tiles** button.



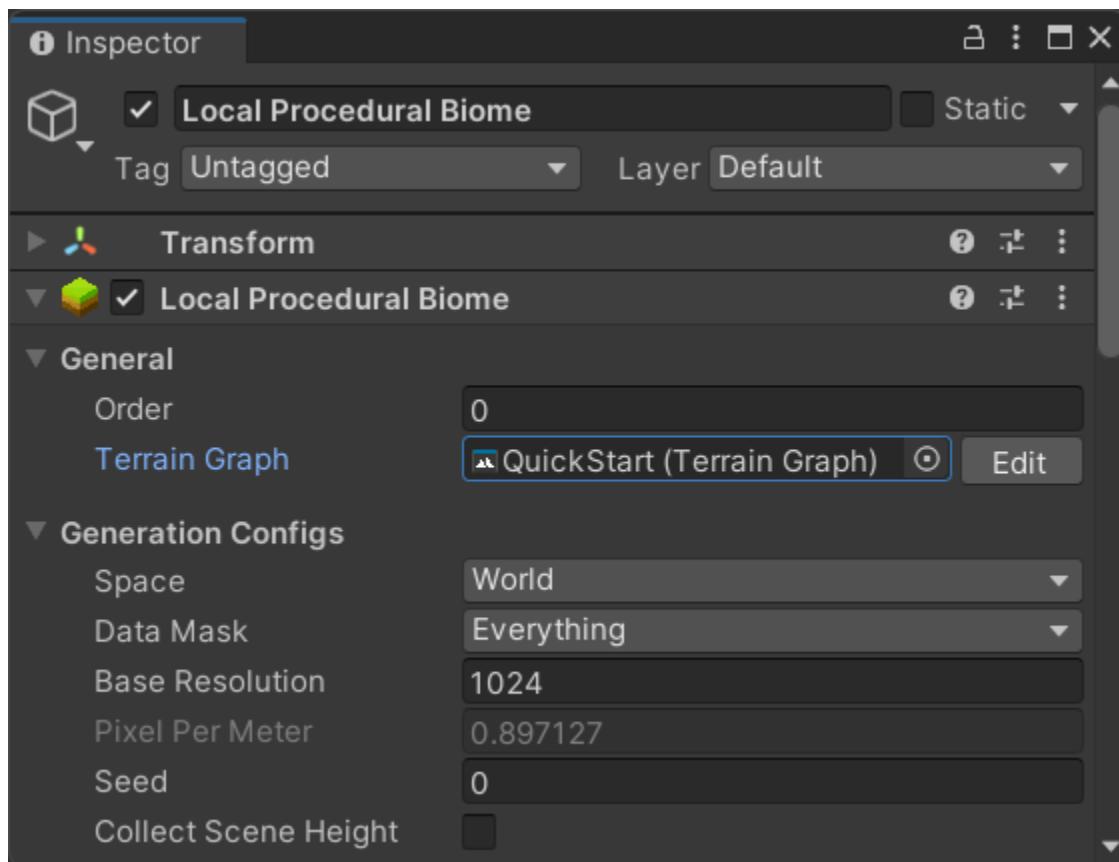
Creating the terrain graph

Terrain graph is an asset containing rules for the terrain generation, defined by the combination of many nodes and connections.

To create a terrain graph, right click on the Project window, then select **Create>Vista>Terrain Graph**.

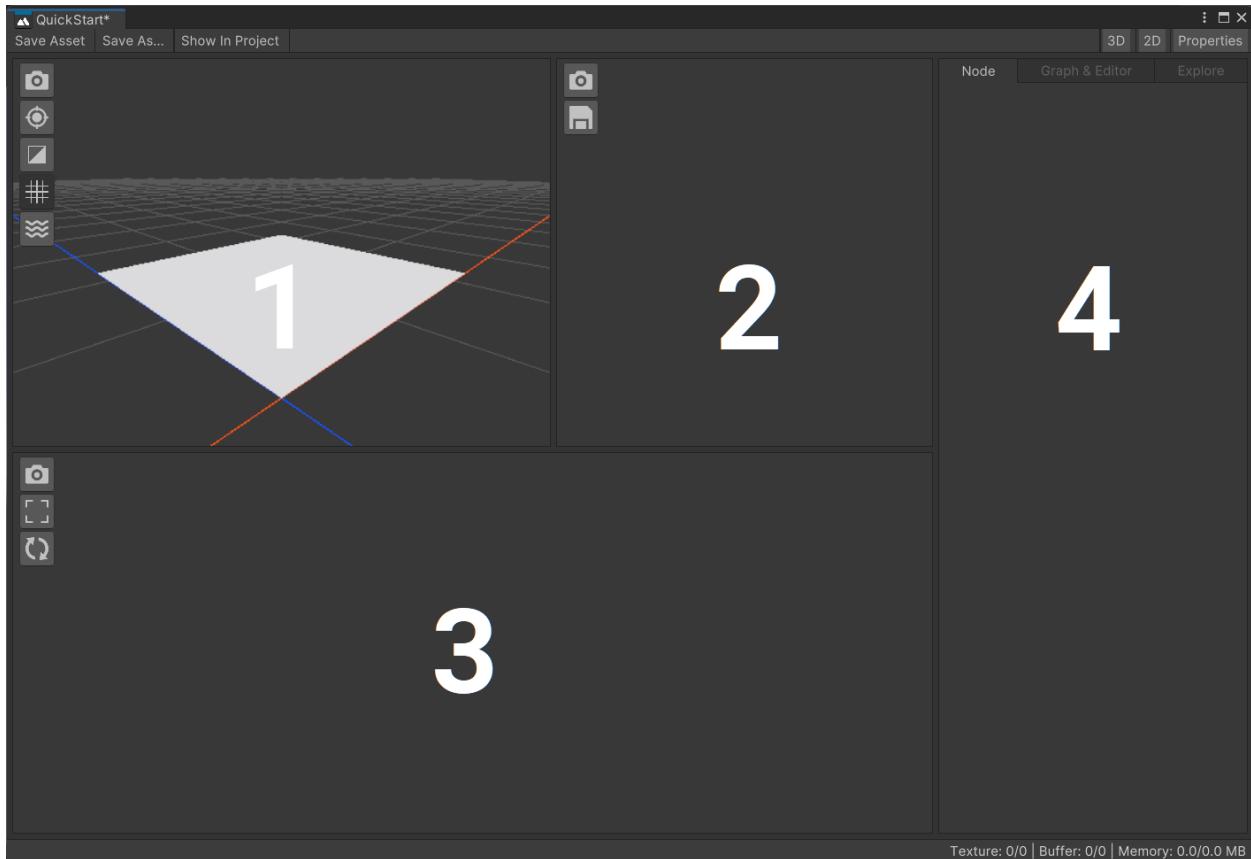


Then, assign the terrain graph to your biome created before.



The Graph Editor

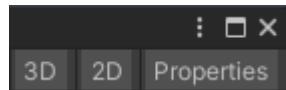
You can open the graph by double clicking on it. But it's better to use the Edit button in the biome Inspector, it will bring up the Graph Editor Window:



In the graph editor, you can find:

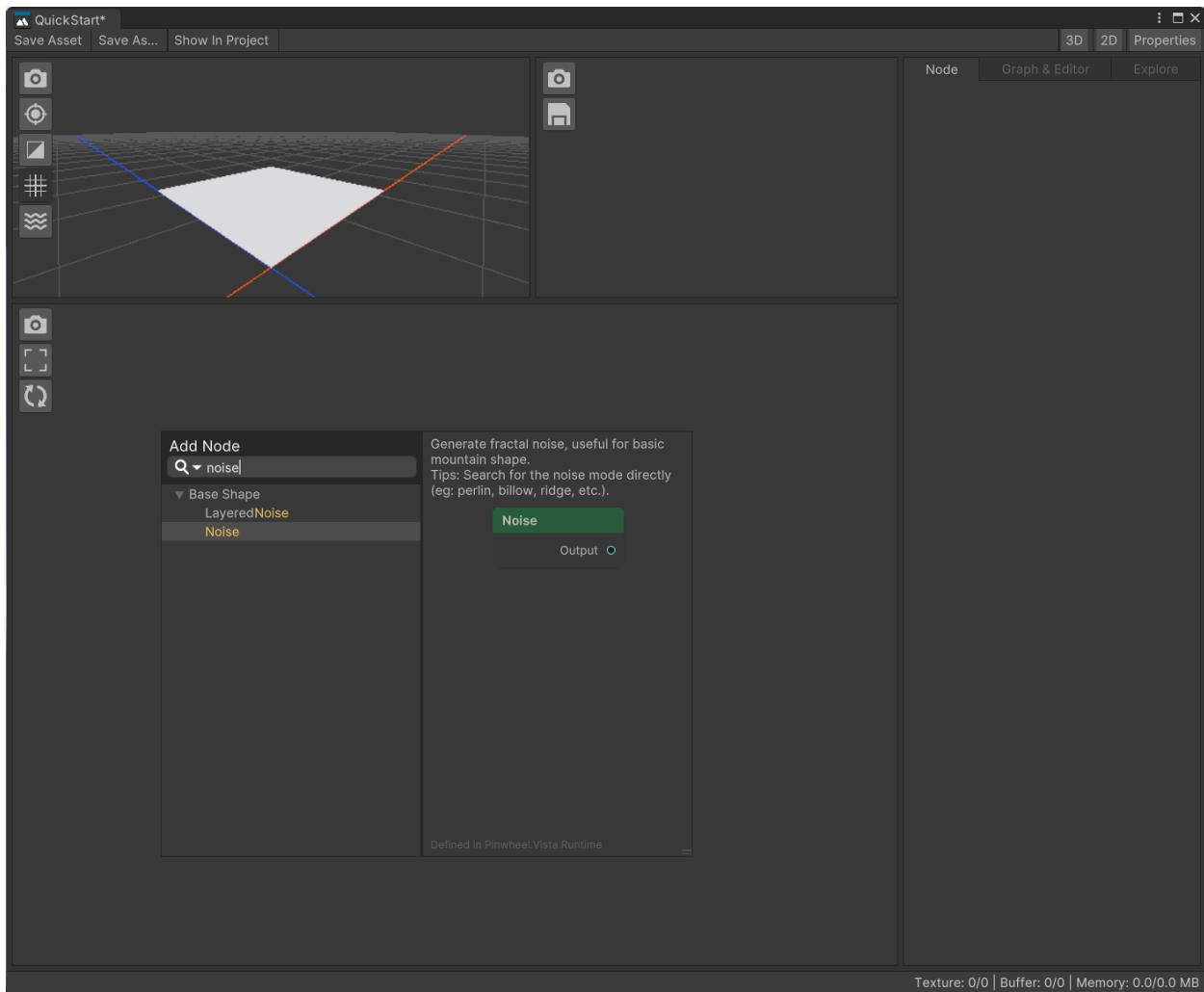
- 3D Viewport (1): Display the preview terrain.
- 2D Viewport (2): Display the generated texture.
- Graph View (3): Display the graph.
- Properties View (4): Display the detail of the selected node as well as the graph itself.

You can show/hide these section using the buttons on the top right corner:

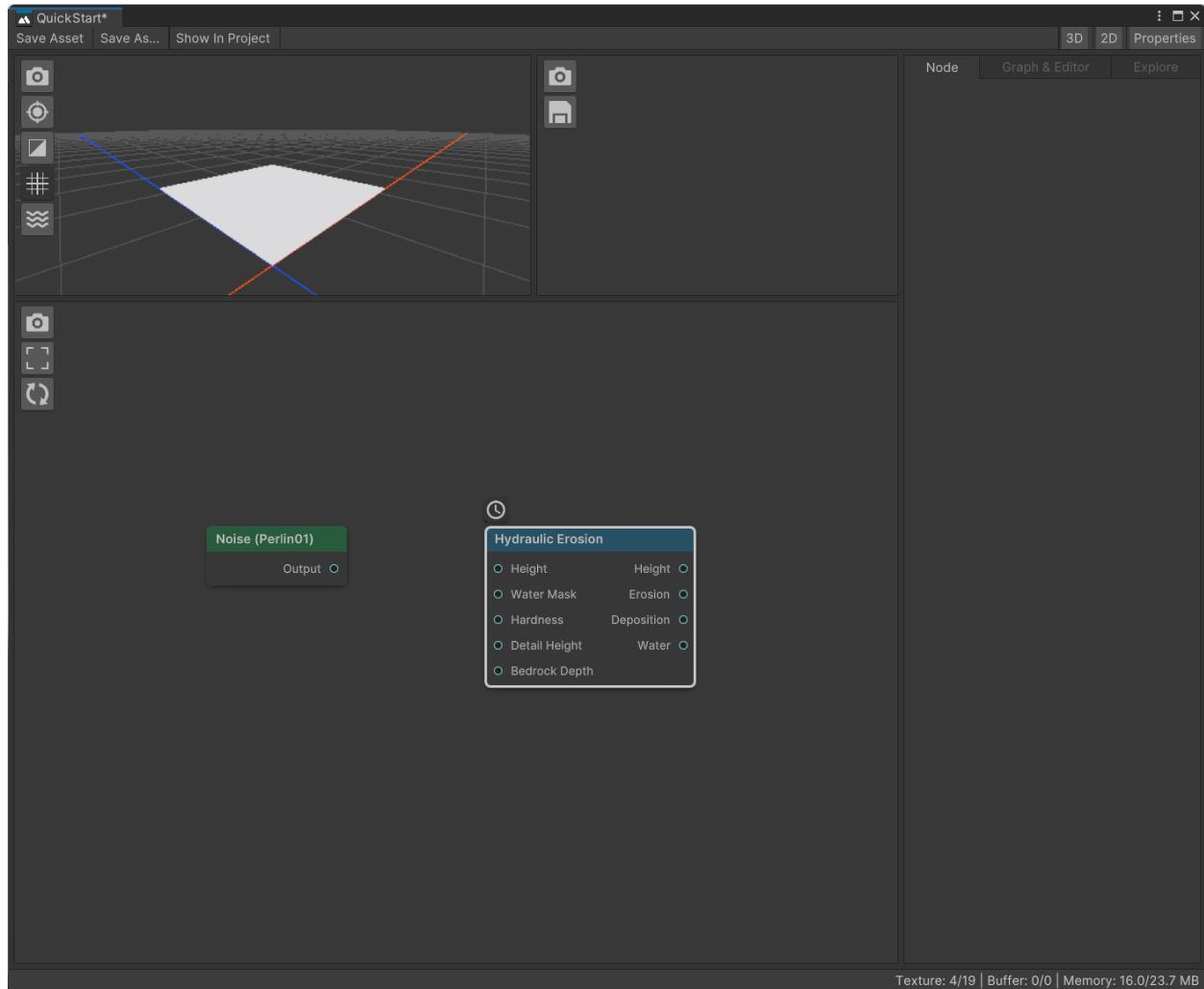


Terraforming

In the Graph View, press Spacebar to bring up the Node Browser, then find and double click on the Noise entry to add a Noise node:

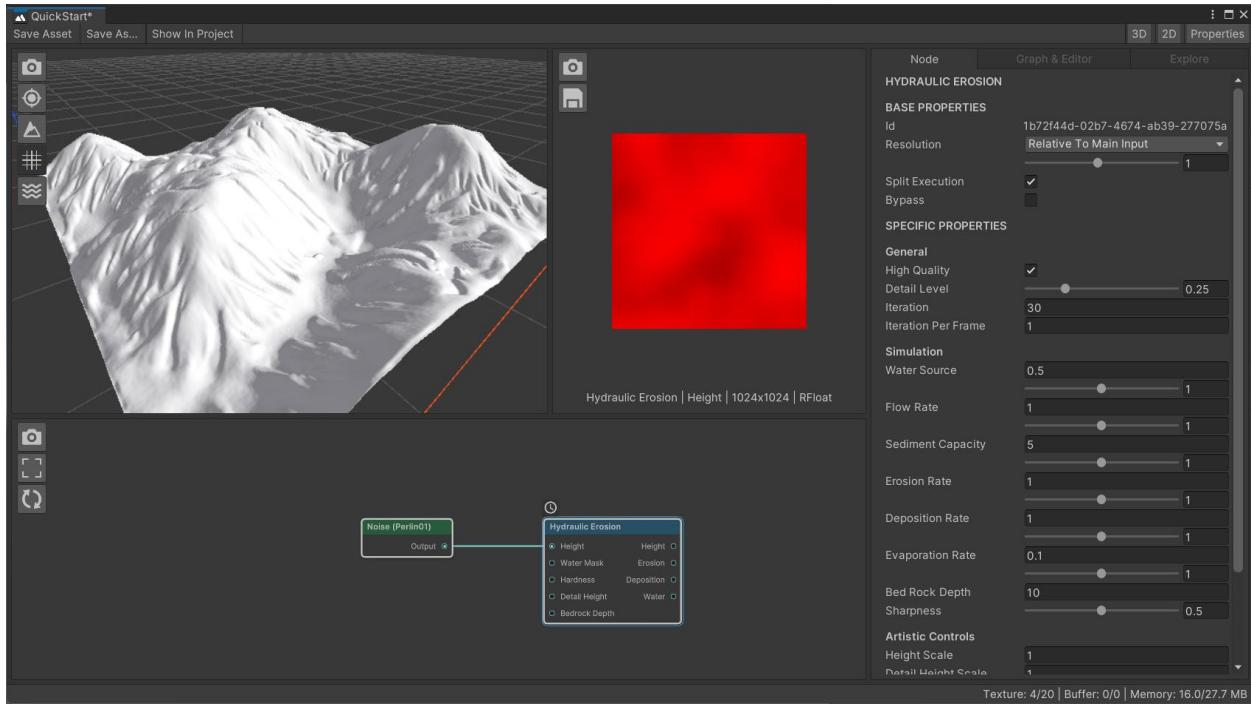


Then, try creating a Hydraulic Erosion node, like this:



To connect 2 nodes, drag from a port of the first node and drop on a port of the second node.

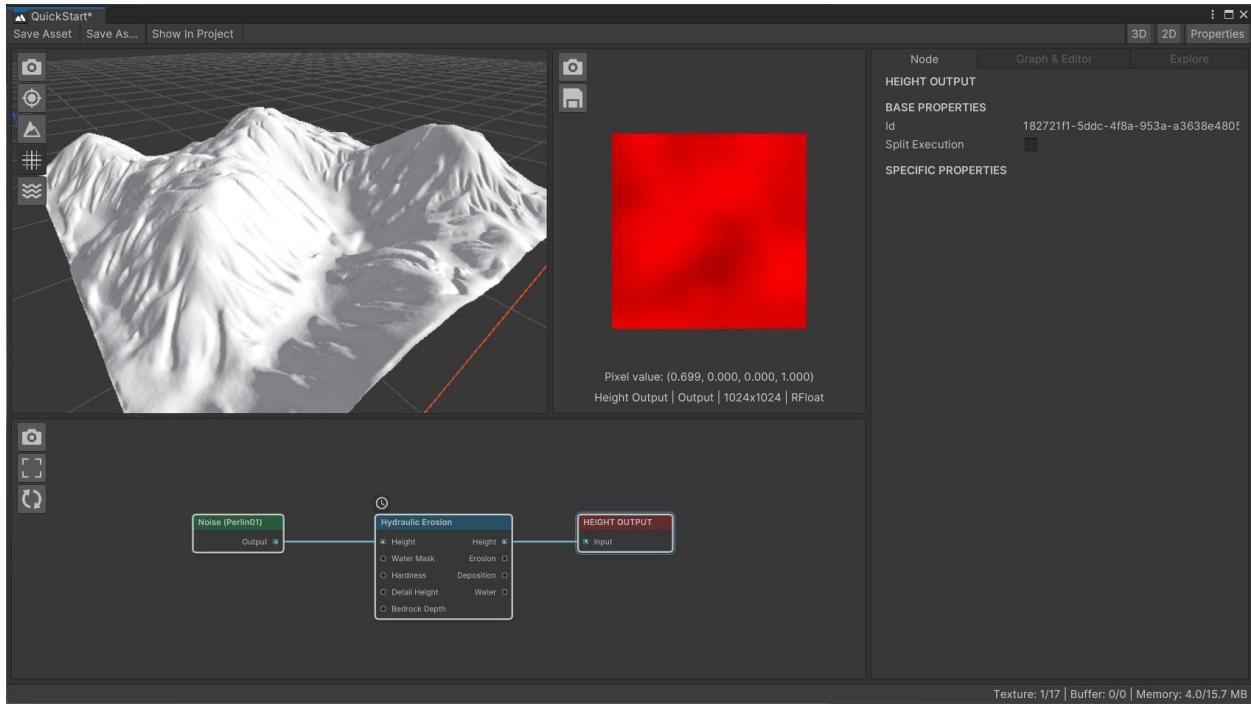
You will see the 3D/2D Viewport updates when selecting a node:



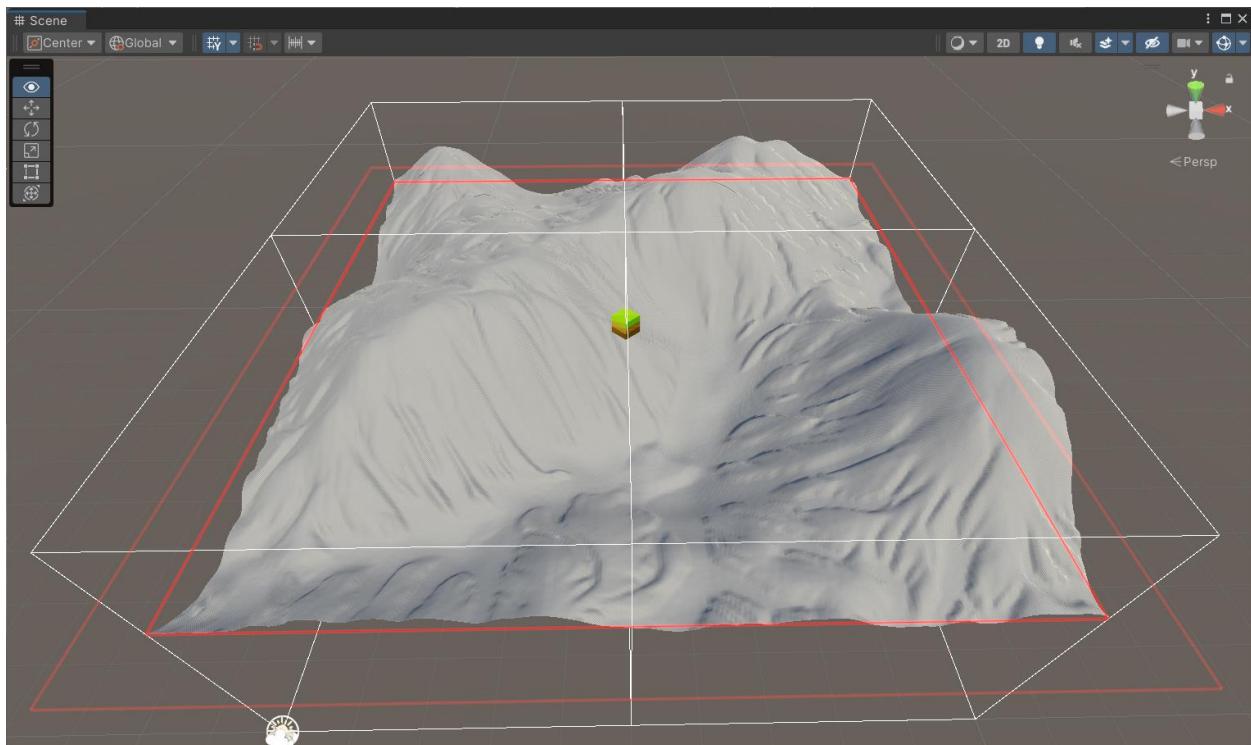
If you don't see the mountain shape appearing in the 3D viewport as the picture above, try clicking on the "Default texture display mode" button on the left of the 3D viewport.

Now, play around with the node properties in the Properties View to see their effects.

To output height map to the terrains in the scene, you need to create and connect a Height Output node, like this:



Hit Save Asset on the top left corner of the editor, and go back to the Scene View:

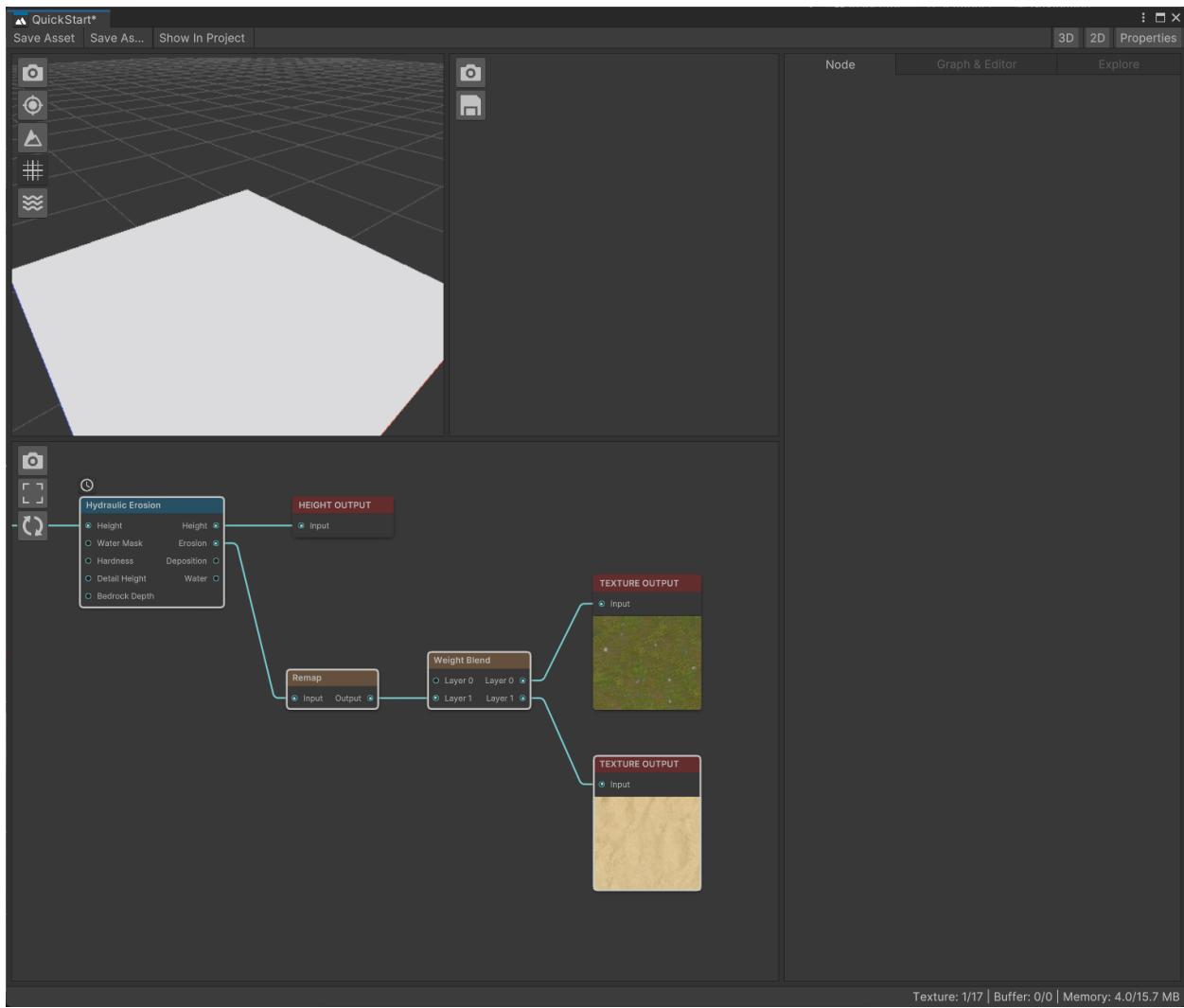


Good job! Now you have a pretty nice scene with eroded terrains.

Texturing

We will do a very quick texturing to visualize the erosion effect on the terrains.

Setup the graph like this:

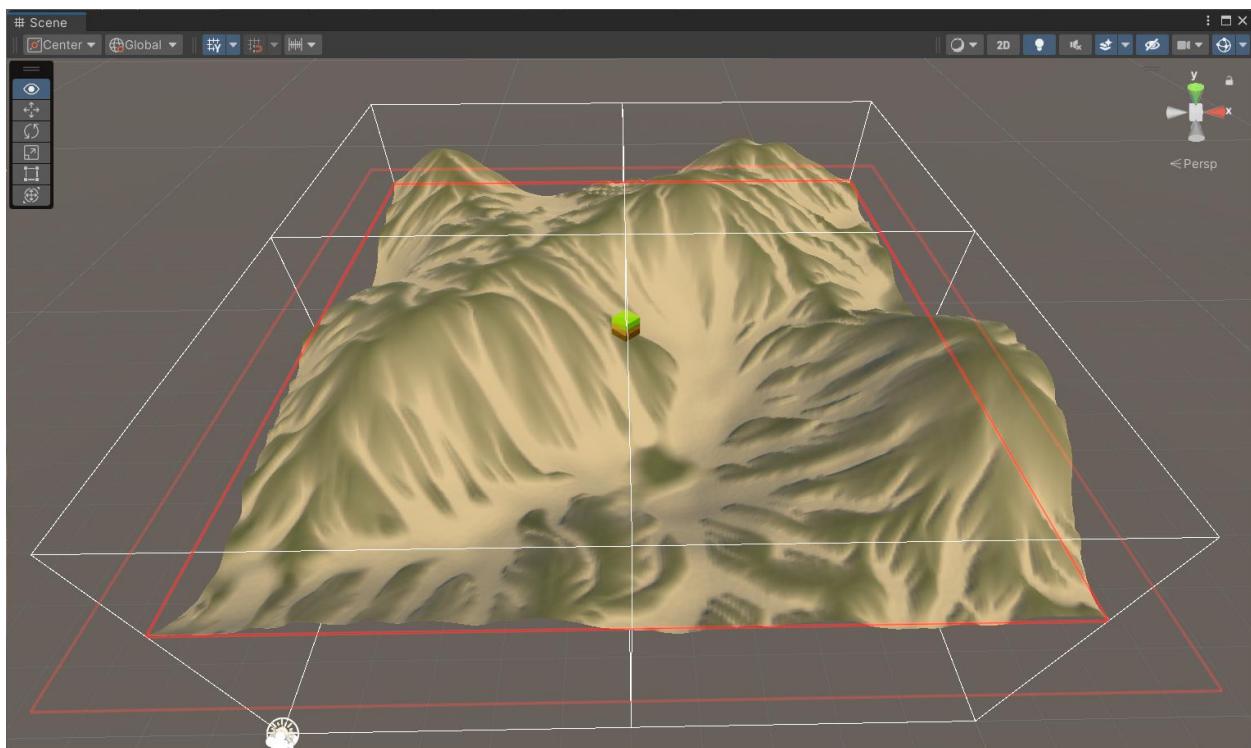


To output texture weight to the terrain, you need to create and connect a Texture Output node. Terrain textures are blended additively, so it's better to use a Weight Blend node to amplify the layers.

Select the first Texture Output Node, then assign the VistaSample_Grass terrain layer to it in the Properties View.

Select the second Texture Output Node, then assign the VistaSample_Sand terrain layer to it in the Properties View.

Hit Save Asset and go back to the Scene View.



That looks nice!

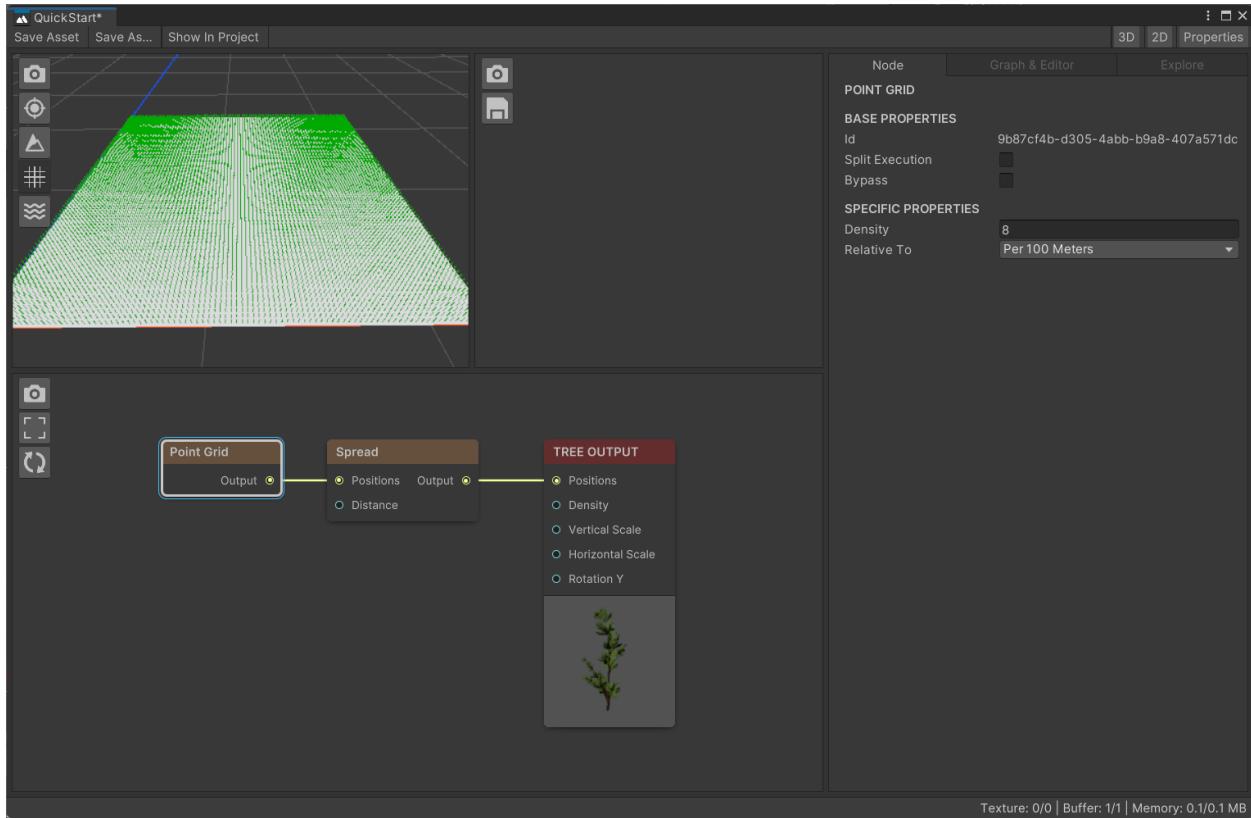
Vegetation planting

To be able to plant trees and details on the terrain, you have to create some Tree Template and Detail Template assets. We have prepared some for you to get started at:

- Vista/Samples/FoliageTemplate/VistaSample_Tree.asset
- Vista/Samples/FoliageTemplate/VistaSample_GrassDetail.asset

To create your own asset, right click on the Project window, then select **Create>Vista>Tree/Detail Template**.

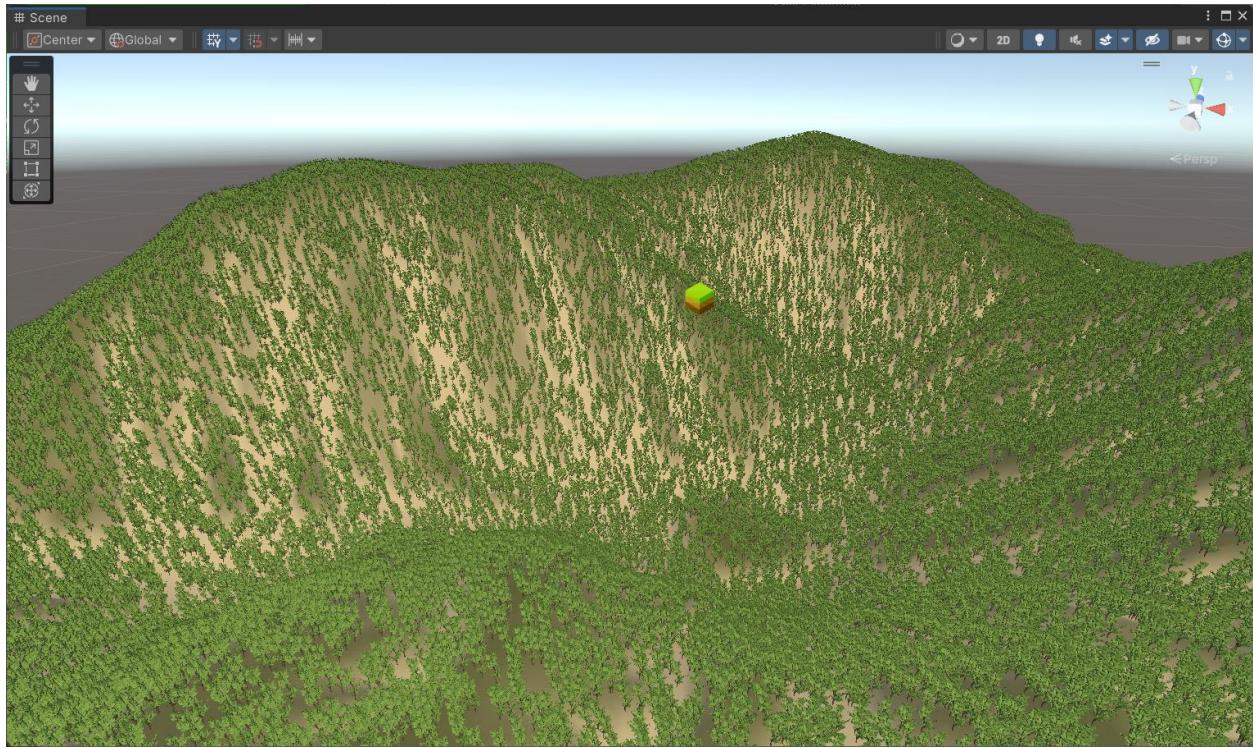
To plant trees on the terrain, you need to create and connect a Tree Output node, try a setup like this:



Select the Tree Output node, then assign the `VistaSample_Tree` asset to its slot in the Properties View. You should see the tree preview is drawn right below the node.

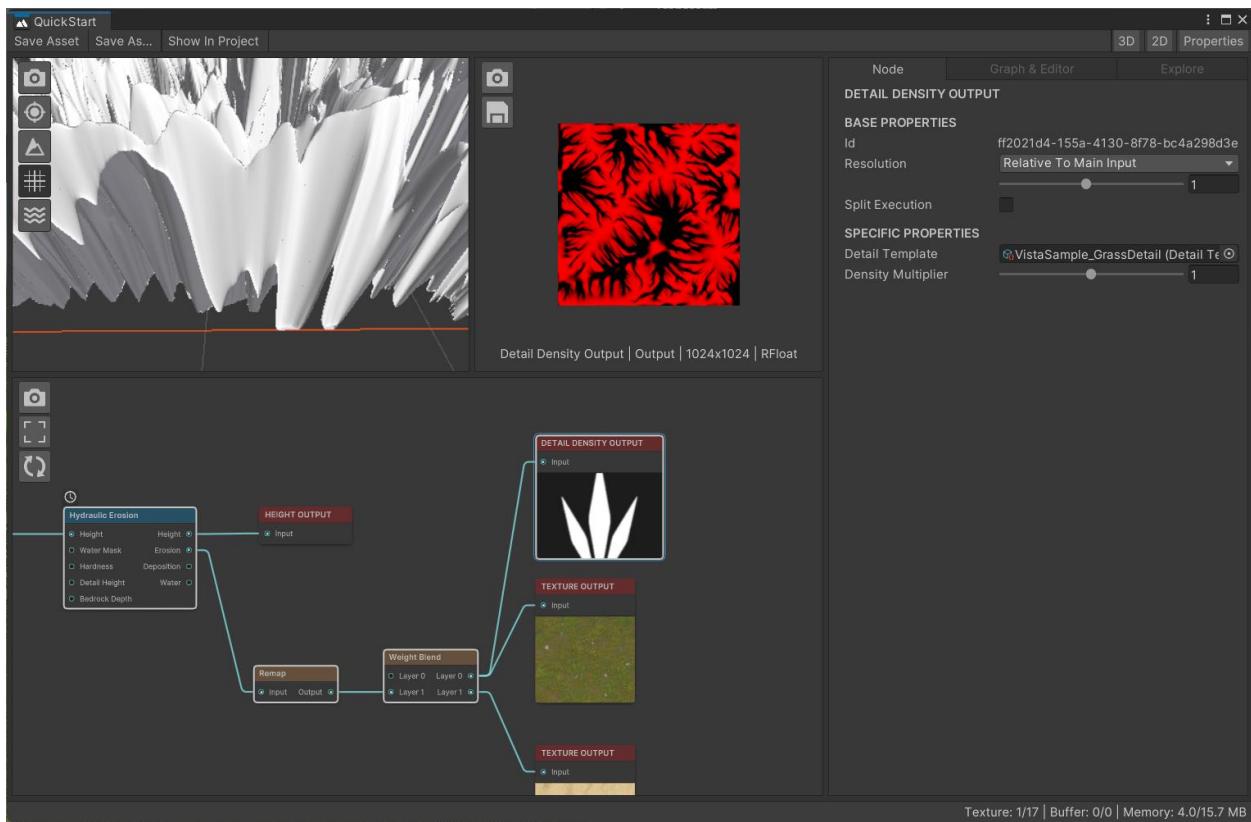
Select the Point Grid node, set its Density to the desired value to decide the number of trees, we will use 8 Per 100 Meters.

Hit Save Asset and go back to the Scene View.



Remember to tweak your terrain settings to see trees in the distance. Just ignore the editor performance for now.

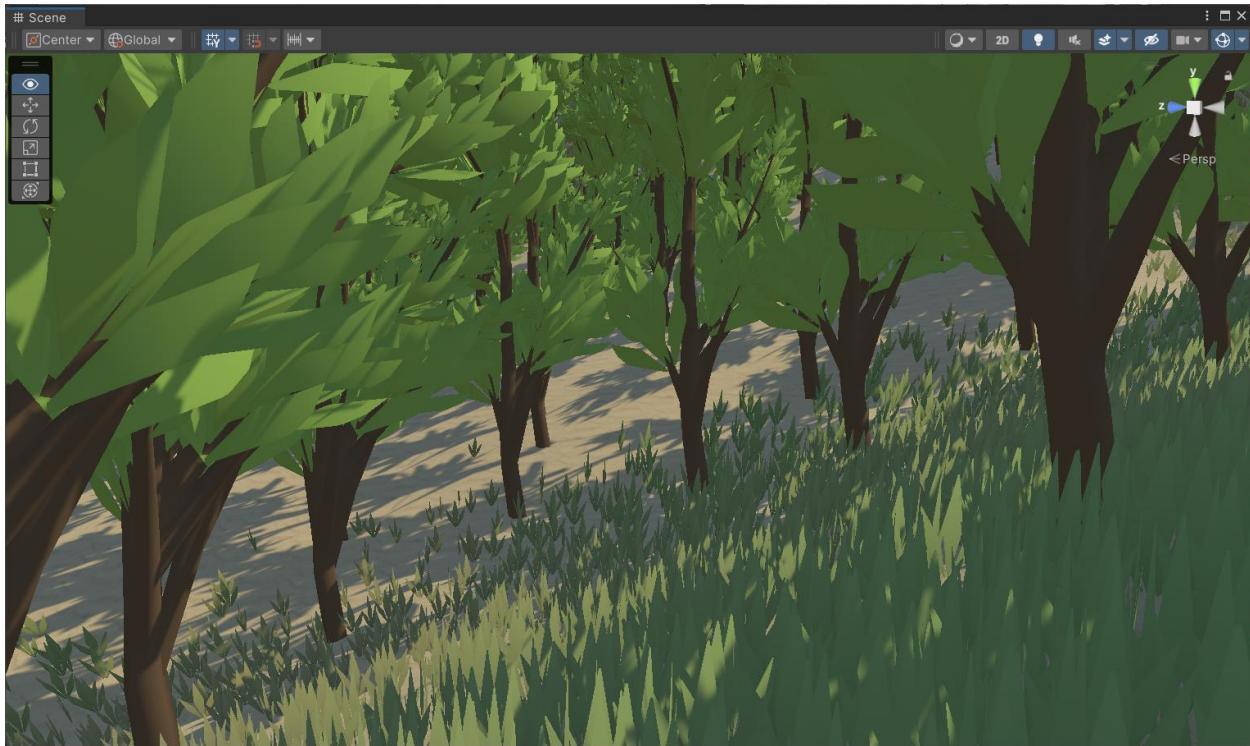
To plant grasses & detail objects, you need to create and connect a Detail Density Output node. Like this:



Select the Detail Density Output node, then assign the VistaSample_GrassDetail asset to its slot in the Properties View, then you should see the grass texture displayed right under the node.

The setup above will scatter grass on the terrain where the terrain texture is mostly green.

Hit Save Asset and go back to the Scene View:



Note: For Polaris terrain, the Detail Density Output node takes no effect since the system doesn't use detail density maps. You have to use the Detail Instances Output node instead.

Game Object spawning

To be able to spawn game objects on the terrain, you have to create an Object Template asset.

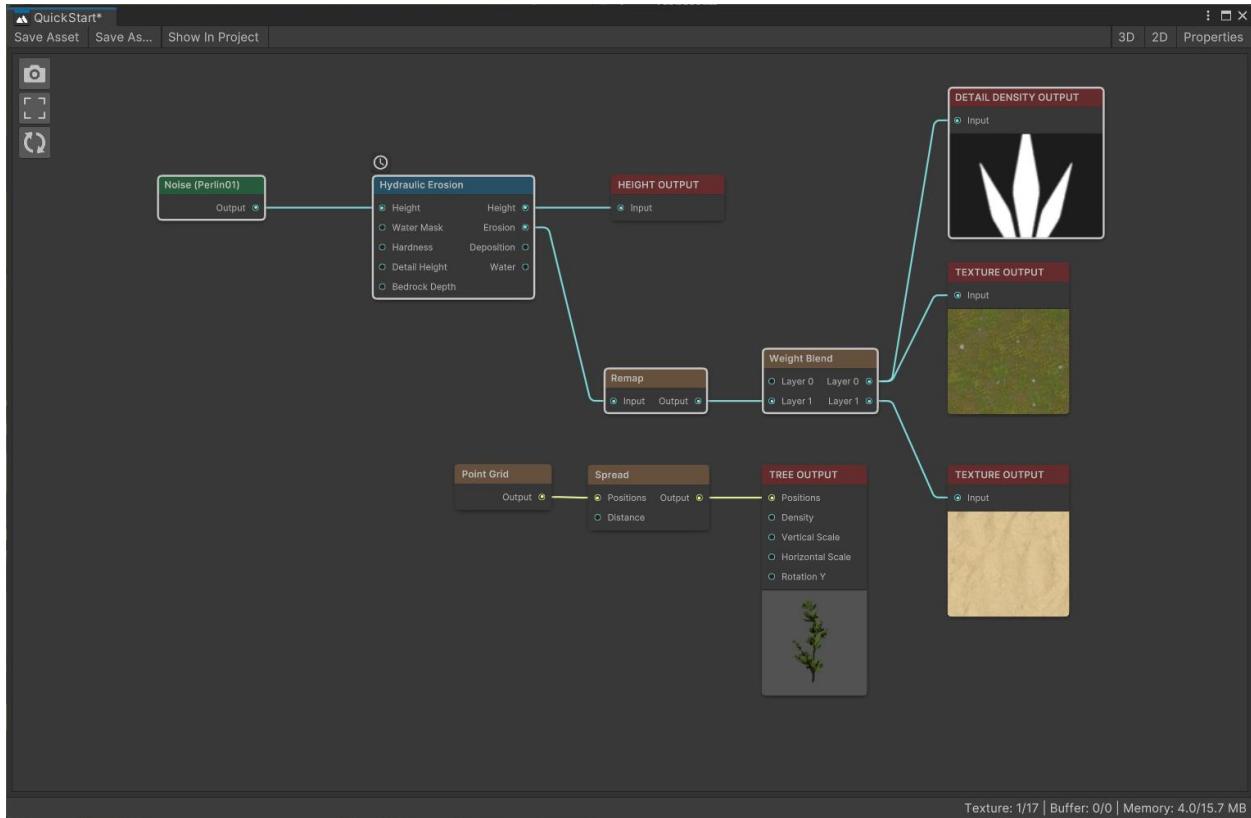
Right click on the Project window, then select Create>Vista>Object Template.

In the graph, create and connect an Object Output node. The setup is pretty similar to tree planting.

We will leave this for you to explore. Have fun!

★ Achievement unlocked - The first graph ★

Congratulations! You've created a great landscape with such a simple graph.



Now try playing around with the graph, add more simulations, more textures and vegetation species to it.

There is also a demo scene located under **Assets/PinwheelStudio/Vista/Samples/Scenes**, you can play around with it.

Ready to do more? There is a [Big World Module](#) that offers more features to create bigger worlds:

- Up to 4K texture resolution.
- Multi-biomes workflow.
- Extended node library with more advanced nodes & simulations.

TERRAIN MANAGEMENT

Vista aims to support all height field terrain systems, including Unity Terrain and [Polaris Terrain](#), and other systems which can utilize textures and data arrays.

Each system has its own specific settings and workflow, so we need to keep the number of properties Vista can control to a minimum, they are:

- Max Height: the maximum height level of a terrain.
- Height Map Resolution: the size in pixels of the texture containing elevation data.
- Texture Resolution: the size in pixel of the textures containing color/rendering data, it can be the splat alpha maps, albedo map, metallic map, etc.
- Detail Density Resolution: the size in pixels of the textures containing grass/details density. Polaris uses an instance array instead so this value should take no effect.

Due to the constraints above, we don't provide a unify terrain creation and management tool at the moment. Instead, we recommend using specific solutions for each system.

For Unity Terrain:

- The [Terrain Tools](#) package (com.unity.terrain-tools).

For Polaris Terrain:

- The [Wizard](#) and the [Group Tool](#).

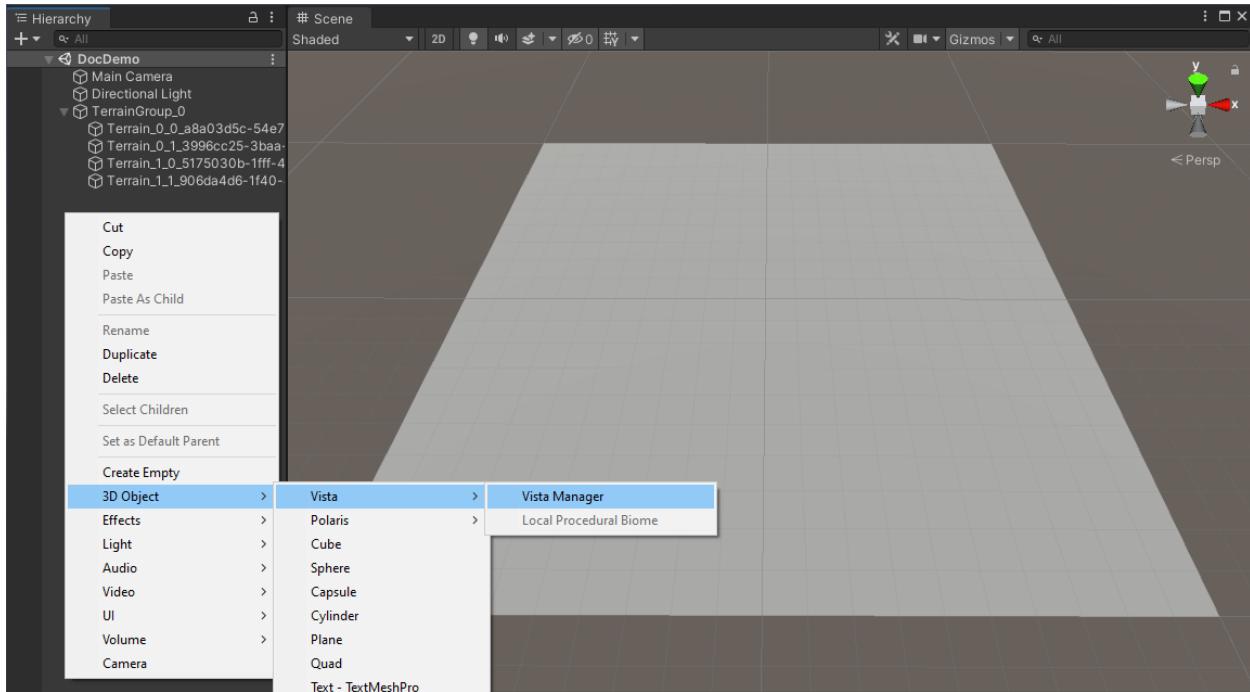
When changing terrain properties, just ignore the ones listed above, as they're going to be overridden during the generation pipeline.

VISTA MANAGER

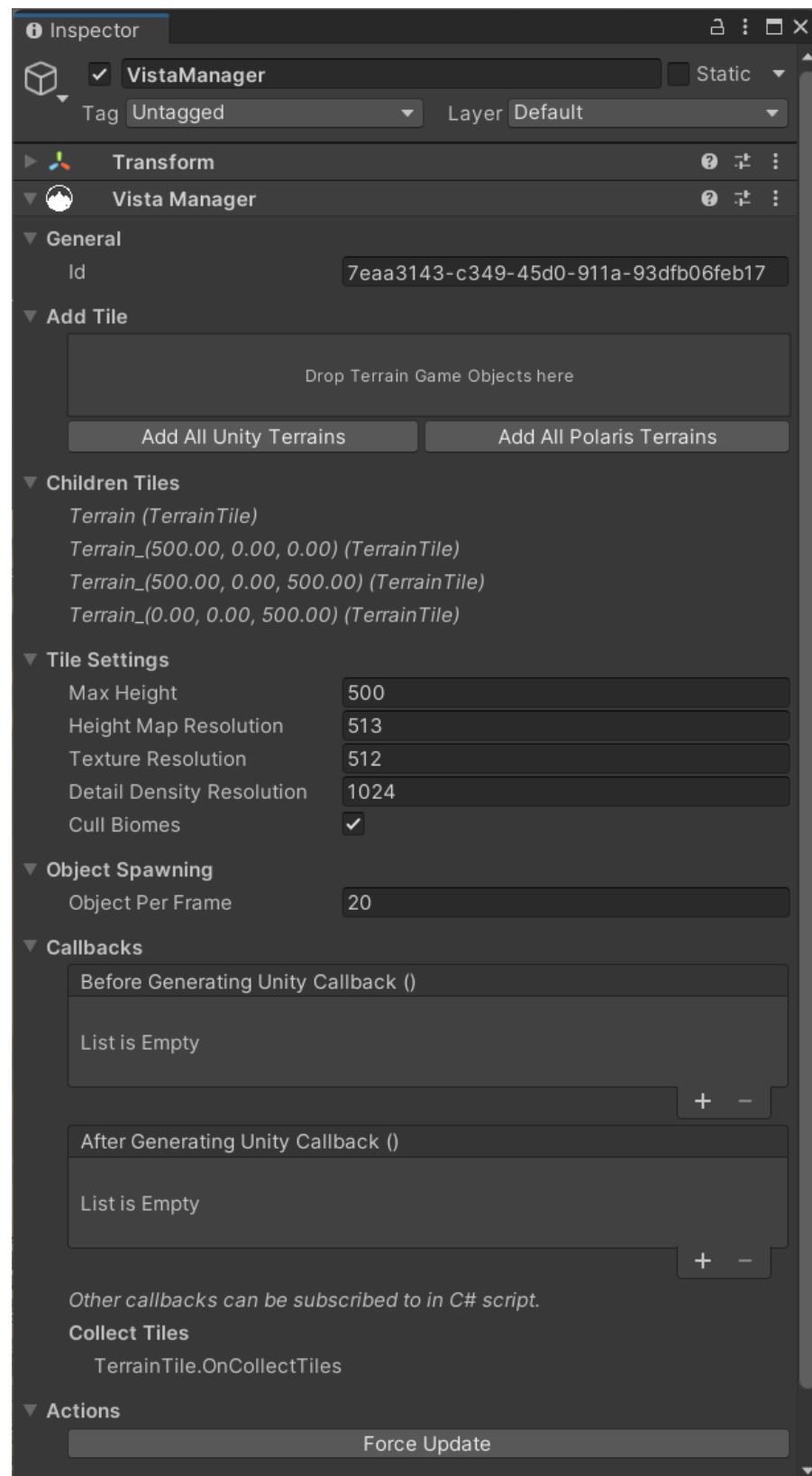
Component overview

Vista Manager (VM) is the component that controls the terrain generation pipeline. There should be at least 01 VM instances in the scene.

To create a VM instance, right click on the Hierarchy, then select **3D Object>Vista>Vista Manager**.



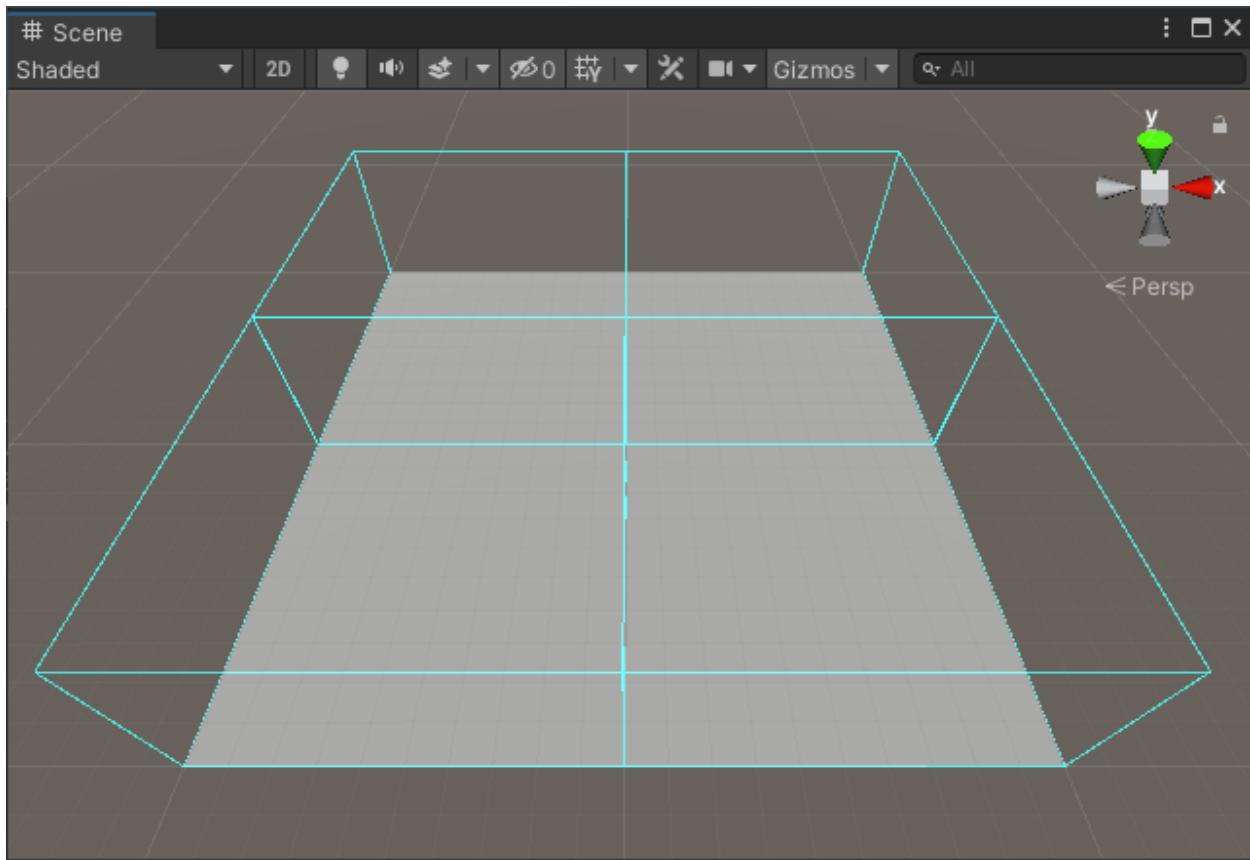
The Vista Manager holds a list of “tiles” which are under its control. A tile is simply a terrain instance of any terrain system, it doesn’t need to be a child game object of the Vista Manager, but can be anywhere in the Hierarchy.



To assign tiles to the Vista Manager, drag the terrain game objects from the Hierarchy and drop them to the “Add Tile” box in the Vista Manager Inspector, or simply click the “Add All Unity/Polaris Terrains” button.

Under the Children Tiles section, you will see a list of terrain objects associated with this VM instance. Clicking on the object’s name will highlight it in the Hierarchy.

Children tiles of a VM instance will be highlighted with boxes gizmos in the Scene View:



Under the Resolution section, you will find the following properties:

- Max Height: the maximum height level of a terrain.
- Height Map Resolution: the size in pixels of the texture containing elevation data.
- Texture Resolution: the size in pixel of the textures containing color/rendering data, it can be the splat alpha maps, albedo map, metallic map, etc.
- Detail Density Resolution: the size in pixels of the textures containing grass/details density. Polaris uses an instance array instead so this value should take no effect.

- Cull Biomes: If on, a tile will only receive data from the overlapped biomes, otherwise it will receive data from all biomes in the group. Usually this option should be turned on for better performance, except when using the MicroSplat Integration as MicroSplat needs to pack terrain layers into texture arrays.

Note that changing the value of these properties won't take effect immediately. The VM instance will alter the value of its children tiles the next time you re-generate the scene.

Under Object Spawning section, you will find the following:

- Object Per Frame: the number of game objects to spawn each frame during the generation pipeline. This is useful for reducing the lag and hiccup of instantiating too many objects at once.

Under the Callbacks section, you will find additional info about pipeline callbacks invocation, useful when making a custom module.

Under Actions section:

- The Force Update button: This will hook up a complete re-generation of all biomes under this VM instance.

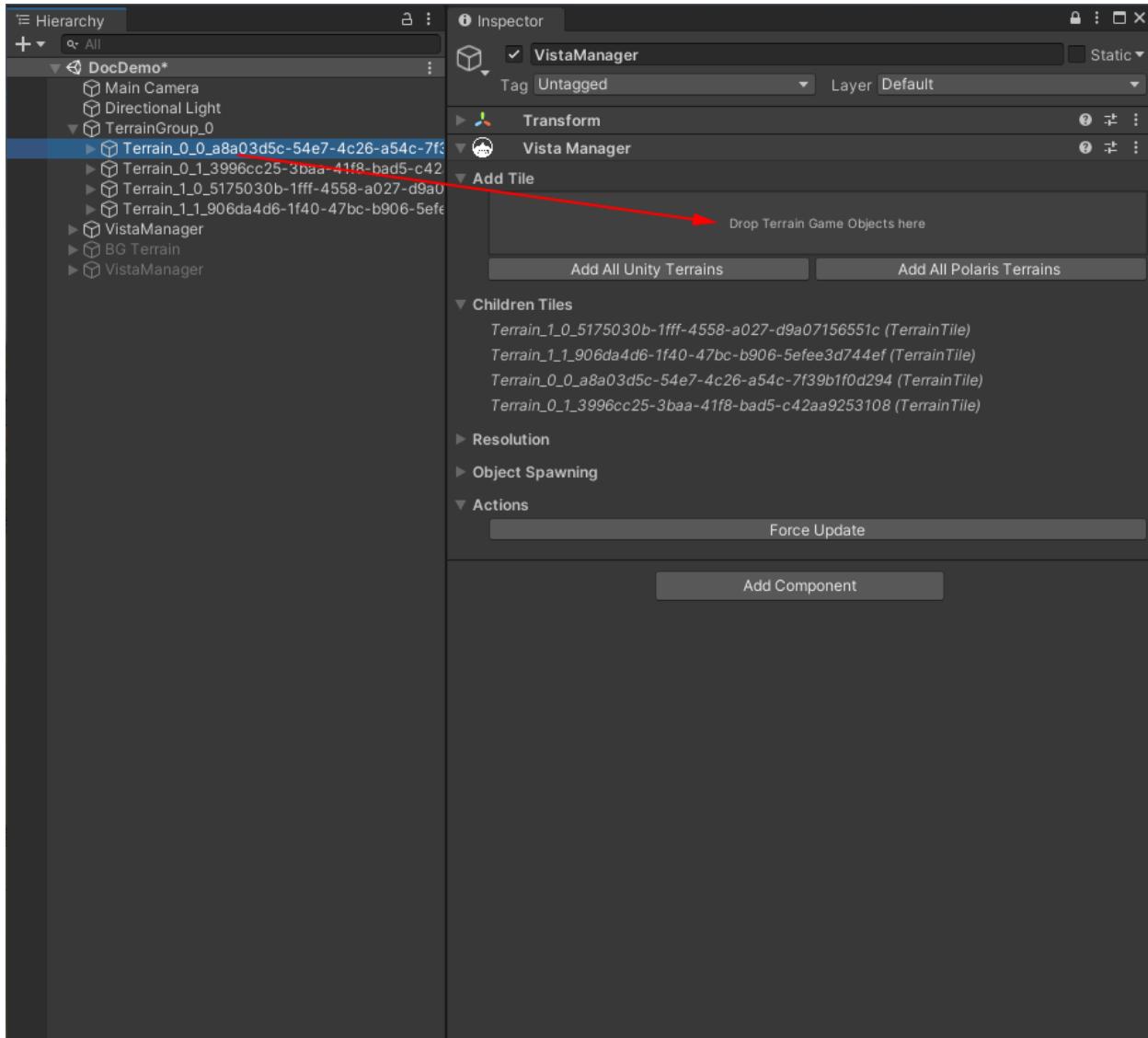
Children tiles

A "tile" is simply a game object with a terrain component attached.

For Unity terrain, the "terrain" component is Terrain.

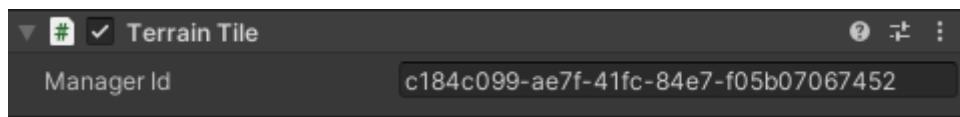
For Polaris terrain, the "terrain" component is GStylizedTerrain.

To add a game object to the VM children tiles list, drag and drop it from the Hierarchy to the gray box in the VM inspector.



To add children tiles in bulk, hit the “Add All Unity/Polaris Terrains” button.

When a game object was added to the VM, there will be some additional components automatically attached to it, such as:



These components will be used internally by the system, you don't need to worry about them much. Just to know they are there.

To remove a game object from the VM tiles list, you need to remove the “Terrain Tile” component (or similar) from it.

Note: A child tile of the VM instance can be anywhere in the scene. Its hierarchy doesn't matter.

Children biomes

Biome is the most fundamental component of Vista. It defines a region in the scene with specific characteristics such as geometry type, texture color as well as foliage species.

The VM instance holds a list of children biomes **as its children game objects**, those biomes are processed sequentially and blended together during the generation pipeline.

Each biome will have an order value assigned to it which decides its priority, higher value will be processed later in the biome queue. If many biomes share the same order, their hierarchy order will be considered (top to bottom).

To create a biome, right click on the VM instance in the Hierarchy, then use the 3D Object>Vista menus.

Important: The core asset only supports 1 biome as it aims for 1K to 2K terrain size where the scenery doesn't change much. If you create many biomes under the VM, it will take the first one in the hierarchy. To enable multi-biomes workflow for larger landscapes, consider getting the [Big World](#) module.

Note: A biome must be the child of a VM instance, otherwise it will be considered as “orphaned” and take no effect.

LOCAL PROCEDURAL BIOME

Component overview

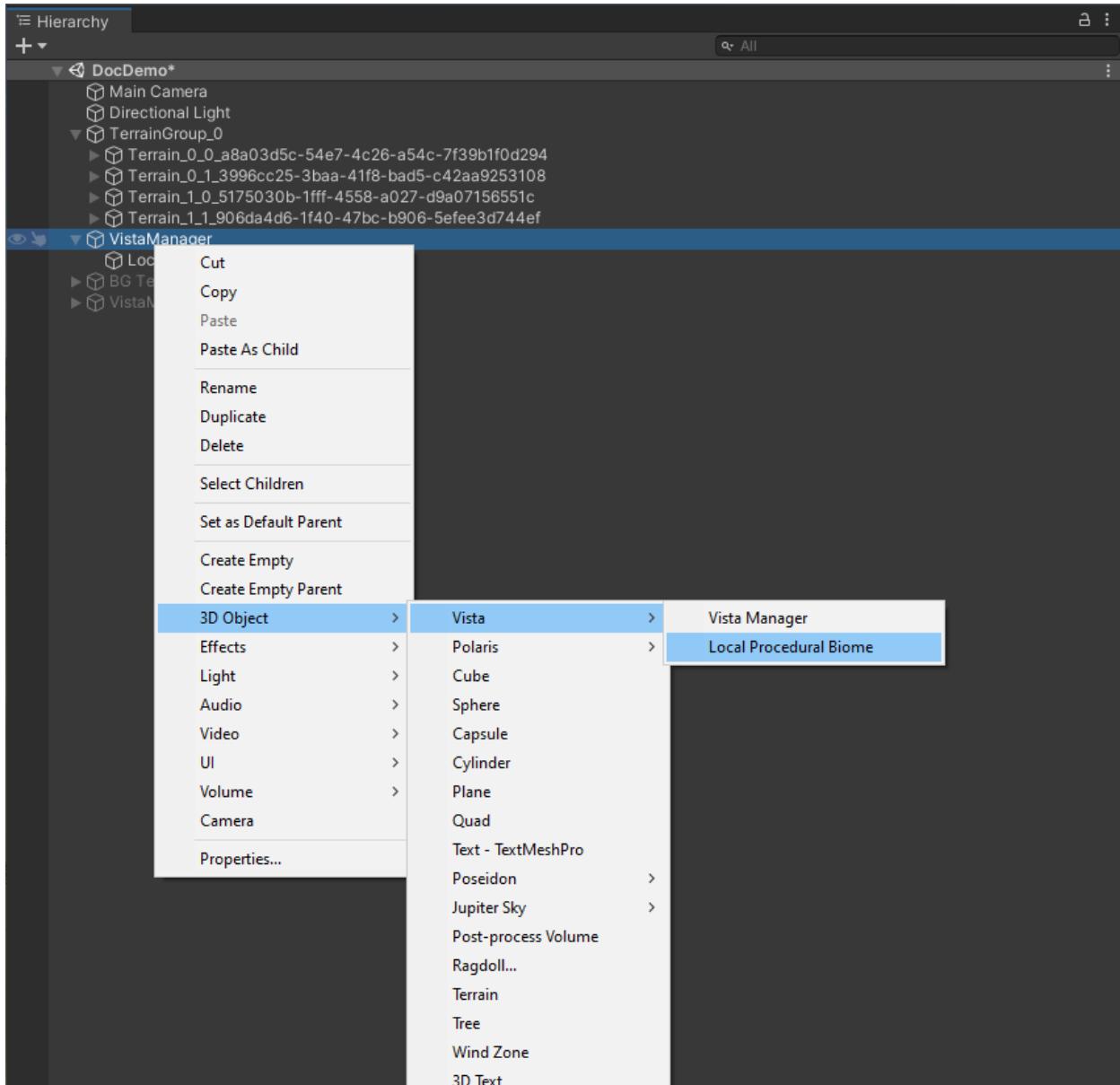
Local procedural biome (LPB) is a biome type with the following characteristics:

- Local: it defines a fixed polygonal region in the scene.
- Procedural: it uses graphs & parameters to generate its data.

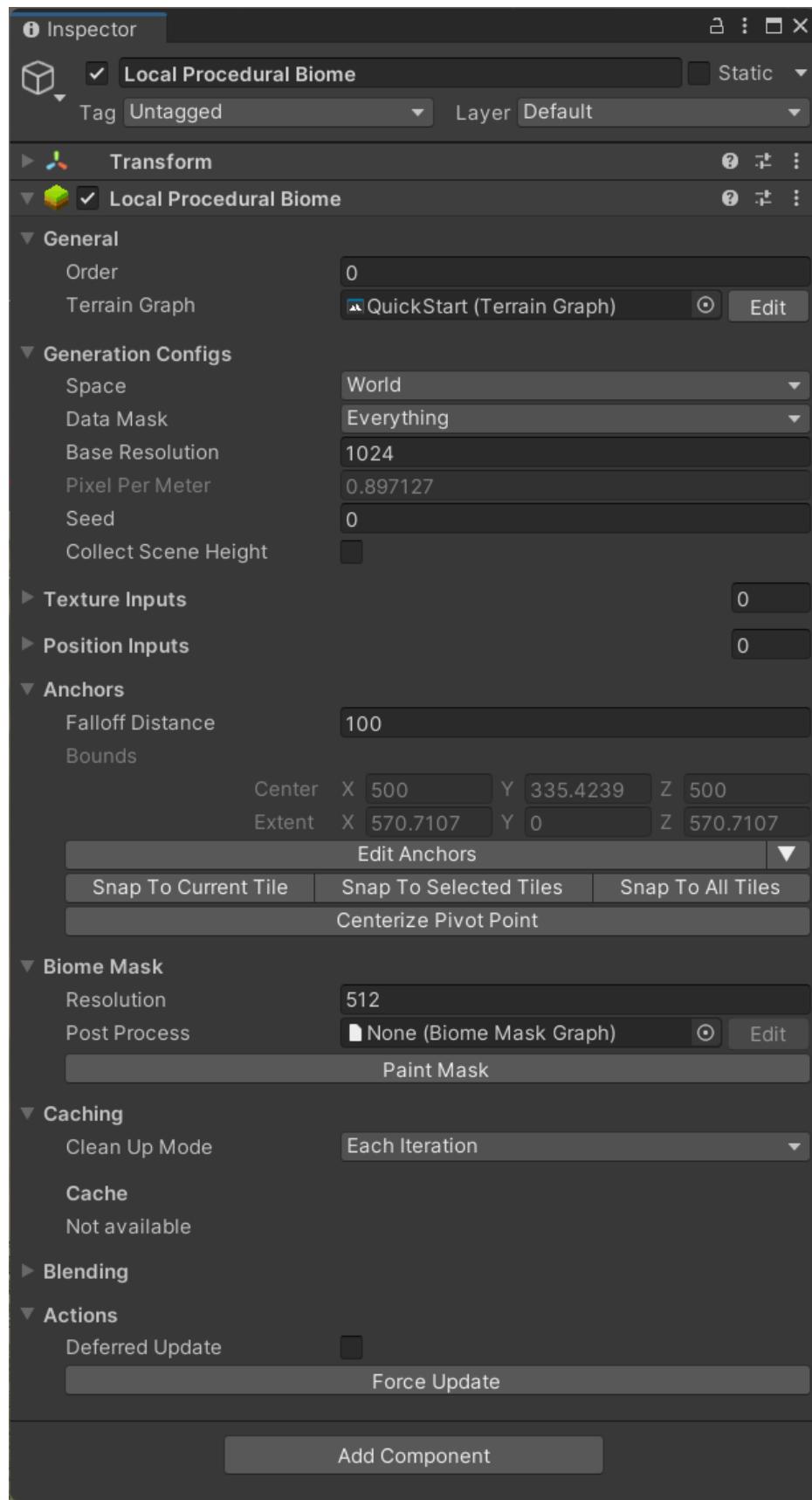
To create a LPB instance, right click on the VM in the Hierarchy then select **3D Object>Vista>Local Procedural Biome**.

Make sure the LPB instance is a child of the VM.

Important: The core asset only supports 1 biome as it aims for 1K to 2K terrain size where the scenery doesn't change much. If you create many biomes under the VM, it will take the first one in the hierarchy. To enable multi-biomes workflow for larger landscapes, consider getting the [Big World](#) module.



In the Inspector, you will see the following properties:



Under the General section:

- Order: processing order of this biome. Biomes with higher order will be processed later in the queue.
- Terrain Graph: the [terrain graph](#) asset contains rules for this biome. You can double click on the graph to open it, but it's better to use the Edit button, since it will feed some data into the graph editor.

Under the Generation Configs section:

- Space: whether to use Local or World space coordinates. If you want an almost identical result no matter where the biome is in the scene, select Local.
- Data Mask: Filter out the biome output where unnecessary data will be ignored. For example, if you uncheck the Height Map flag, the graph won't output height data even when you have added a Height Output Node.
- Base Resolution: the base/expected size in pixels of generated textures. The actual output may have different resolution since nodes can override this value. **The core asset limits this resolution to 1024 as it's already perfect for terrain from 1K to 2K meters in width and length.** To unlock the full power and have up to 4K textures, please install the [Big World](#) module.
- Pixel Per Meter: The number of pixels to cover 1 meter in world space, calculated based on the Base Resolution and the biome anchors. Higher value means higher quality but uses more VRAM..
- Seed: an integer to randomize the graph output. Different values will generate completely different outputs
- Collect Scene Height: Should it render a scene height map and feed to the graph as input? To access this texture, create a Graph Input node and use the "Scene Height" input name.

Under the Texture Inputs and Position Inputs section:

- These lists contain the Texture2D asset and [Position Container](#) asset that you want to feed into the graph as input, which in turn can be accessed using the Input node. Note that the name you set for each input here should match with the name in the Input node, it's also your responsibility to ensure input names are unique.

Under the Anchors section:

- Falloff Distance: size of the falloff area. Falloff area is where the biome data will gradually fade out and blend with underlying biomes.
- Bounds: the biome bounding box in world space.
- Edit Anchors: enable [anchor editing](#) in the Scene View.
- Snap To Current Tile: automatically fit the biome to the tile right at its position.
- Snap To Selected Tiles: automatically fit the biome to tiles in selection.
- Snap To All Tiles: automatically fit the biome to all tiles.
- Centerize Pivot Point: Move the biome's pivot point to the center of its anchors.

Under the Biome Mask section:

- Resolution: the size in pixels of the biome mask texture. This texture is generated using the biome's anchors and will be used for biomes blending.
- Post Process: a [biome mask graph](#) used for post processing the mask texture. You can use this graph to add more detail such as exponential fade, noise falloff effect to the mask for a more interesting transition between biomes.

Note: This section is only available with the [Big World](#) module installed as this graph type is only used for the biomes blending process.

Under Caching section:

- Clean Up Mode: Decide when to release the cache data. Choose Each Iteration when your graph is still in draft to ensure data is up to date. Choose Manually if you're working with another biome and don't want to re-generate this one every time. Note that the cache is forced to clean up on some event such as scene reloading, object deactivating, etc.
- Cache: Display the data from the last generation.

Note: This section is only available with the [Big World](#) module installed. If not, Clean Up Mode will fallback to Each Iteration.

Under Blending section:

- Height Map: Blend mode for the height map. Let A be the current blend result, B be the height map of this biome:
 - Linear: The height map will gradually change from A to B.
 - Addictive: B will be added to A.

- Subtractive: B will be subtracted from A.
- Min/Max: Take the min/max value of A and B.
- Textures: Blend mode for terrain textures, including alpha maps, albedo map, metallic map & generic textures. Only Linear mode is available in this version, which means the texture will gradually change from one to the other.
- Detail Density: Blend mode for detail density maps. You will use Linear or Addictive for most cases. Let A be the current blend result, B be the density map of a detail template in this biome:
 - Linear: The density map will gradually change from A to B, which means a grass type will win over the others and become dominant.
 - Addictive: B will be added to A, which means many grass species will coexist in the region.
 - Subtractive: B will be subtracted from A.
 - Min/Max: Take the min/max value of A and B.
- Instances: Blend mode for instance buffers, including trees, grass instances (Polaris), objects & generic buffers. Let A be the current blend result, B be the instances of a tree template in this biome:
 - Linear: The tree instances will gradually change from A to B, which means a tree type will win over the others and become dominant.
 - Addictive: B will be added to A, which means many tree species will coexist in the region.

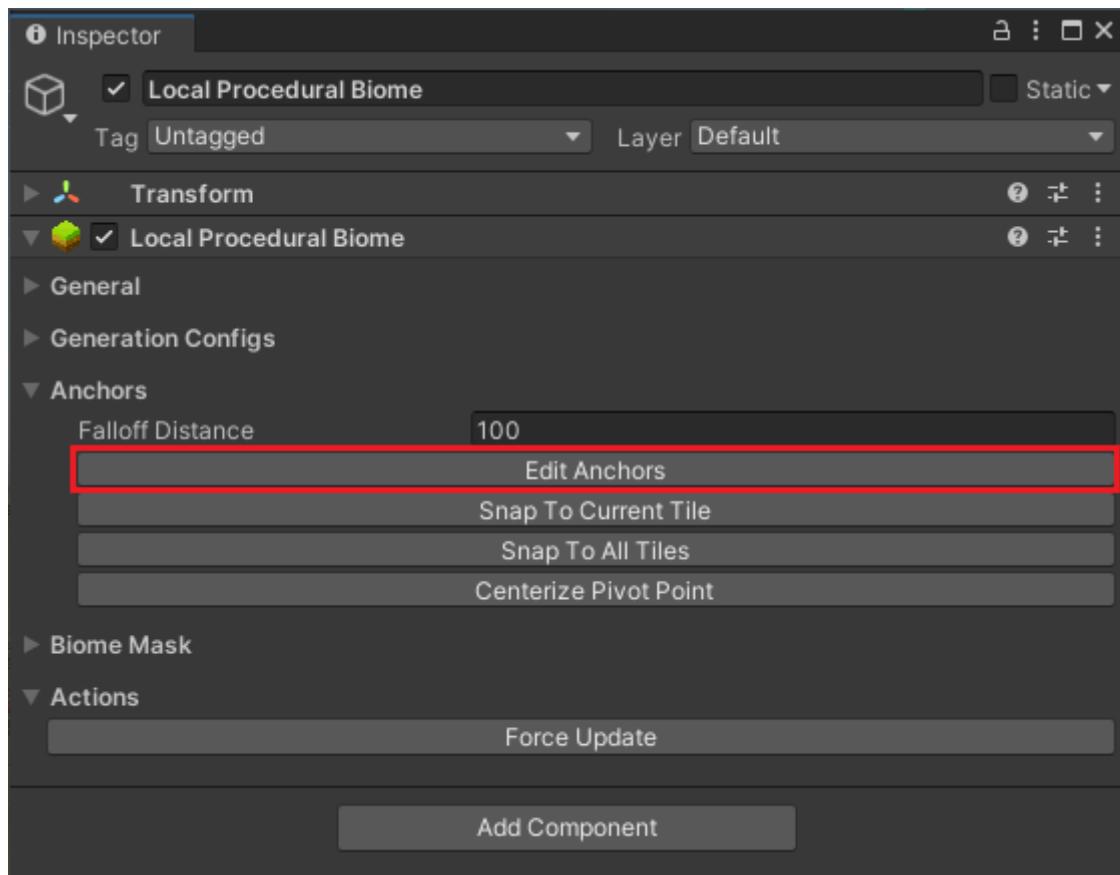
Note: The Blending section is only available if you have installed the [Big World](#) module because of the biome limit described in previous sections.

Under the Actions section:

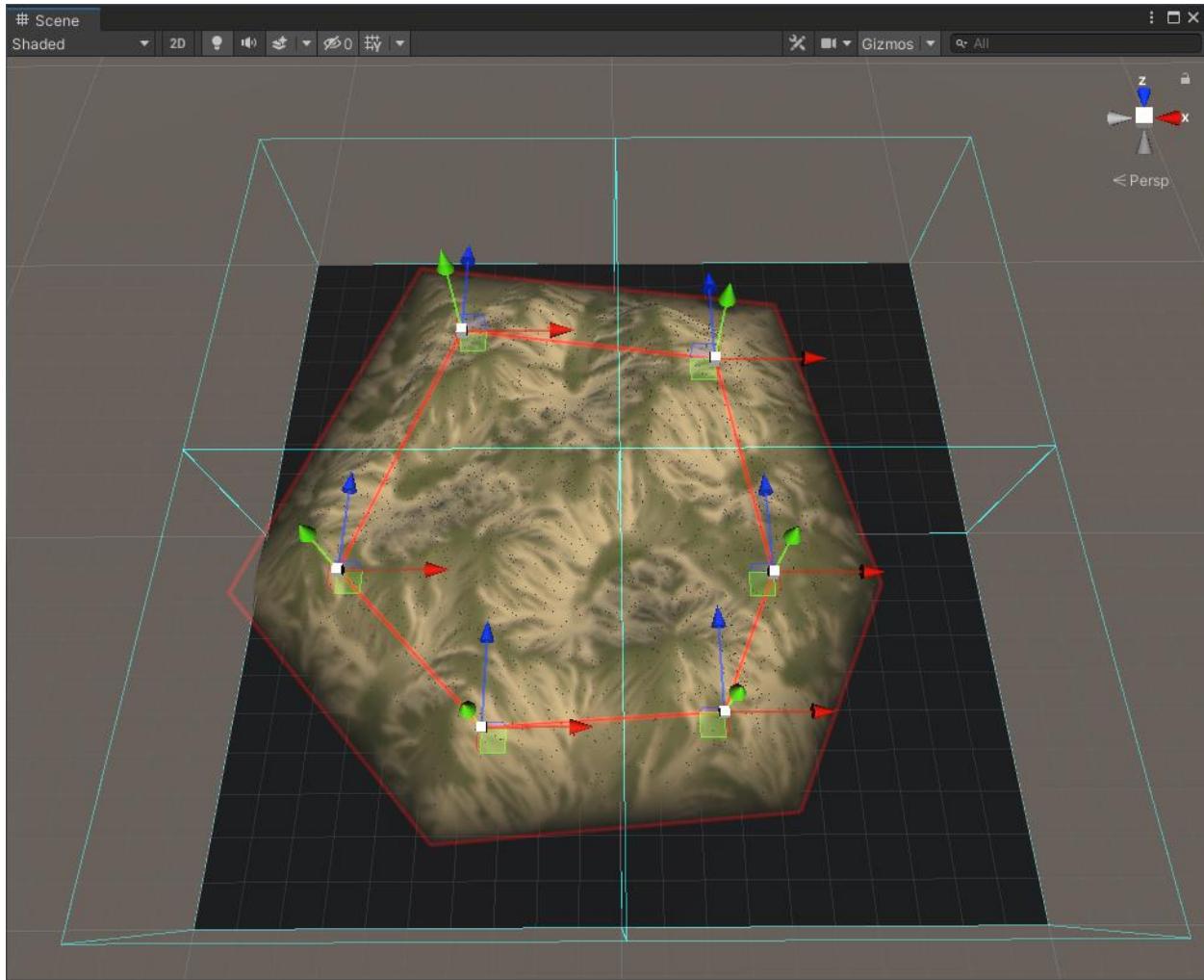
- Deferred Update: Turn this on if you want to make several changes to the biome before regenerating.
- Force Update: invoke a full re-generate pipeline on the VM instance this biome belongs to.

Anchors editing

To edit the boundaries of a LPB instance, click on the Edit **Anchor** button in the Inspector.



In the Scene View, you will see some Gizmos appear:



- Drag the arrows to move an anchor around.
- Shift Click to add a new anchor in between of the 2 nearest ones.
- Ctrl Click on an anchor (the white cube) to remove it.

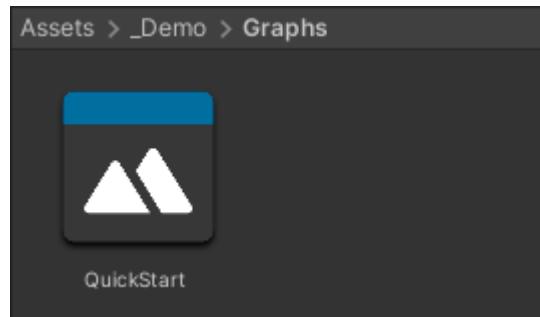
A biome will affect all tiles whose polygon overlaps, these tiles are highlighted with cyan boxes.

When you are happy with the shape, hit **End Editing Anchors** in the Inspector.

TERRAIN GRAPH

Terrain graph is an asset containing rules for the terrain generation, defined by the combination of many nodes and connections.

To create a terrain graph, right click on the Project window, then select **Create>Vista>Terrain Graph**.



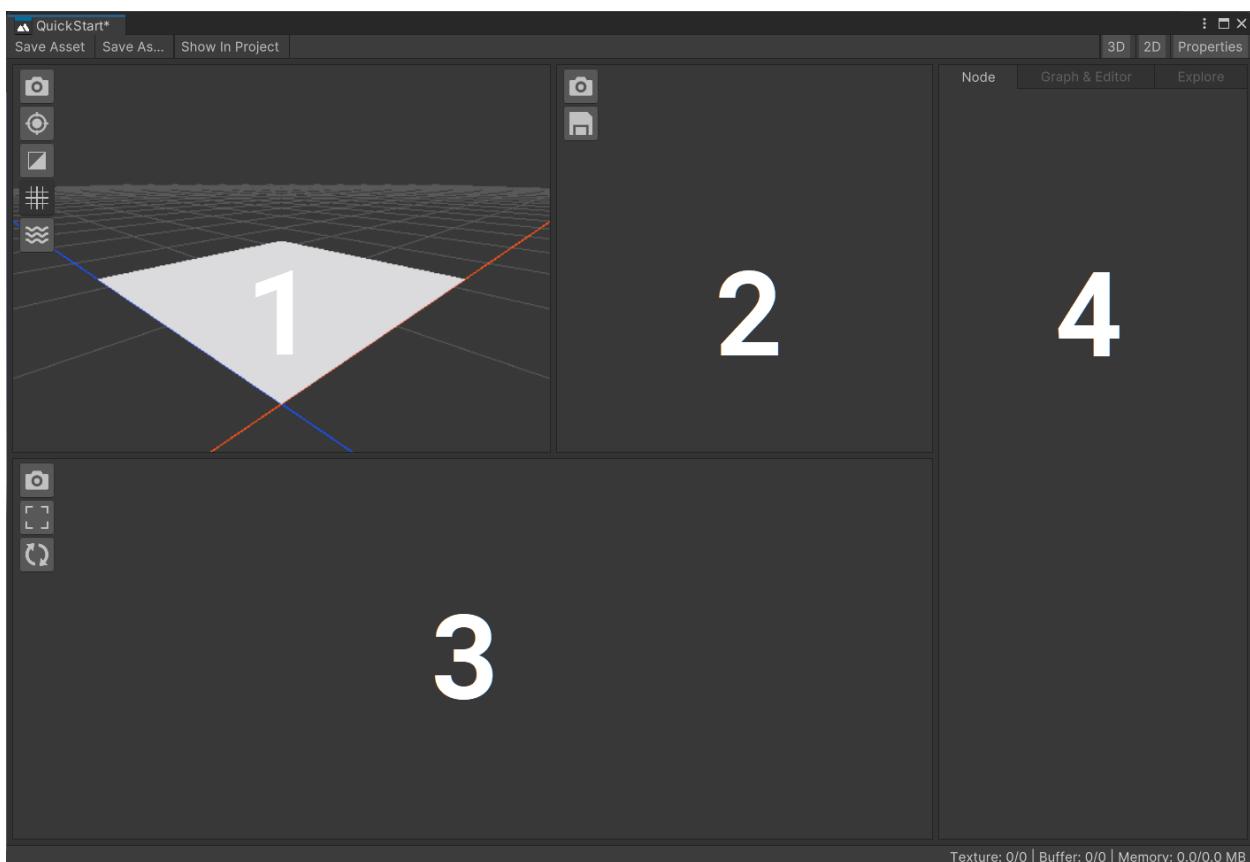
To edit the graph, double click on the graph asset. However, it's better to use the "Edit" button in your biome's Inspector to take advantage of contextual inputs.

GRAPH EDITOR

Overview

Graph editor is where you author Vista graph assets such as [terrain graph](#) and [biome mask graph](#).

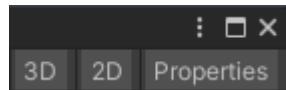
To open a graph editor, simply double click on a graph asset. However, it's better to use the "Edit" button next to the graph slot in your biome's Inspector, so you can have the correct contextual data such as world boundaries, resolution, seed, etc. to do the work easier.



In the graph editor, you can find:

- 3D Viewport (1): Display the preview terrain.
- 2D Viewport (2): Display the generated texture.
- Graph View (3): Display the graph.
- Properties View (4): Display the detail of the selected node as well as the graph itself.

You can show/hide these sections using the buttons on the top right corner:



You can also find that there is a Toolbar on top and Status Bar at the bottom of the editor.

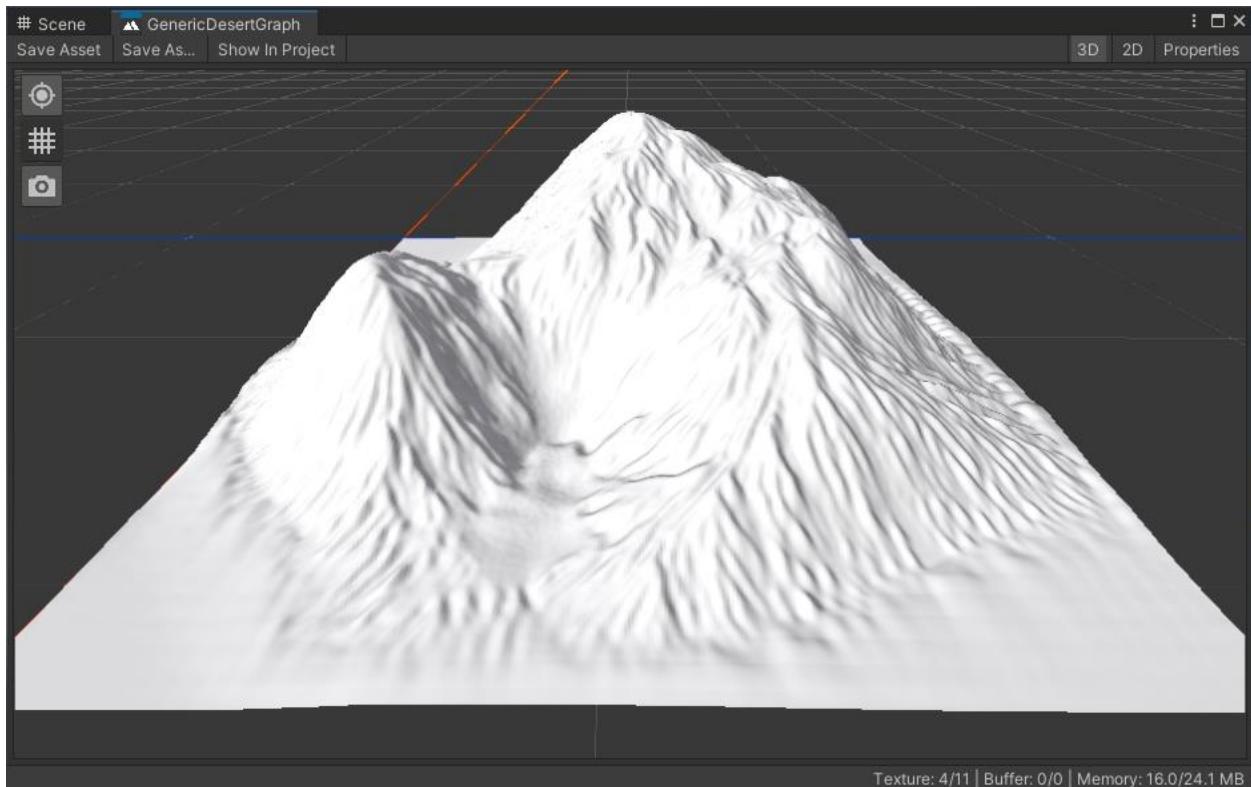
The Toolbar



The Toolbar contains some buttons for handling files and workspace.

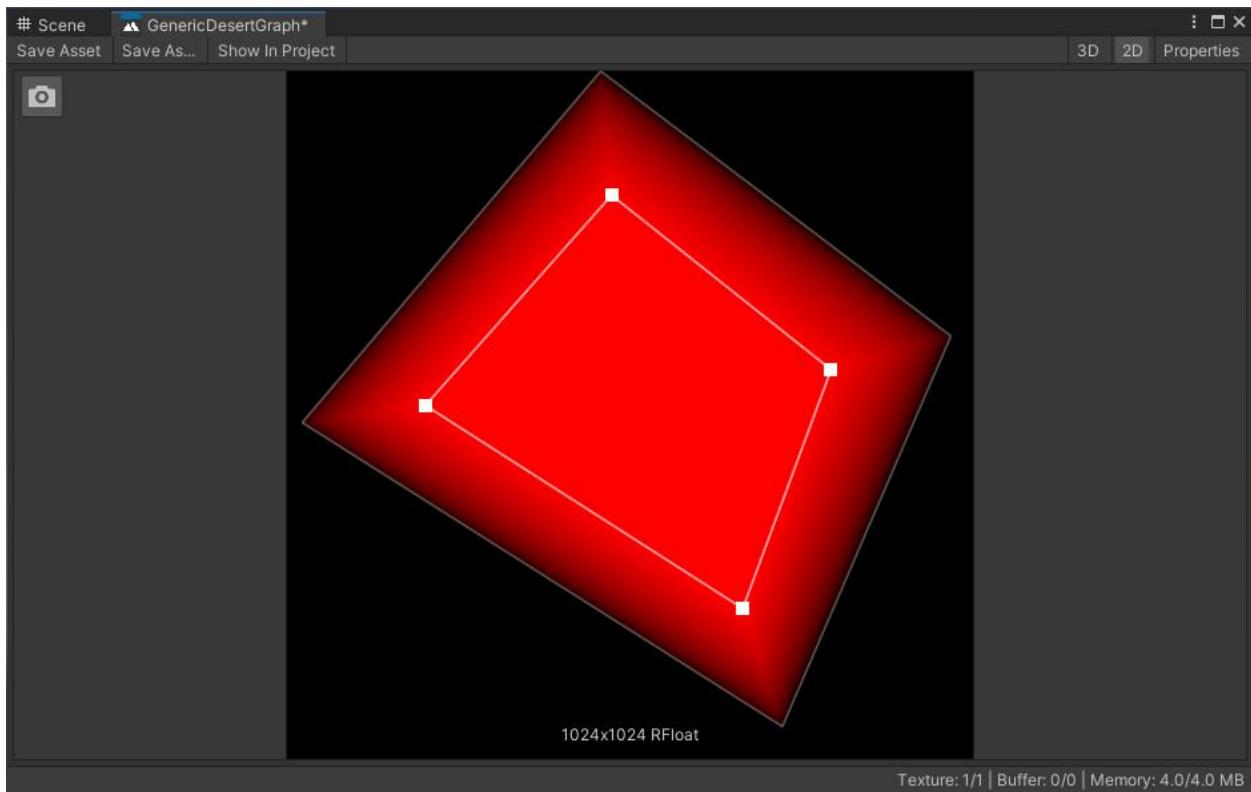
- Save Asset/ Save As...: save the graph to an existing or new asset.
- Show In Project: highlight the opening graph in the Project window.
- 2D/3D/Properties: show/hide the corresponding viewport.

The 3D Viewport



The 3D Viewport visualizes generated data in 3D space. You can navigate in the 3D Viewport the similar way to the Scene View.

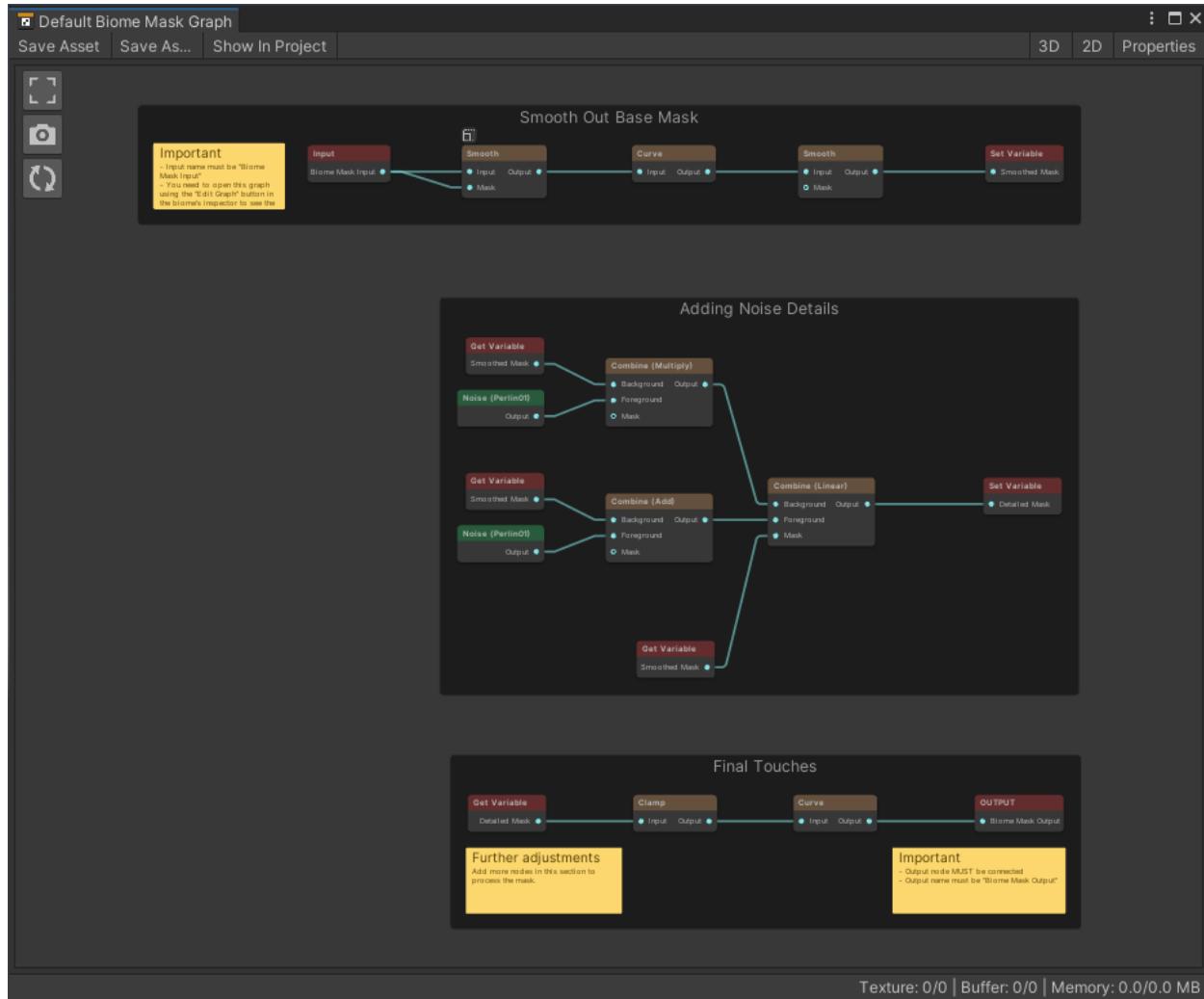
The 2D Viewport



The 2D Viewport visualizes generated data in 2D space. In some cases it can also be used for editing nodes.

The Graph View

The Graph View is where you create and connect nodes for your graph.



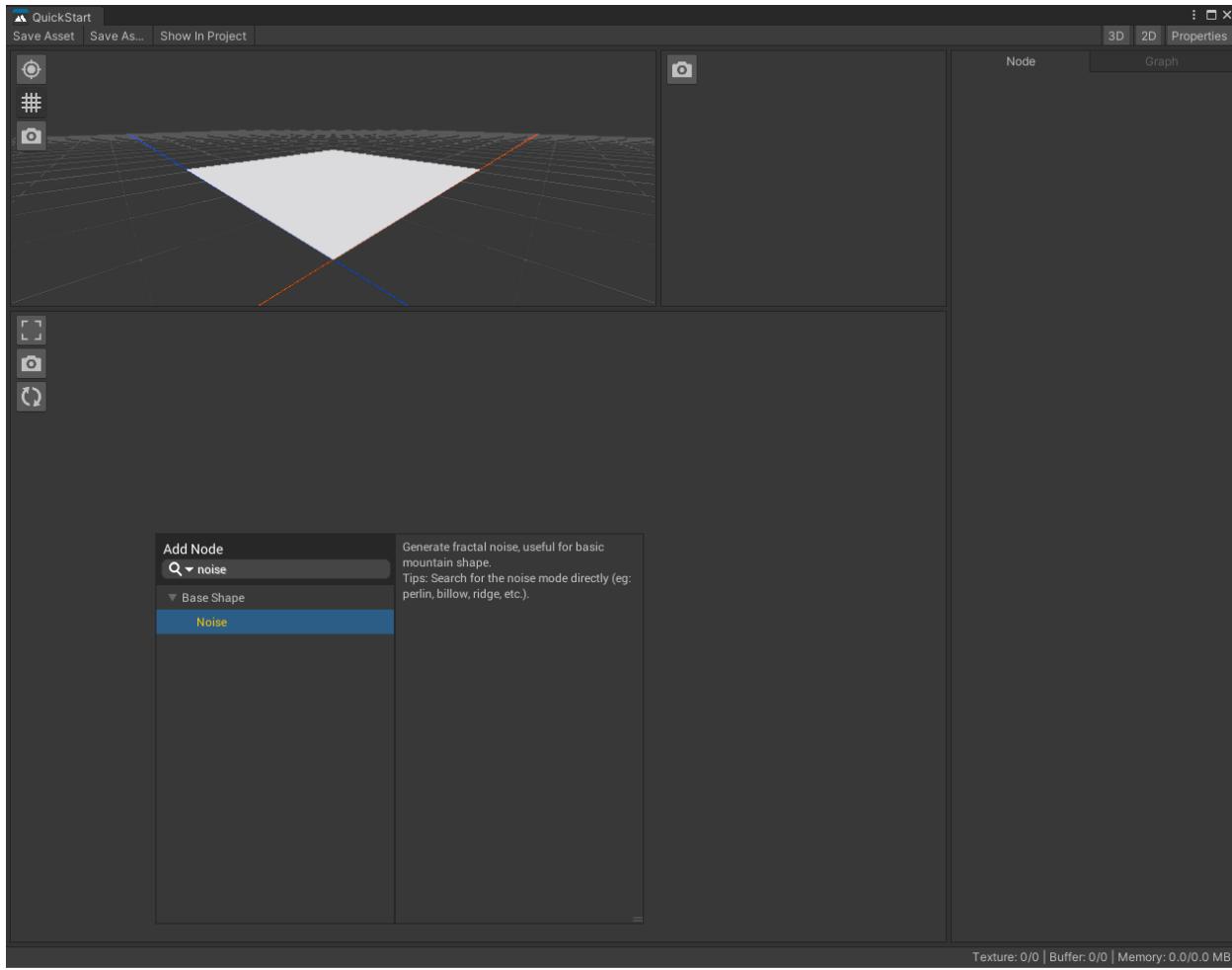
Basic interaction

To interact with the Graph View:

- Press & hold the middle mouse to pan.
- Scroll to zoom.
- Left mouse to select an element.
- Right mouse to show the context menu.

Adding new nodes

To add a new node, right clicking on a blank space to open up the context menu and select Add Node, or just press Spacebar, it will bring up the node selector:



In the node selector, type in the node you want to add, then press Enter or double clicking on its entry.

Connecting nodes

To connect 2 nodes, drag from a port of the first node and drop on a port of the second node.

Adding sticky notes

Sticky notes allow you to write some quick notes about the graph functionalities and designs.

To add a sticky note, right click on a blank space to open up the context menu, then select **Add Sticky Note**.

Grouping elements

To keep your graph tidy, it's better to gather the graph elements to a group that do a specific task such as terraforming, texturing, planting, etc.

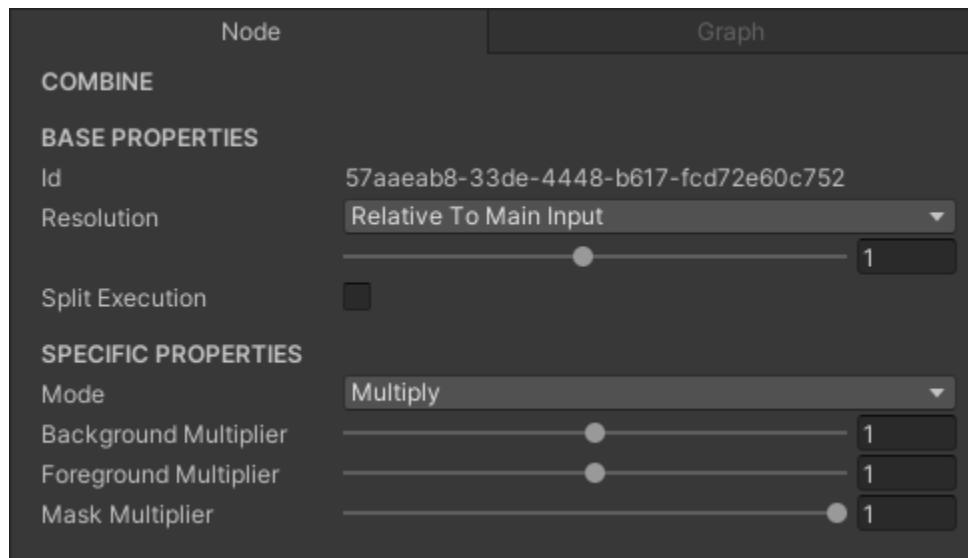
To do that, select the elements you want to group, right click to open up the context menu, then select **Group Selection**.

To remove elements from a group, right click on them and select **Ungroup Selection**.

The Properties View

The Properties View displays all detailed information about the selected node, as well as the graph asset itself.

In the Node tab, you can edit the currently selected node:



In the Graph tab, you can edit the graph detail:

	Node	Graph
Resolution	512	
World Bounds	X X Width Height Length	440.6675 405.9609 1119.461 600 1052.184
Seed	0	
Allow Split Execution	<input checked="" type="checkbox"/>	

Note that **these parameters are for debugging/previewing purposes only**, the actual values that are used in each generation pipeline will be fed from the biome.

The Status Bar

Texture: 3/6 | Buffer: 0/0 | Memory: 3.0/4.0 MB

The Status Bar provides some useful information about the graph memory usage, such as current/peak texture count, buffer count and VRAM consumption.

Data visualization

When selecting a node in the Graph View, its data (the first output slot) will be visualized in the 3D and 2D View.

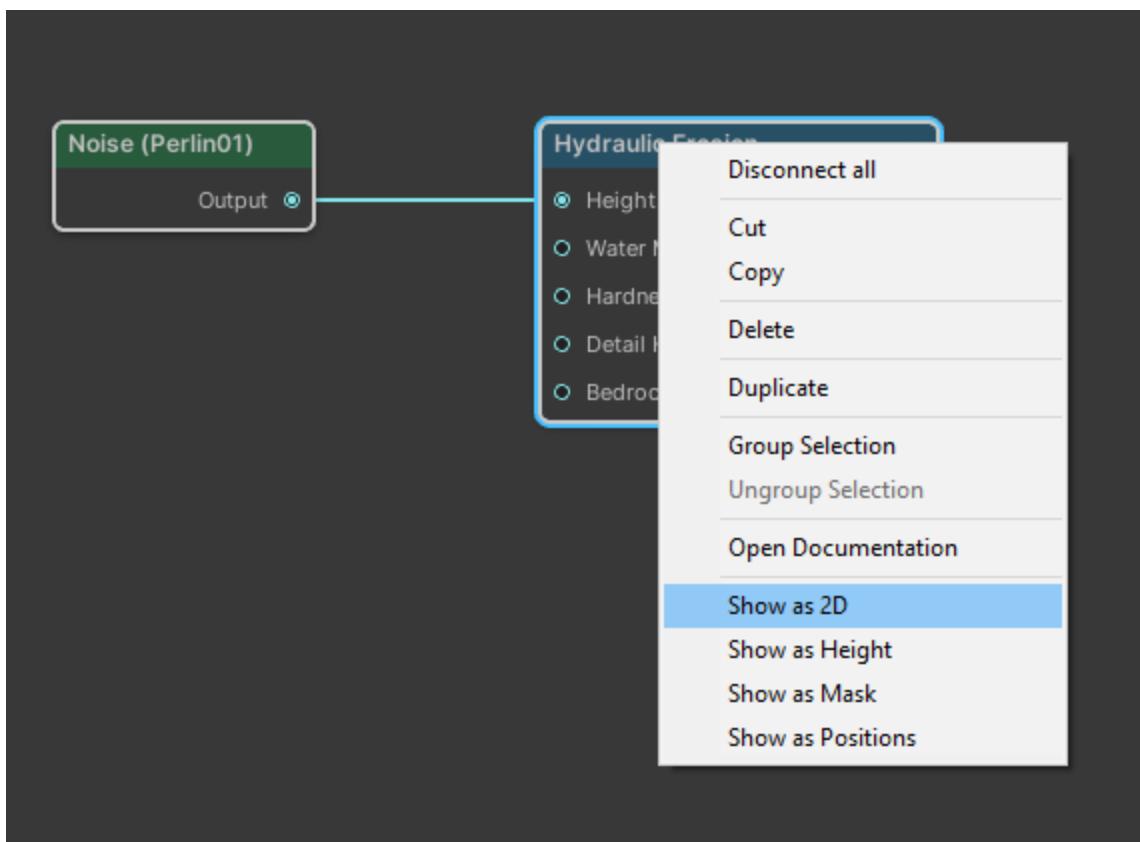
As default, texture type data (cyan & pink slot for RFloat and RGBA32 format) will be interpreted as “height value” and array type data (yellow slot) will be interpreted as “position” in the 3D View.

To interpret texture data as “mask” instead of “height value” and vice versa, use the Default Texture Display Mode in the left hand side of the 3D viewport.



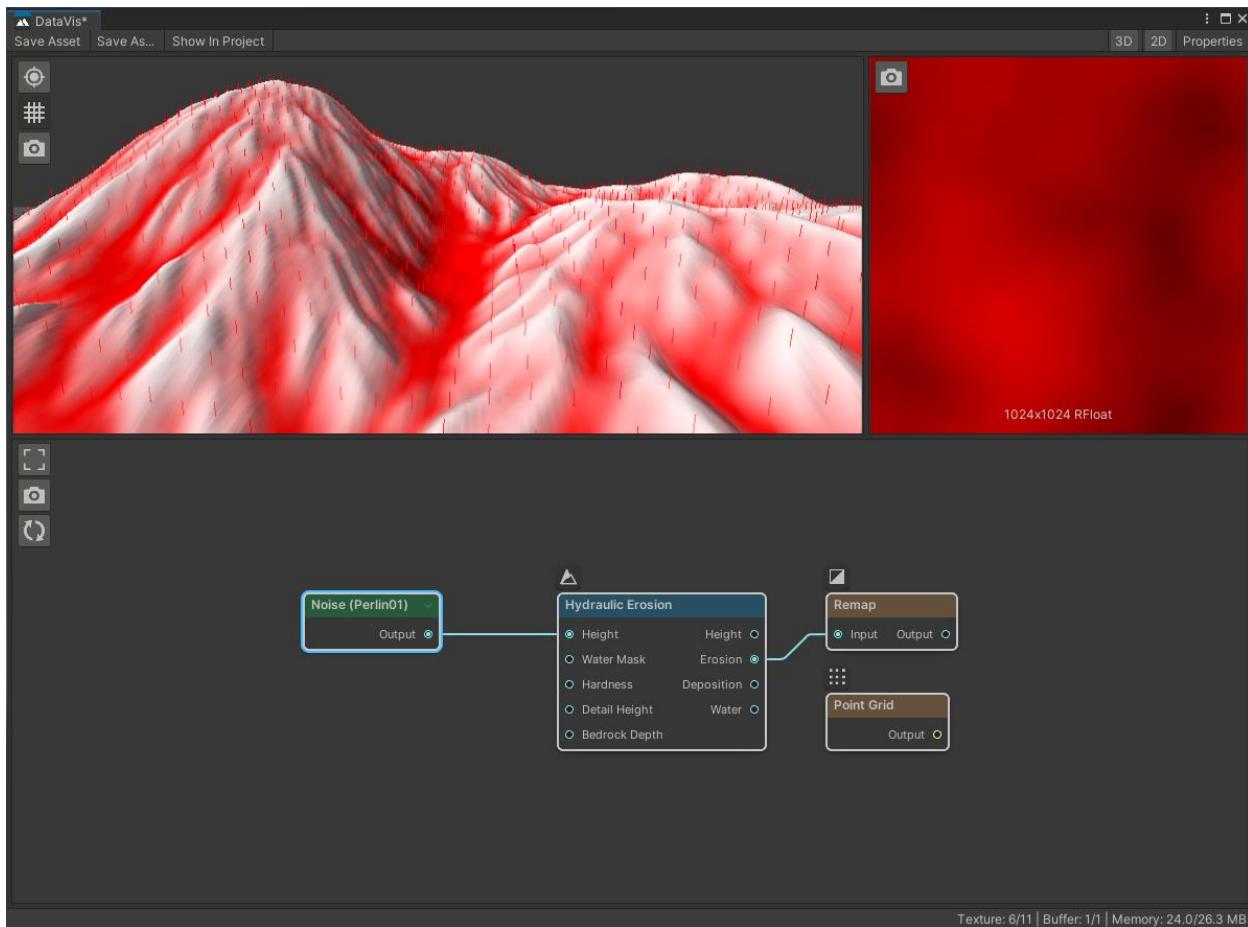
2D View can only display texture type data.

However, you can lock specific nodes to display certain kinds of data. To do so, right click on a node then select:

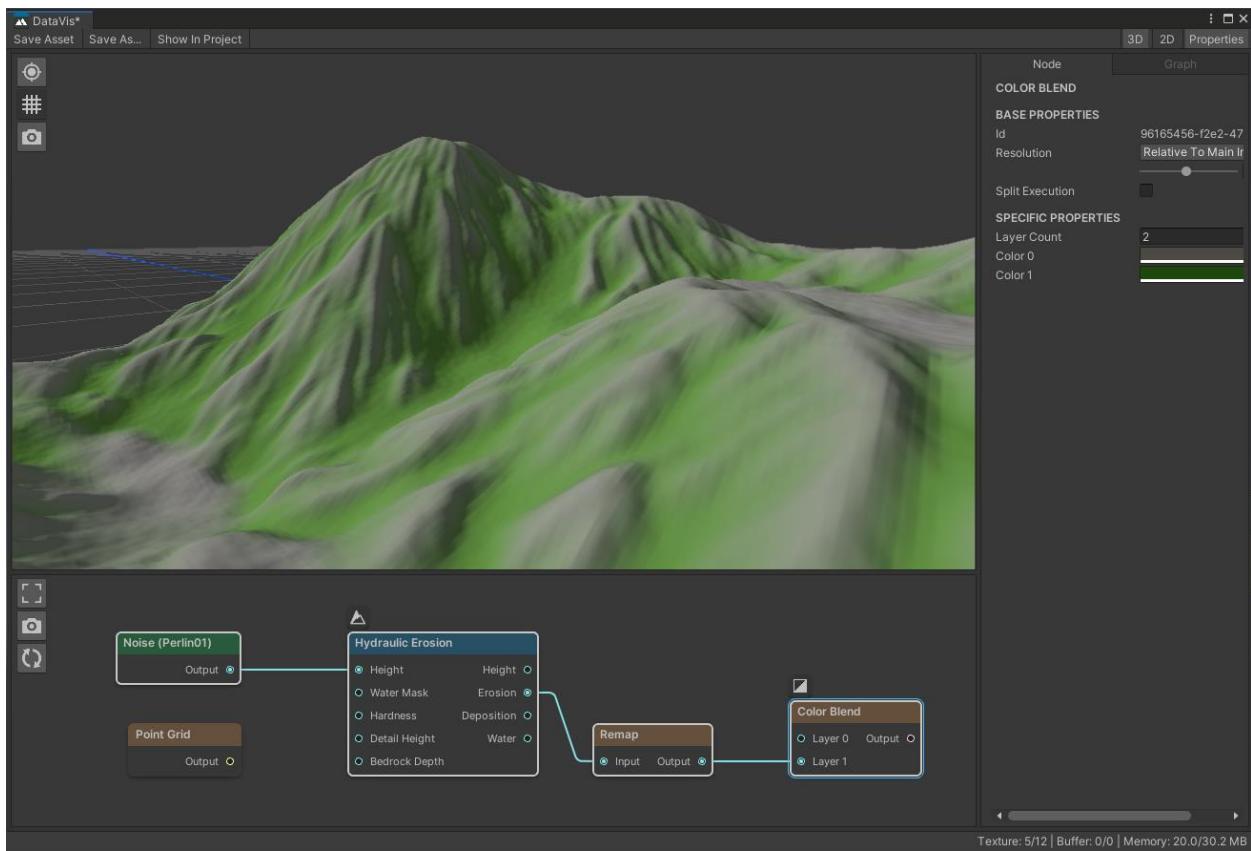


- Show as 2D: display the output texture in 2D View.
- Show as Height: display the output texture in 3D View as height map.

- Show as Mask: display the output texture in 3D View as mask map. If the output format is RFloat, it will be rendered as white-red texture. If the output format is RGBA32, it will be rendered as is.
- Show as Position: display the output array in 3D view as red markers indicate the position.

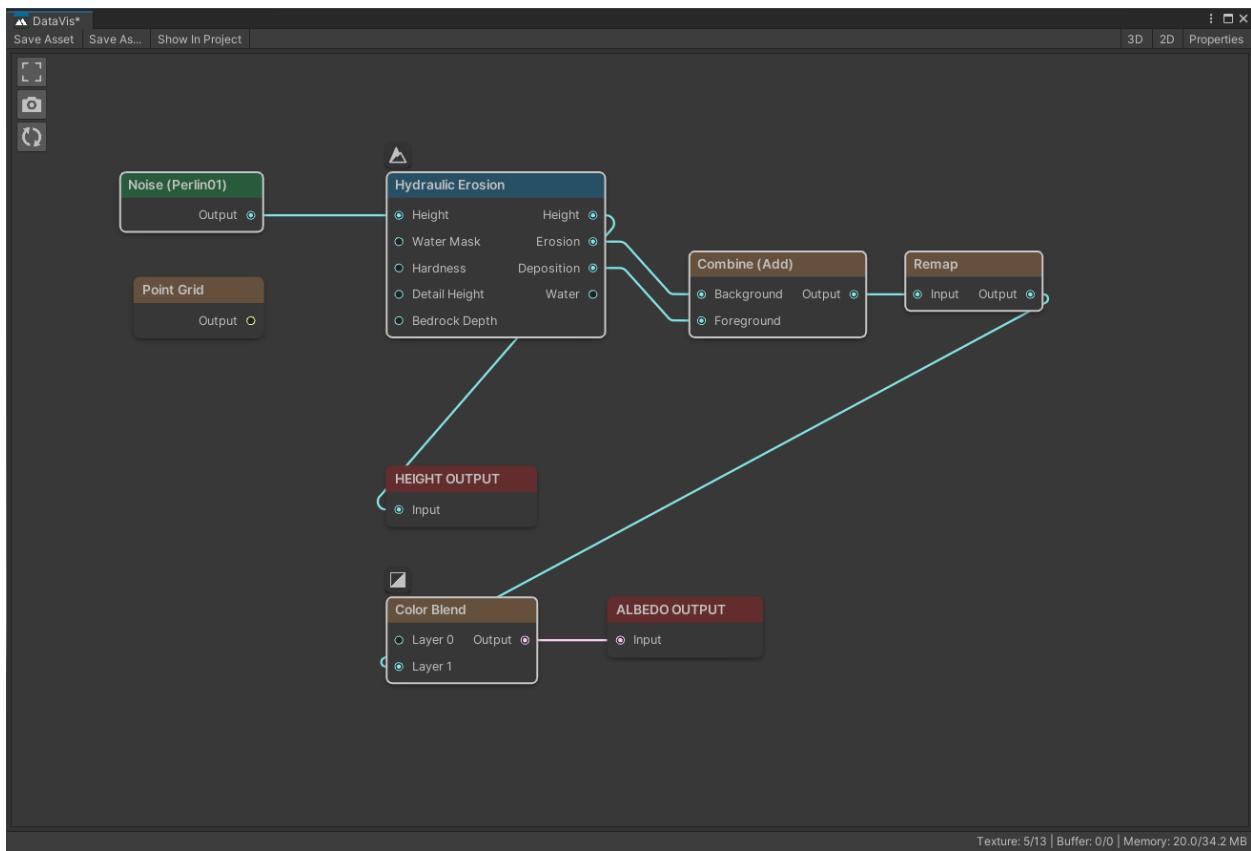


Note that there is some limitation on displaying splat textures, you need to setup a workaround using Color Blend node like this:

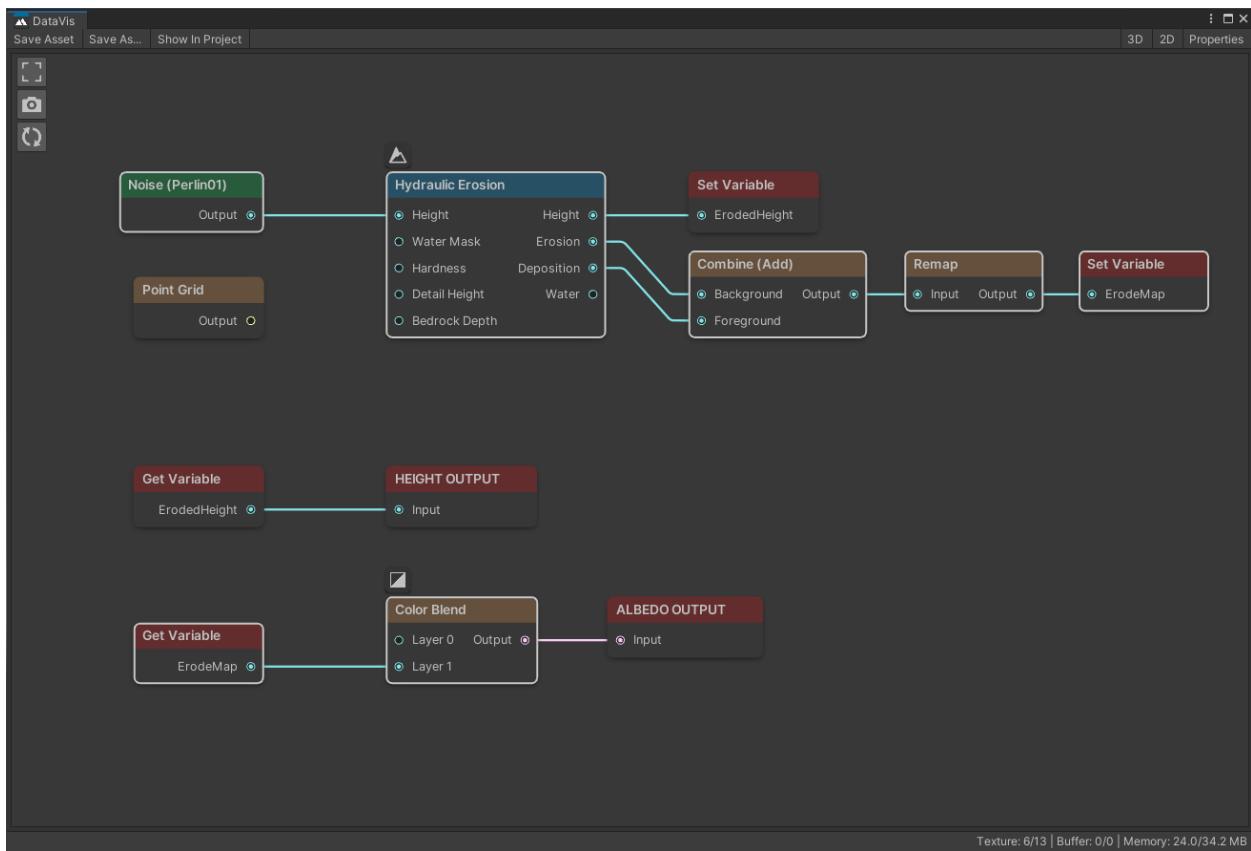


Local variables

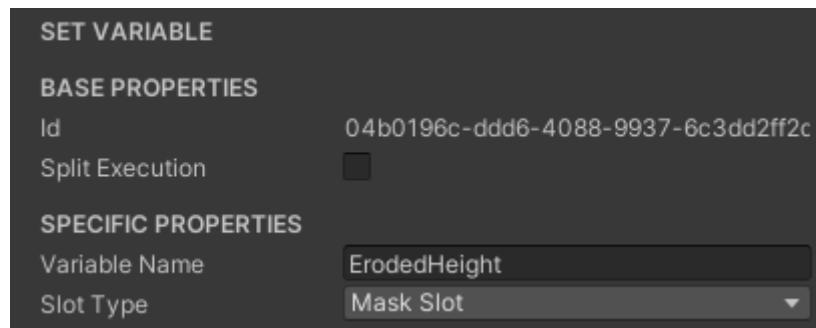
To keep the graph tidy without messy wiring, you can use the Set Variable and Get Variable node to organize your graph from this:



To this:



In the Get/Set Variable node properties, you have to assign an appropriate variable name, and select the slot data type:



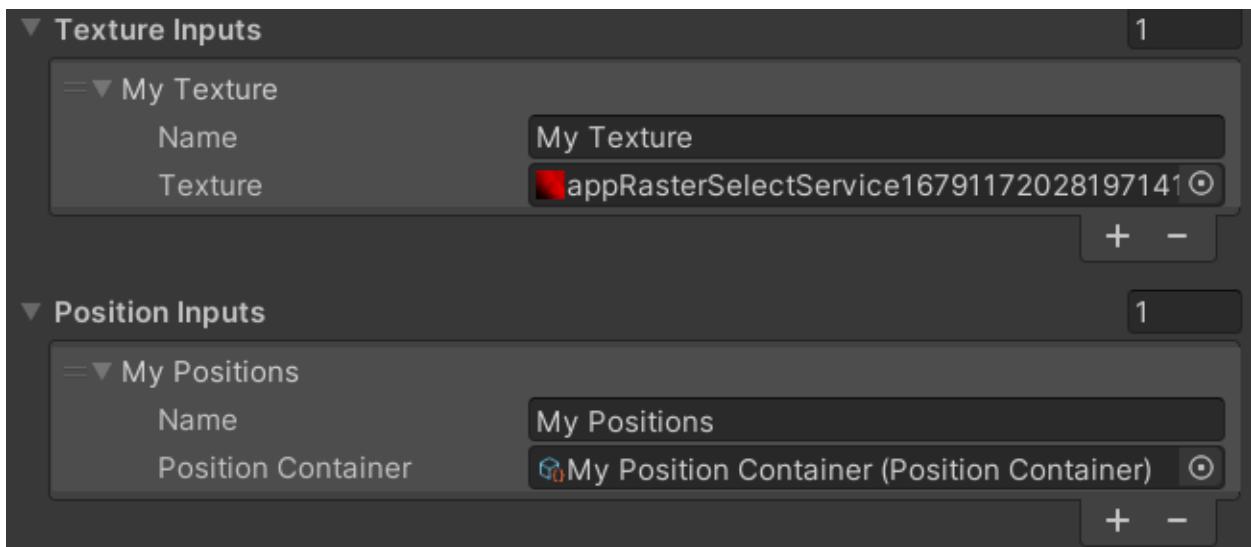
Graph input/output

You can use the Graph Input/Output node to transfer data between the graph and the outside.

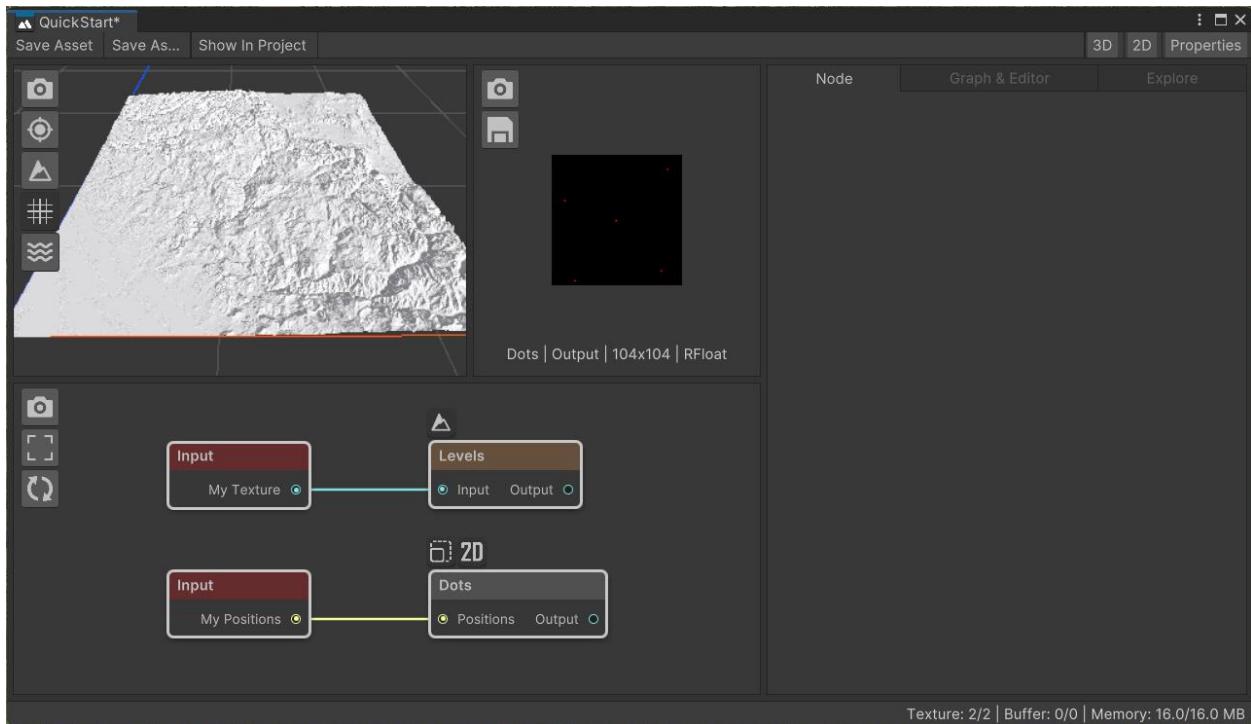
They are mainly for taking advantage of graph data and populating in custom modules.

Also, Graph Input/Output nodes can be used in the Subgraph workflow, which will be discussed in the Productivity Boost module documentation.

You can feed input to the graph by using LPB inspector:



In the graph, accessing those input by using Input Node:

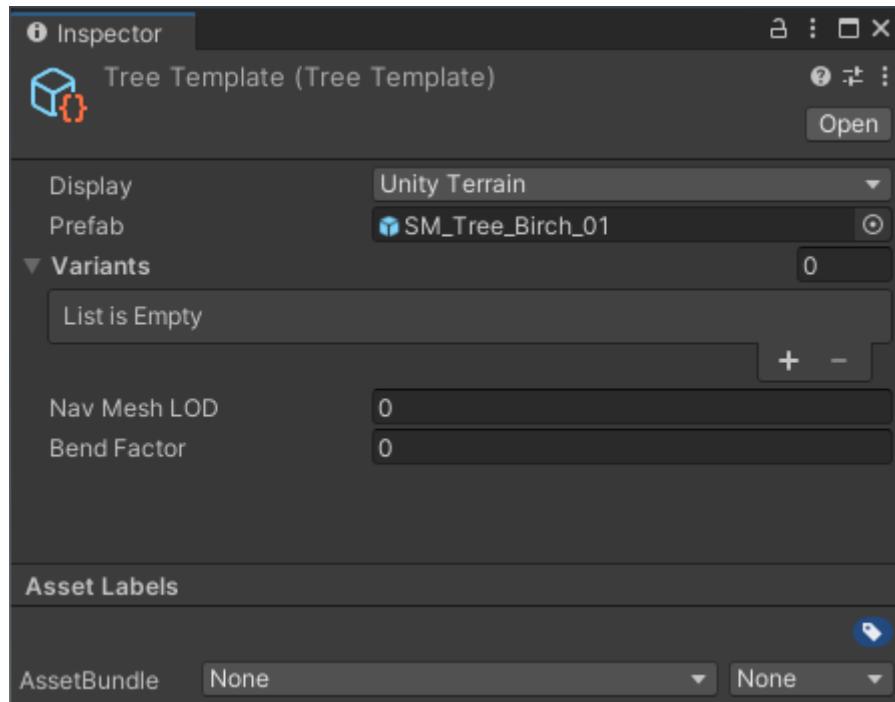


TREE TEMPLATE

Tree Template is an asset type containing prefab and settings for terrain trees.

To create one, right click on the Project window, then select **Create>Vista>Tree Template**.

In the Inspector, you can see the following properties:



- Display: select the display mode, which will only show the settings that matter (Unity Terrain, Polaris Terrain, Custom)
- Prefab: the tree prefab.
- Variants: other variants of the prefab that will be populated using the same data output from the graph. **Only available when the Productivity Boost module is active.**
- Other properties: Please see [this page](#) & [this page](#) for detail.

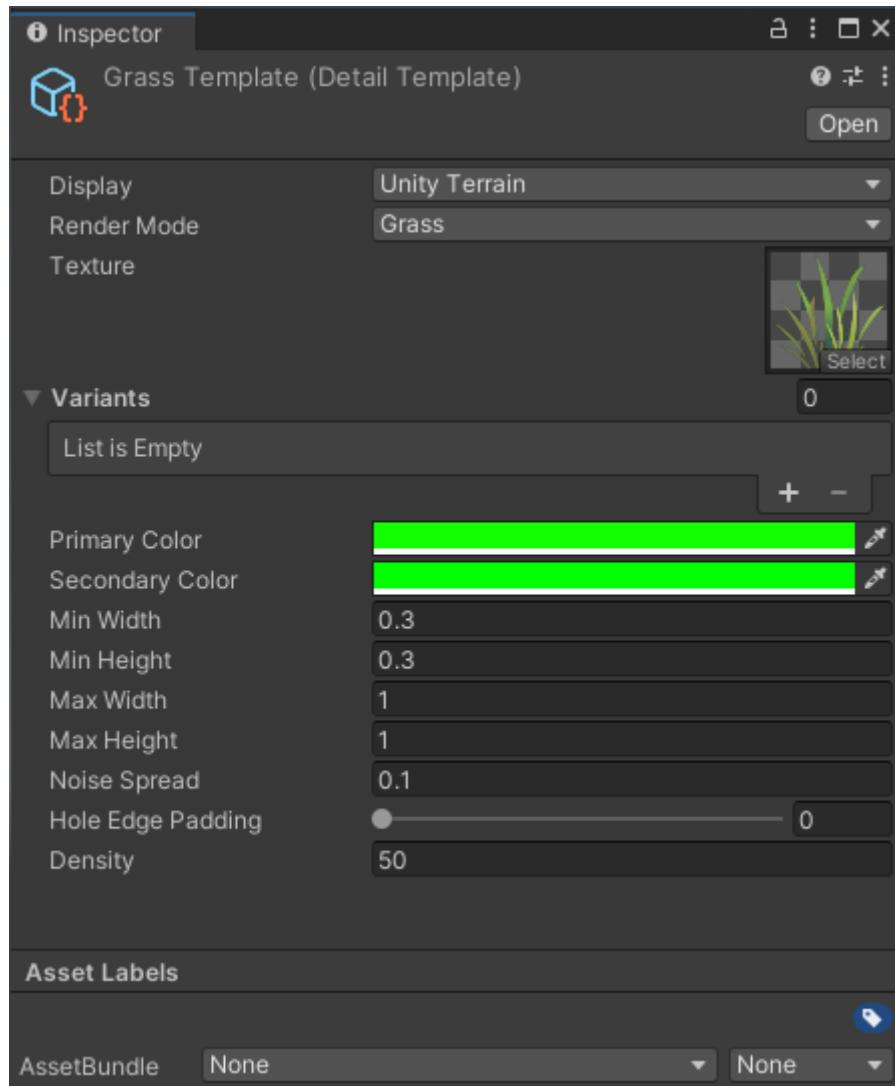
Note: Changing the settings won't take effect instantly, you need to invoke a generation process.

DETAIL TEMPLATE

DetailTemplate is an asset type containing settings for terrain grass & detail.

To create one, right click on the Project window, then select **Create>Vista>Detail Template**.

In the Inspector, you can see the following properties:



- Display: select the display mode, which will only show the settings that matter (Unity Terrain, Polaris Terrain, Custom)
- Render Mode: select if it should be rendered as grass texture or detail object.
- Texture/Prefab: the grass texture/prefab.

- Variants: other variants of the texture/prefab, that will be populated using the same data output from the graph. **Only available when the Productivity Boost module is active.**
- Density: multiplier for the density map.
- Other properties: Please see [this page](#) & [this page](#) for detail.

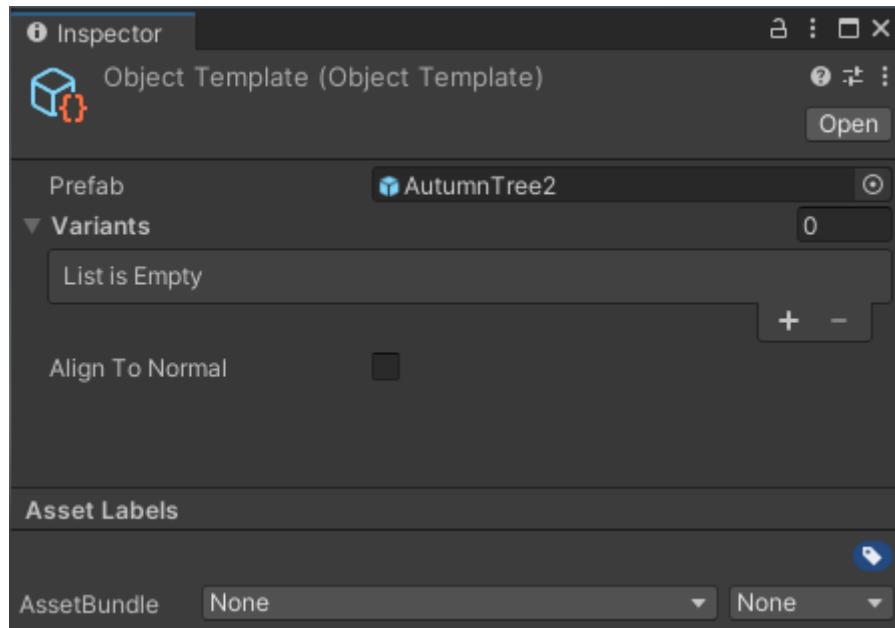
Note: Changing the settings won't take effect instantly, you need to invoke a generation process.

OBJECT TEMPLATE

Object Template is an asset type containing prefab and settings for terrain objects.

To create one, right click on the Project window, then select **Create>Vista>Object Template**.

In the Inspector, you can see the following properties:



- Prefab: the object prefab.
- Variants: other variants of the prefab that will be populated using the same data output from the graph. **Only available when the Productivity Boost module is active.**
- Align To Normal: should the object align to the surface normal vector on spawning.

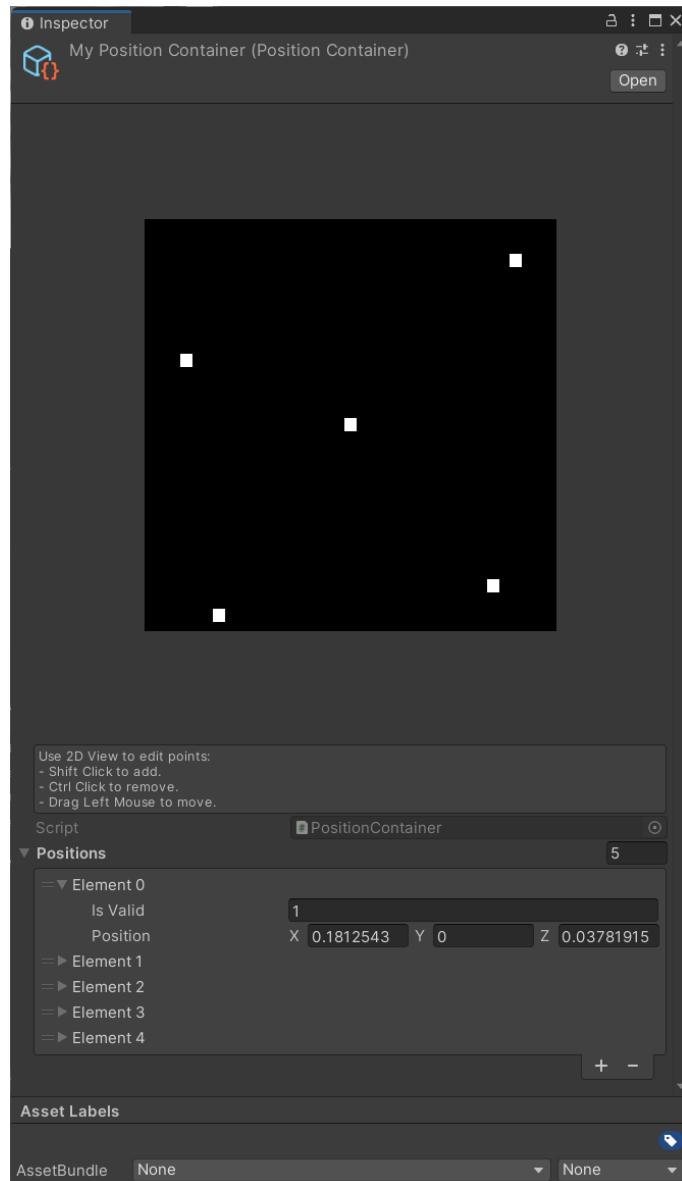
Note: Changing the settings won't take effect instantly, you need to invoke a generation process.

POSITION CONTAINER

Position Container is an asset type to provide position data as input to the graph.

To create one, right click on the Project window, then **Create>Vista>Position Container**.

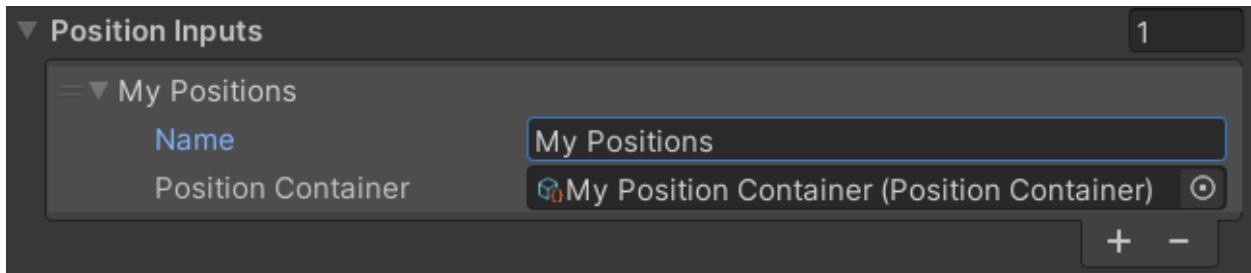
In the Inspector, you will see a black canvas like this:



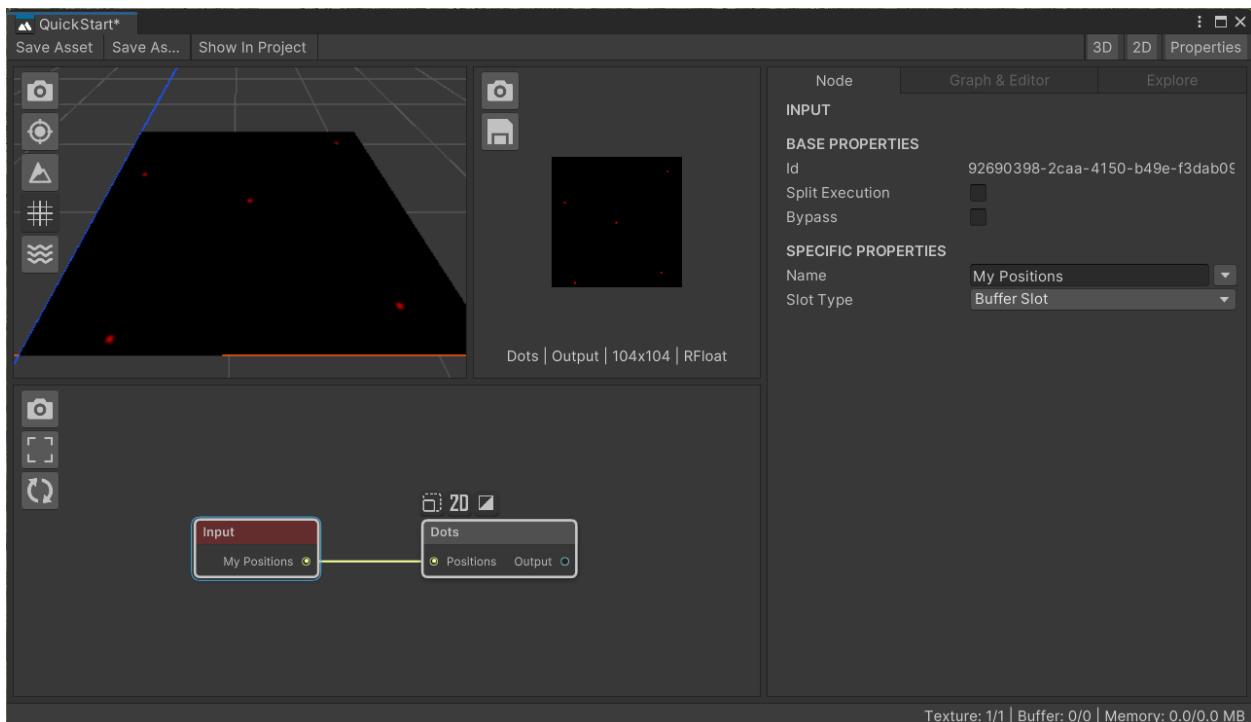
Shift Click to add a new point, Ctrl Click to remove a point, use left mouse to drag a point.

Points are in a normalized unit, that will be later map to the biome space.

In the LPB inspector, click on the + icon under the Position Inputs list, and drop the asset to its slot, give it a variable name:



In the graph, access to the position input like this:



RUNTIME GENERATION

Vista can generate terrains at runtime. Please build the Demo scene located at **Assets/PinwheelStudio/Vista/Samples/Scenes/Demo.unity** to have a try.

However, most of its generators are built using shaders, some use keywords to define their features. These keywords are usually stripped off the build by Unity that make them work incorrectly. For example: Terrace mask doesn't take into account, etc.

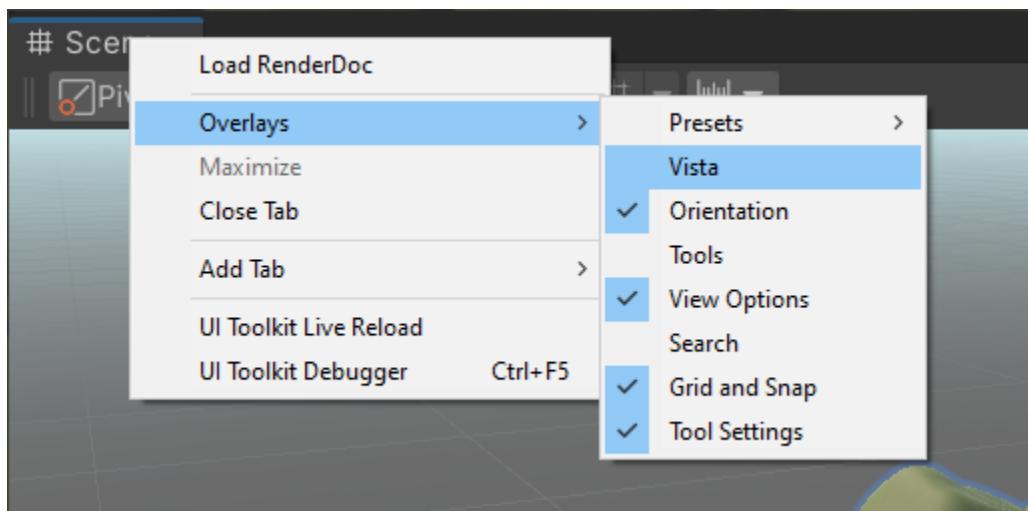
To fix it, you have to add all shaders located under the **Assets/PinwheelStudio/Vista/Runtime/Resources/Vista/Shaders** folder (.shader) to the **Always Included Shaders** list in your Graphics Settings (Edit/Project Settings/Graphics)

UTILITIES

Scene View Overlay

From Unity 2021.2 and up, you can use the Scene View Overlay to perform quick action with Vista.

To display the Overlay, right click on the Scene View tab header, select Overlays>Vista



Then, somewhere in the Scene View (usually the lower left corner), the Overlay will appear:



You can use those buttons to re-generate the terrains, show/hide gizmos, etc. depending on which modules you have. Future modules can add additional buttons to this menu.

NODES REFERENCE

This section was generated using an internal tool, the heading id changes every time it is updated, so we can't link to the exact node section from the graph editor. Please use the Search tool (Ctrl/Cmd + F) to look for the node you want.

If you have specific questions, please reach out via email or Discord for a discussion.

Nodes can be included in different modules other than the core, there will be a line indicating which module they are in.

Albedo Output Node

Included in: Vista - Core

Output to the terrain albedo texture (or base color). Mainly for Polaris & custom terrain since Unity terrain doesn't use albedo map.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

Angular Gradient Node

Included in: Vista - Core

Generate a simple angular gradient.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

- Start Point: The gradient start point
- End Point: The gradient end point
- World Position: Should it use world position or normalized position

Angular Warp Node

Included in: Vista - Core

Adding distortion to the image. The direction mask will be map to [Min Angle, Max Angle] to define the warping direction.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Min Angle: Minimum angle of the warp effect
- Max Angle: Maximum angle of the warp effect
- Direction Multiplier: Adjust the direction map
- Intensity Multiplier: Adjust the intensity map

Append Node

Included in: Vista - Core

Append 2 buffers.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

Axial Gradient Node

Included in: Vista - Core

Generate a simple axial gradient

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Start Point: The gradient start point
- End Point: The gradient end point
- World Position: Should it use world position or normalized position
- Repeat: Should the gradient repeated or clamped

Biome Scale Node

Included in: Vista - Core

Retrieve the biome scale value as mask map. This value can be multiplied with your height map to dynamically adjust the biome max height using its transform.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

Blur Node

Included in: Vista - Core

Perform Gaussian blur filter on the image. It's expensive for high radius since it executes in one frame, consider using Smooth node if you need progressive operation.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Radius: Radius of the blur effect

Clamp Node

Included in: Vista - Core

Ensure the value of the image will be in [min, max] range.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Min: Min value, mask value cannot go below this
- Max: Max value, mask value cannot go above this

Color Blend Node

Included in: Vista - Core

Perform an alpha blend of several color layers. Default value for the first slot is white, other slots is black.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution

- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Layer Count: Number of layer to paint. Layer with higher index will be painted on top of the lower ones

Combine Node

Included in: Vista - Core

Combine 2 images with a specific operation. Tips: Search for the blend mode directly (eg: add, sub, mul, max, min, linear, dif, etc.).

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Mode: The combine mode
- Background Multiplier: Tune the background before combining
- Foreground Multiplier: Tune the foreground before combining
- Mask Multiplier: Tune the mask before combining

Concave Node

Included in: Vista - Core

Highlight the pixels that are lower than their neighbors.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

- Epsilon: Slightly expand/shrink the result
- Tolerance: Relax the comparison to expand the concave area

Constant Height Node

Included in: Vista - Core

Remap the world space height to [0,1].

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Target Height: Target height value in world space

Contrast Brightness Node

Included in: Vista - Core

Apply a contrast & brightness filter on the image. Similar to other image processing software.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Contrast: Adjusting image contrast, low contrast will flatten out the terrain, where high contrast makes terrain features more distinctive
- Brightness: Adjusting image intensity, or the overall high of the terrain

Convex Node

Included in: Vista - Core

Highlight the pixels that are higher than their neighbors.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Epsilon: Slightly expand/shrink the result
- Tolerance: Minimum number of neighbor pixels that should pass the convex test

Crack Node

Included in: Vista - Big World

Simulate earth cracks at specific positions.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Smoothness: Lower value creates jaggy cracks, higher value creates smoother one
- Width: Width of the cracks
- Length: Length of the cracks
- Depth: Depth of the cracks
- Angle Limit: The surface angle limit
- Iteration Count: The number of iteration to run the simulation

Curve Node

Included in: Vista - Core

Remap the image with a curve.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Curve: The remap curve

Default Value Node

Included in: Vista - Big World

Switch to a default texture/buffer if the provided input is null.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Slot Type: Data type of the slot

Detail Density Output Node

Included in: Vista - Core

Output to the terrain's detail density map. Polaris terrains don't use density map, use Detail Instance Output Node instead.

- Id: Id of the node in this graph

- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Detail Template: Template asset for this detail. Right click on the Project window, select Vista>Detail Template to create one
- Density Multiplier: Multiplier for detail density

Detail Instance Output Node

Included in: Vista - Core

Output grass/detail instances to the terrain. Unity terrains don't use detail instances, use Detail Density Output Node instead.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Density Multiplier: Adjust the density map
- Vertical Scale Multiplier: Adjust the height scale map
- Horizontal Scale Multiplier: Adjust the width scale map
- Min Rotation: Min tree rotation in degree
- Max Rotation: Max tree rotation in degree
- Rotation Multiplier: Adjust the rotation map
- Seed: An integer to randomize the result
- Detail Template: Template for the detail

Directional Warp Node

Included in: Vista - Core

Adding distortion to the image where the direction mask will be converted to a normal map to define the warp direction.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Intensity Multiplier: Adjust the intensity map

Direction Mask Node

Included in: Vista - Core

Highlight the geometry where it's normal vector points toward a specific angle range. Ex: Having different texture for the north & south region of a mountain.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Angle: Target angle around Y-axis, in degree
- Tolerance: Expand the target angle
- Falloff: Remap the mask

Dots Node

Included in: Vista - Core

Turn position data to pixels.

- Id: Id of the node in this graph

- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

Expand Node

Included in: Vista - Core

Gradually expand/grow the mask. Useful when you want to include more regions in the mask.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Ratio: Expansion ratio, how fast a pixel bleed its content to neighbor pixels
- Iteration: How many iterations to simulate
- Iteration Per Frame: How many iterations to perform in a frame

Falloff Detail Node

Included in: Vista - Core

Adding minor detail to a base mask.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Detail Multiplier: The multiplier for the detail mask

Flatten At Node

Included in: Vista - Big World

Flatten the geometry at specific locations. Useful when you want some flat areas to spawn game objects onto.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Rotation:
- Min Rotation: Min tree rotation in degree
- Max Rotation: Max tree rotation in degree
- Rotation Multiplier: Adjust the rotation map
- Size:
- Size: Size of the splat images
- Scale Multiplier: Adjust the scale map
- World Space: Is the size in world space or texture space

Flatten Node

Included in: Vista - Core

Flatten the geometry using a mask.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Target Height: The target value to flatten to
- World Height: Is the target height in world space

Geometry Mask Node

Included in: Vista - Big World

Highlight the area which satisfies some geometry rules.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Blend Mode: How to blend the mask
- Height:
- Enable:
- Min Height: Minimum height level
- Max Height: Maximum height level
- Transition: A curve to remap the mask
- Slope:
- Enable:
- Min Angle: Minimum surface angle in degree
- Max Angle: Maximum surface angle in degree
- Transition: A curve to remap the mask
- Direction:
- Enable:
- Direction: Target angle around Y-axis, in degree
- Tolerance: Expand the target angle
- Falloff: Remap the mask

Get Variable Node

Included in: Vista - Core

Get a local variable in this graph. Use in parallel with Set Variable Node to stay organized.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Variable Name: Name of the variable
- Slot Type: Data type of the slot

Gradient Map Node

Included in: Vista - Big World

Create a color texture from a mask by mapping it to a gradient. Useful when you want to do something such as 'color by height' or 'color by normal' texturing.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Gradient: The gradient to map to
- Wrap Mode: How to sample the gradient texture when loop > 1
- Loop: The number of time to repeat the gradient pattern

Height Mask Node

Included in: Vista - Core

Highlight the area which is in a specific height range.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution

- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Min Height: Minimum height level
- Max Height: Maximum height level
- Transition: A curve to remap the mask

Height Output Node

Included in: Vista - Core

Output the height map to the terrain.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

Hole Output Node

Included in: Vista - Core

Output the hole map to the terrain. Black is surface, red is hole.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Threshold: Value under this will become 0, otherwise 1

Hydraulic Erosion Node

Included in: Vista - Core

Adding realism to the terrain by applying erosion caused by water flow. Best practice: Don't use just one node for this effect, instead chain many of them with Detail Level from low to high to have erosion effect at different scales. Low Detail Level may introduce some pixel artifacts but it can be hidden with later nodes.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- General:
 - High Quality: If true, it will simulate water flow in 8 directions, otherwise 4
 - Detail Level: Smaller value runs faster and produces larger features, while larger value is more expensive but produces more micro details
 - Iteration: The number of simulation step to perform
 - Iteration Per Frame: The number of step to perform in a single frame
 - Simulation:
 - Water Source: The amount of water pour into the system in each iteration
 - Flow Rate: Water flow speed. Default value is fine, too high may cause numerical error
 - Sediment Capacity: The amount of sediment that water can carry. Default value is fine.
 - Erosion Rate: Strength of the erosion, higher value will pick up more soil and carve deeper into the terrain
 - Deposition Rate: Strength of the deposition, higher value will add more soil back to the terrain, while lower value will make the deposition wide spread
 - Slope Factor: Decide the slope steepness where soil can deposit
 - Evaporation Rate: Strength of the evaporation that remove water from the system
 - Bed Rock Depth: Depth of the bed rock layer where it's barely eroded.
 - Sharpness: Control the sharpness of eroded features.
- Artistic Controls:
 - Height Scale: A multiplier to terrain height to further enhance the erosion effect
 - Detail Height Scale: A multiplier to the detail height map to randomize the water flow and create more eroded features on the very flat areas

- Erosion Boost: A multiplier to enhance the erosion effect
- Deposition Boost: A multiplier to enhance the deposition effect
- Utilities:
- Select Template...:

Input Node

Included in: Vista - Core

A generic entry point of the graph which will be used to take data from the outside into the graph with sub-graph or in code. This will create a corresponding slot in sub-graph node.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Name: Name of the input
- Slot Type: Data type of the slot

Landslide Node

Included in: Vista - Big World

Simulate the process of landmass break from its base and slide down to lower area.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Intensity: Intensity of the effect.
- Flow Rate: Soil 'flow' speed, how fast the soil move to lower position. Default value is fine, too high may cause numerical error

- Resting Angle: The angle in degree where soil stop sliding
- High Quality: If true, it will simulate soil movement in 8 directions, otherwise 4
- Detail Level: Smaller value runs faster and produces larger features, while larger value is more expensive but produces more micro details
- Iteration: The number of simulation step to perform
- Iteration Per Frame: The number of step to perform in a single frame

Layered Noise Node

Included in: Vista - Big World

Generate multiple noise layer and stack them on top of other, each layer can have different noise type

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Base Scale: Scale of the first layer.
- Seed: An integer to randomize the result
- Mode: Noise mode
- Strength: Amplitude multiplier for this layer
- Add Layer: Add a new layer

Levels Node

Included in: Vista - Core

Apply a simple levels filter to the image.

- Id: Id of the node in this graph

- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- In Low: Make the dark areas of the image darker, can be used to adjust the low-altitude areas of the terrain
- In Mid: Adjust the value range mid-point
- In High: Make the bright areas of the image brighter, can be used to adjust the high-altitude areas of the terrain
- Out Low: Lower bound to remap the value range
- Out High: Upper bound to remap the value range

Load Texture Node

Included in: Vista - Core

Load pixel data from a texture asset. You can load both color texture (RGBA, splat map, color map) or RAW file (height map). Raw images (.raw, .r16) will be imported as textures in the asset folder so you can pick them with the selector.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Texture: The texture to load

Math Node

Included in: Vista - Core

Perform simple numerical math operations on the texture. Tips: Search for the operation directly (eg: add, sub, mul, pow, abs, minus, oneminus, etc.).

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Enabled: Is the math config enable?
- Number: The number to add
- Operator: The math operator to apply on each texture pixel

Mesh Density Output Node

Included in: Vista - Core

Output the mesh density (sub-division) map to the terrain. Mainly for Polaris and custom terrain since Unity terrain doesn't use this map.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

Metallic Smoothness Output Node

Included in: Vista - Core

Output the Metallic Smoothness Map to the terrain, embedded in R and A channel, correspondingly. Mainly for Polaris and custom terrain since Unity terrain doesn't use this map.

- Id: Id of the node in this graph

- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Metallic Multiplier: Multiplier for metallic value (R channel)
- Smoothness Multiplier: Multiplier for smoothness value (A channel)

Noise Node

Included in: Vista - Core

Generate fractal noise, useful for basic mountain shape. Tips: Search for the noise mode directly (eg: perlin, billow, ridge, etc.).

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Pattern:
- Offset: Offset the noise origin
- Scale: Scale of the noise
- Lacunarity: The change in scale of the noise at each layer
- Persistence: The change in amplitude of the noise at each layer
- Layer Count: Number of noise layer to generate and stack up
- Mode: The noise pattern to generate
- Seed: An integer to randomize the result
- Warp:
- Mode: Add warp effect to the noise pattern
- Min Angle: Min rotation of the warp vector
- Max Angle: Max rotation of the warp vector
- Intensity: Strength of the warp effect
- Remap:
- Remap: Remap the final noise value

Object Output Node

Included in: Vista - Core

Output instances (game object) of a specific prefab (and its variants) to the terrain.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Density Multiplier: Adjust the density map
- Vertical Scale Multiplier: Adjust the height scale map
- Horizontal Scale Multiplier: Adjust the width scale map
- Min Rotation: Min tree rotation in degree
- Max Rotation: Max tree rotation in degree
- Rotation Multiplier: Adjust the rotation map
- Seed: An integer to randomize the result
- Object Template: Template for the game object

Offset Node

Included in: Vista - Core

Offset position samples in the direction defined by the direction map. The direction map will be converted to a normal map to determine the offset vector.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Min Angle: The minimum angle indicates offset direction on XZ plane (in degrees)
- Max Angle: The maximum angle indicates offset direction on XZ plane (in degrees)

- Angle Multiplier: A multiplier for direction map
- Intensity Multiplier: A multiplier for intensity map

Output Node

Included in: Vista - Core

A generic exit point of the graph which can be used to take out data with sub-graph or in code.

This will create a correspond slot in the sub-graph node.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Name: Name of the output
- Slot Type: Data type of the slot

Paint Color Node

Included in: Vista - Hand Painting

Perform painting on a 2D canvas with color

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Resolution: The size of the canvas in pixels, use smallest size possible because the texture data will be embedded in the graph asset, which will affect the editor performance when loading the graph.
- Radius: Radius of the brush
- Falloff: Falloff range of the brush
- Color: Color of the brush

Paint Mask Node

Included in: Vista - Hand Painting

Perform painting on a 2D canvas with grayscale value [0-1]

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Resolution: The size of the canvas in pixels, use smallest size possible because the texture data will be embedded in the graph asset, which will affect the editor performance when loading the graph.
- Radius: Radius of the brush
- Falloff: Falloff range of the brush
- Opacity: Opacity of the brush

Path Node

Included in: Vista - Splines

Paint a path follow a spline as a mask. The mask can be later used in texturing with Weight Blend or Color Blend node

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Falloff: Define the shape of the path mask where the curve will be mirrored using the spline center curve

Point Grid Node

Included in: Vista - Core

Generate a set of point samples in a uniform grid manner. Use in conjunction with Offset and Spread node to have a randomly looking points set.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Density: Size of the point grid
- Relative To: Decide if the density is related to absolute biome size, or number of instance per 100 meters.

Polygon Mask Node

Included in: Vista - Core

Draw a polygon defined by a set of points selected manually.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Falloff: Define the size of the falloff region
- Use 2D View to edit points: - Shift Click to add. - Ctrl Click to remove. - Drag Left Mouse to move.:

Radial Gradient Node

Included in: Vista - Core

Generate a simple radial gradient.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Center: Center point of the gradient
- Radius: Radius of the gradient
- World Position: Should it use world position or normalized position
- Repeat: Repeat the pattern

Ramp Node

Included in: Vista - Splines

Conform the terrain height to a spline, useful for matching with the road mesh. Note that it doesn't generate any road mesh.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Falloff: Define the blend factor between the current terrain height and the spline height.
The mapping function is $[0,1] \rightarrow [\text{terrainHeight}, \text{splineHeight}]$

Range Select Node

Included in: Vista - Core

Select the pixels within a value range.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Min: Min value of the range
- Max: Max value of the range

Remap Node

Included in: Vista - Core

Remap the image value to an output range. Similar to normalize & auto levels filter. Useful when you want to brighten up a very dark image such as erosion/deposition data.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Range: The min/max remap range
- Min: Output range min value
- Max: Output range max value

Rgba Merge Node

Included in: Vista - Core

Merge 4 grayscale masks into a single RGBA texture.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup

- Bypass: Ignore the execution of this node
- Multipliers:
 - R Multiplier: Multiplier for R channel
 - G Multiplier: Multiplier for G channel
 - B Multiplier: Multiplier for B channel
 - A Multiplier: Multiplier for A channel
- Custom Labels:
 - Title: Custom title
 - R Label: Custom label for R input
 - G Label: Custom label for G input
 - B Label: Custom label for B input
 - A Label: Custom label for A input
- Set Labels:

Rotation Node

Included in: Vista - Core

Extract the rotation of input geometry. The value is corresponding the angle of the normal vector on XY plane.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Remap 01: Remap the radians value to 0-1 range, otherwise it will be in [0, 2PI]

Select Position Node

Included in: Vista - Core

Manually select a set of points.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Edit Positions:
- Use 2D View to edit points: - Shift Click to add. - Ctrl Click to remove. - Drag Left Mouse to move.:

Set Variable Node

Included in: Vista - Core

Register a local variable in the graph. Use in parallel with Get Variable node to stay organized.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Variable Name: Name of the variable
- Slot Type: Data type of the slot

Shape Node

Included in: Vista - Core

Generate a primitive shape. Tips: Search for the shape directly (eg: square, disc, hemisphere, cone, etc.)

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup

- Bypass: Ignore the execution of this node
- Shape: Shape type to generate
- Scale: Scale the whole shape
- Inner Size: Size of the inner part of some shape

Sharpen Node

Included in: Vista - Core

Apply a simple sharpen filter on the image.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Intensity: Strength of the sharpen effect

Shrink Node

Included in: Vista - Core

Gradually shrink the mask. Useful when you want to include less regions in the mask.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Ratio: Expansion ratio, how fast a pixel loses its content to neighbor pixels
- Iteration: How many iterations to simulate
- Iteration Per Frame: How many iterations to perform in a frame

Slide Node

Included in: Vista - Big World

Simulate the process of objects (rock, debris, etc.) slide down the slope.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Iteration: Number of iteration to simulate
- Trail Curvature: Control how smooth/curvy the trails are

Slope Mask Node

Included in: Vista - Core

Highlight the geometry that is in a specific slope angle.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Min Angle: Minimum surface angle in degree
- Max Angle: Maximum surface angle in degree
- Transition: A curve to remap the mask

Smooth Node

Included in: Vista - Core

Gradually smooth out the image.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Iteration Count: Number of smooth iteration
- Iteration Per Frame: Number of iteration to perform in a frame if Split Execution is on

Snow Fall Node

Included in: Vista - Big World

Simulate the process of snow fall and deposit, to hide the geometry below.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Snow Amount: The amount of snow falls into the system in each iteration
- Flow Rate: Snow 'flow' speed, how fast the snow move to lower position. Default value is fine, too high may cause numerical error
- Resting Angle: The angle in degree where snow stop sliding
- High Quality: If true, it will simulate snow movement in 8 directions, otherwise 4
- Detail Level: Smaller value runs faster and produces larger features, while larger value is more expensive but produces more micro details
- Iteration: The number of simulation step to perform
- Iteration Per Frame: The number of step to perform in a single frame

Splatter Node

Included in: Vista - Big World

Stamp a mask at specific positions. Useful when you want to apply different texture where it is a village, a large rock or a large tree, etc.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Rotation:
- Min Rotation: Min tree rotation in degree
- Max Rotation: Max tree rotation in degree
- Rotation Multiplier: Adjust the rotation map
- Size:
- Size: Size of the splat images
- Scale Multiplier: Adjust the scale map
- World Space: Is the size in world space or texture space
- Intensity:
- Intensity Multiplier: Adjust the intensity of the mask

Spline Extract Node

Included in: Vista - Splines

Extract a spline information to be used in other operation. There are other nodes for common spline based actions such as Ramp, Path, Thin Out Along, etc. node

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

Spread Node

Included in: Vista - Core

Spread the source positions into many others. Use this node to have a more realistic tree spawning.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Count: Number of new elements to spread from the source element
- Distance: Distance of the new elements from the source element
- Seed: An integer to randomize the result
- Keep Source Points: Should it include source points in the result?

Terrace Node

Included in: Vista - Core

Apply a terrace/quantize effect on the geometry. Best practice: Chain 2 or more of this node with different step height, use in conjunction with erosion nodes will yield impressive result.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Terrace:
- Step Height: Height of a step in world space
- Remap: Remap the step value
- Height Mask:
- Enable: Use the provided height map to mask out the terrace
- Min Height: Minimum height level to apply terrace effect

- Max Height: Maximum height level to apply terrace effect
- Blend: A curve to blend between original and terraced areas
- Slope Mask:
- Enable: Use the provided height map to calculate slope steepness and mask out the terrace
- Min Angle: Minimum slope angle to apply terrace effect
- Max Angle: Maximum slope angle to apply terrace effect
- Blend: A curve to blend between original and terraced areas
- Outline:
- Position:
- Tolerance:

Texture Output Node

Included in: Vista - Core

Output the terrain's weight map (or splat control map, alpha map) of a terrain layer. Tips: Use Weight Blend node to adjust the weight map to ensure the final result is not overshoot.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Terrain Layer: The terrain layer associate with this output
- Order: The sorting order of this layer in the graph output. Some terrain shaders provide per-texture properties that require terrain textures to be in the correct order. If many nodes have the same order, they will be sorted using their position in the node list.

Thermal Erosion Node

Included in: Vista - Big World

Simulate the erosion caused by high temperature. Best practice: Chain up 2 or more of this node with Detail Level from low to high to have better result with eroded features at different size.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- General:
- High Quality: If true, it will simulate soil transport in 8 directions, otherwise 4
- Detail Level: Smaller value runs faster and produces larger features, while larger value is more expensive but produces more micro details
- Iteration: The number of simulation step to perform
- Iteration Per Frame: The number of step to perform in a single frame
- Simulation:
- Erosion Rate: Strength of the erosion, higher value causes more soil to slide down the slope
- Resting Angle: The angle in degree where soil stop sliding
- Artistic Controls:
- Height Scale: A multiplier to terrain height to further enhance the erosion effect
- Detail Height Scale: A multiplier to the detail height map to randomize the water flow and create more eroded features on the very flat areas
- Erosion Boost: A multiplier to enhance the erosion effect
- Deposition Boost: A multiplier to enhance the deposition effect

Thin Out Along Node

Included in: Vista - Splines

Remove instances along a spline. Similar to the Thin Out node but with a spline mask.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Falloff: Define the shape of the path mask where the curve will be mirrored using the spline center curve
- Invert: Perform invert on the spline mask. If on, the instances inside the spline path will be removed. If off, the instances outside the spline path will be removed.
- Input Count: The number of input position buffer to perform the thin out action

Threshold Node

Included in: Vista - Core

Apply a threshold filter on the image. Value lower than Min will turn black, while value higher than Max will turn white.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Low: Lower threshold, value below this point will turn 0
- High: Upper threshold, value above this point will turn 1
- Compare Mode: How to compare the value against the threshold

Transform 2d Node

Included in: Vista - Core

Perform some 2D transformation on an image.

- Id: Id of the node in this graph

- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Slot Type: Data format of the input
- Tiling Mode: How to tile the image
- Offset: Offset of the image in %
- Rotation: Rotation of the image in degree
- Scale: Scale of the image in %
- Background Color: The background color when there is no tiling

Transform Points Node

Included in: Vista - Core

Perform some 2D transformation on a set of points.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Offset: Offset of the points set in %
- Rotation: Rotation of the points set in degree
- Scale: Scale the points set in %

Tree Output Node

Included in: Vista - Core

Output tree instances of a prefab (and its variants) to the terrain.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup

- Bypass: Ignore the execution of this node
- Density Multiplier: Adjust the density map
- Vertical Scale Multiplier: Adjust the height scale map
- Horizontal Scale Multiplier: Adjust the width scale map
- Min Rotation: Min tree rotation in degree
- Max Rotation: Max tree rotation in degree
- Rotation Multiplier: Adjust the rotation map
- Seed: An integer to randomize the result
- Tree Template: Template for the tree

Thin Out Node

Included in: Vista - Core

Remove some samples from the set using a mask. The blacker the mask, the higher change that the samples will be removed.

- Id: Id of the node in this graph
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Mask Multiplier: Multiplier for Mask texture
- Seed: An integer to randomize the result

Value Check Node

Included in: Vista - Core

A simple way to check the content of an image. Pixels with value lower than 0 will be highlighted with green, higher than 1 will be highlighted with cyan.

- Id: Id of the node in this graph

- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node

Voronoi Node

Included in: Vista - Core

Generate a fractal voronoi/worley pattern.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Pattern:
- Offset: Offset the pattern origin
- Scale: Scale of the pattern
- Amplitude Exponent: Decide the shape of the pattern, sharp or round, blob like
- Lacunarity: The change in scale of the noise at each layer
- Persistence: The change in amplitude of the noise at each layer
- Layer Count: Number of layer to generate and stack up
- Inverse: Inverse the pattern to create a blob look
- Seed: An integer to randomize the result
- Output Raw Cell: Output the pattern as solid cells
- Warp:
- Mode: Add warp effect to the noise pattern
- Min Angle: Min rotation of the warp vector
- Max Angle: Max rotation of the warp vector
- Intensity: Strength of the warp effect
- Remap:
- Remap: Remap the final noise value

Water Flow Node

Included in: Vista - Big World

Simulate the process of water flow over the surface. No erosion or deposition happens.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution
- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Water Source: The amount of water pour into the system in each iteration
- Rain Rate: Rain probability
- Flow Rate: Water flow speed. Default value is fine, too high may cause numerical error
- Evaporation Rate: Strength of the evaporation that remove water from the system
- High Quality: If true, it will simulate water flow in 8 directions, otherwise 4
- Detail Level: Smaller value runs faster and produces larger features, while larger value is more expensive but produces more micro details
- Iteration: The number of simulation step to perform
- Iteration Per Frame: The number of step to perform in a single frame

Weight Blend Node

Included in: Vista - Core

An easy way to blend texture weight maps. Weight maps will be stacked layer by layer; lower layers will be adjusted to ensure their sum is not overshoot. Default value for the first layer is white, other layers is black.

- Id: Id of the node in this graph
- Resolution: Decide the texture's resolution

- Split Execution: Split the node execution to several frames to reduce hiccup
- Bypass: Ignore the execution of this node
- Layer Count: Number of layer to blend. Layer with higher index will be 'painted' on top of the lower ones. Similar to how you use the Paint Texture in Terrain Tool

MODULE: BIG WORLD

Overview

This module provides more advanced features to unlock the full power of Vista that allows you to create bigger environment at ease, including:

- Multi-biomes workflow: Place an unlimited biome in the scene and let them blend nicely together.
- Up to 4K textures for each biome: More pixels, more coverage, more detail.
- Extended node library: Landslide, crack, snow fall, water flow and more than that.

Version: 2023.1.0

Requirements: Vista 2023.1.0+

Multi-biomes workflow

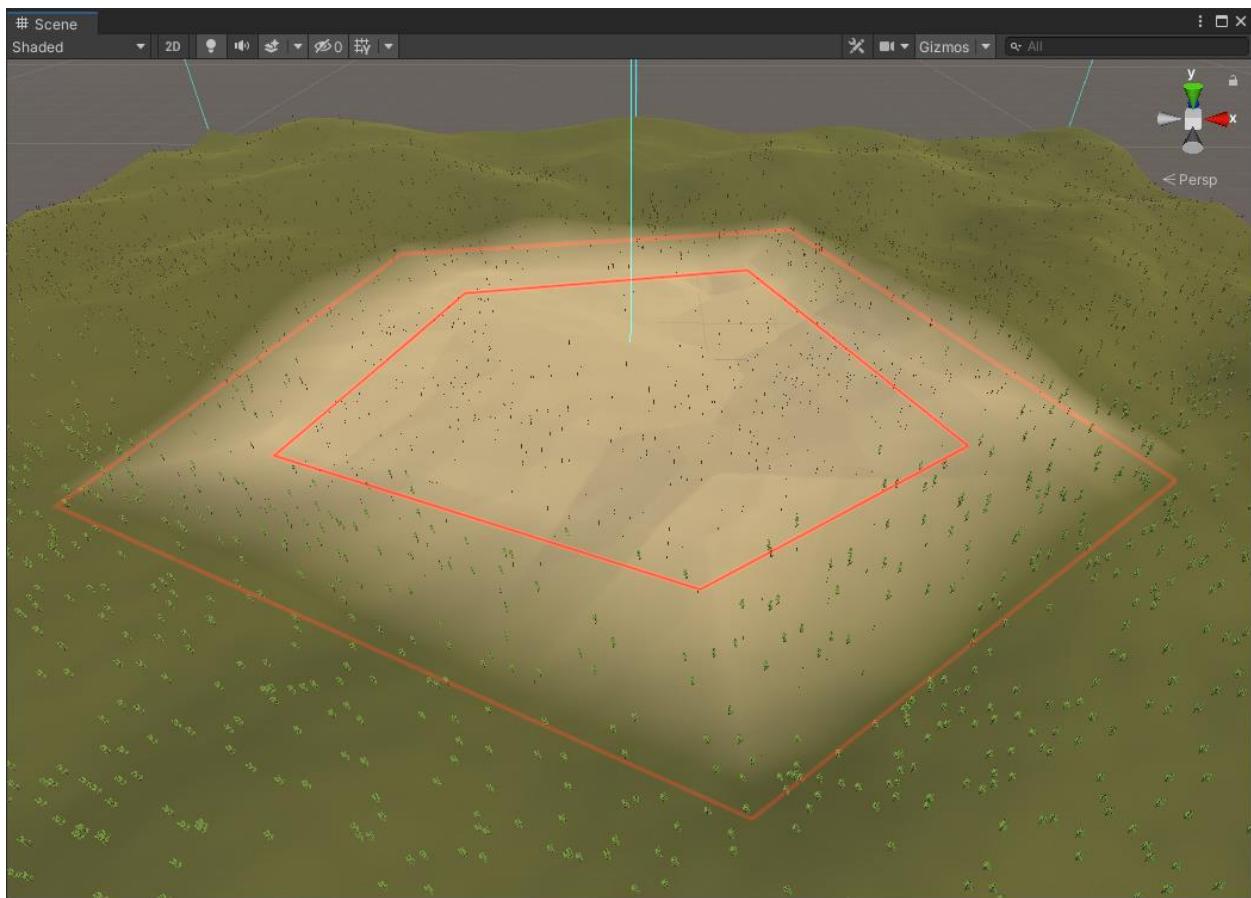
You can now create more than 1 biome anywhere in the scene, the process is similar to [this section](#). If those biomes overlap, they will be blended together with a very nice transition.

Biomes blending

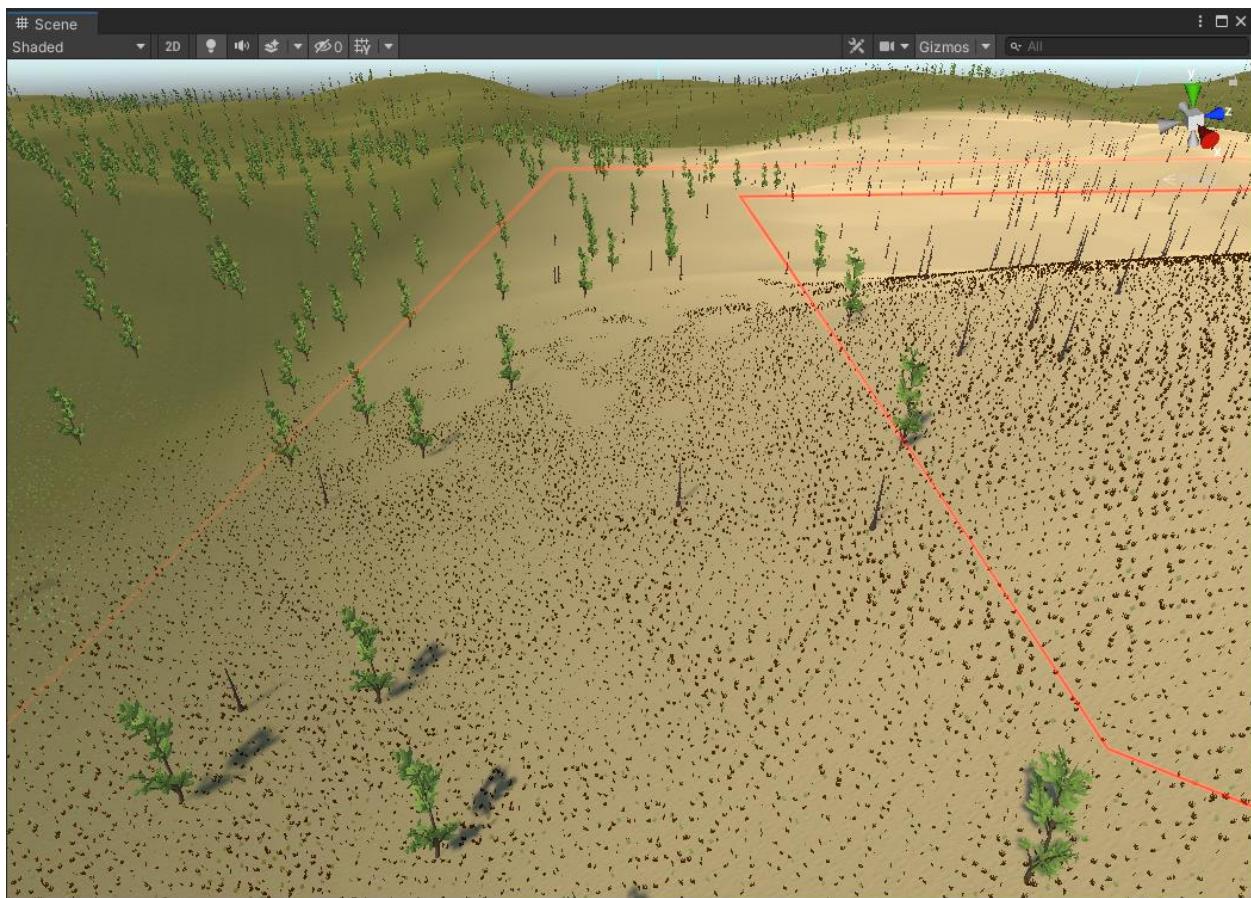
Overview

Vista can blend multiple biomes together with a nice transition between them. Biomes belonging to a VM instance are sorted by their order, later biomes will be “blended” on top of the previous ones using biome mask.

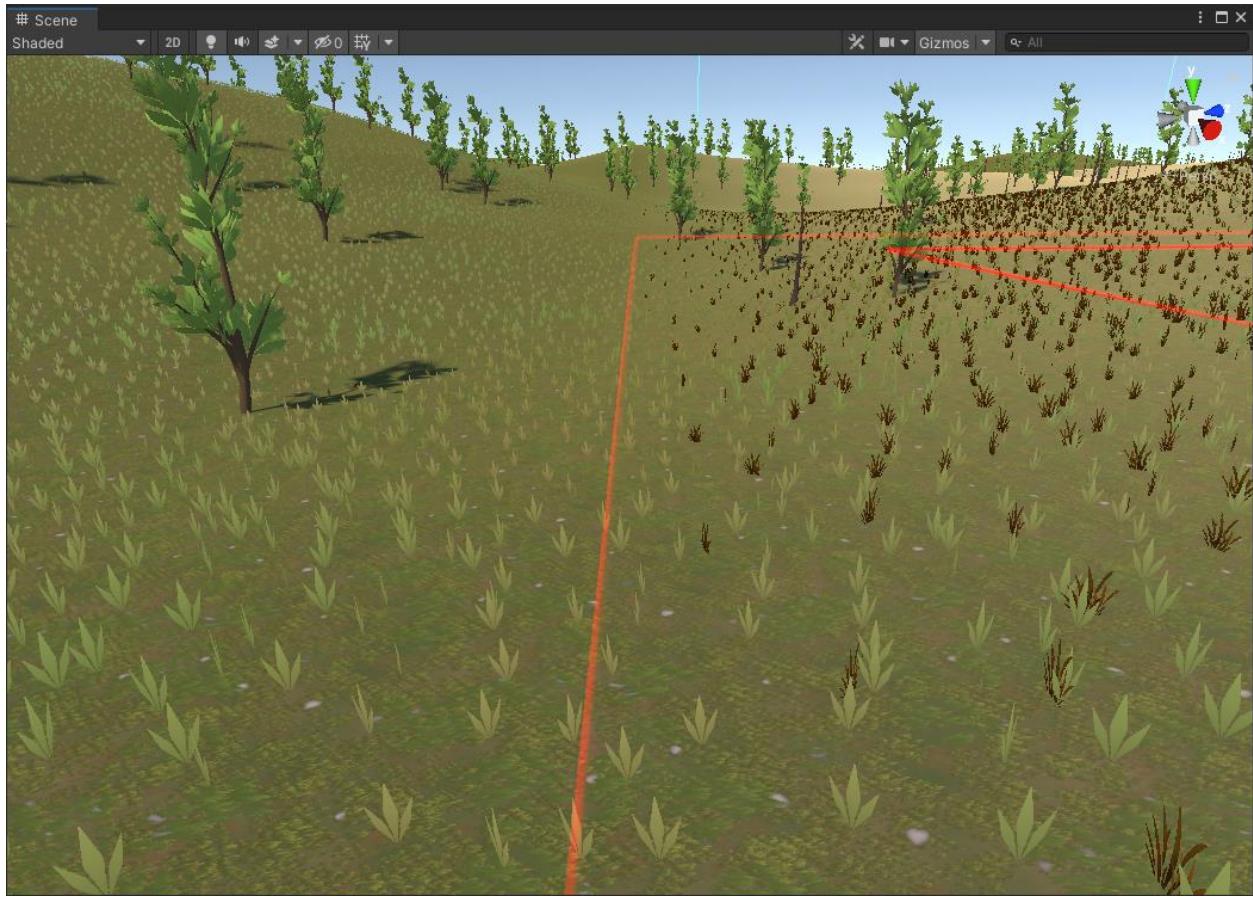
Here's an example of biome blending:



In the image above, you can see that terrain geometry and texture are gradually changing from the grassy to the desert one.



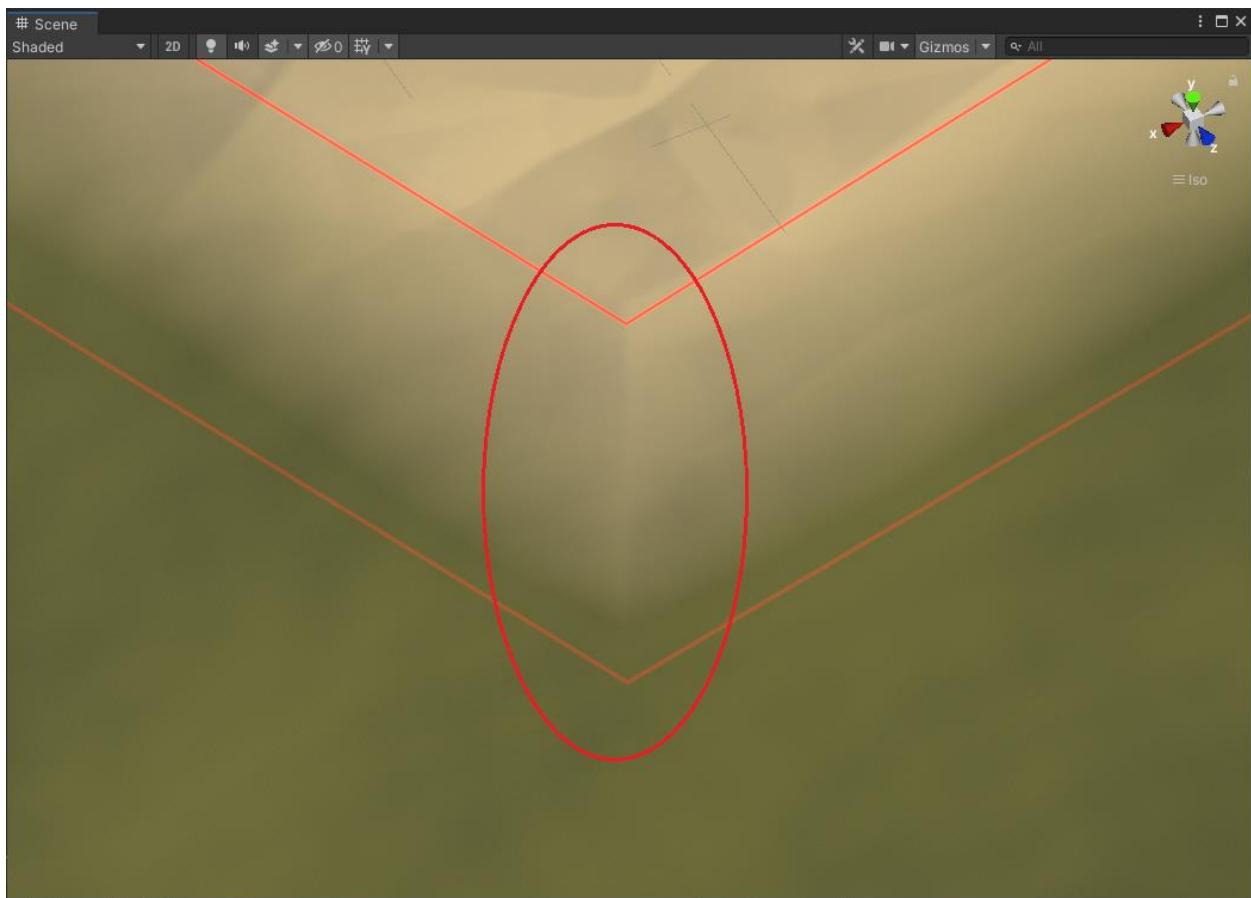
Green trees are thinned out and then completely disappear when it goes from grassy to desert biome.



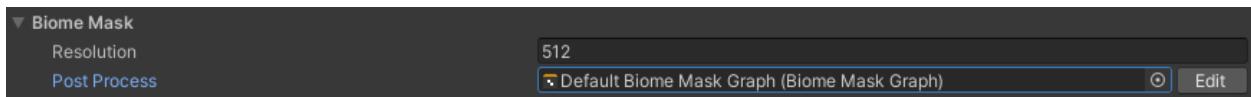
Grass and detail objects also get blended.

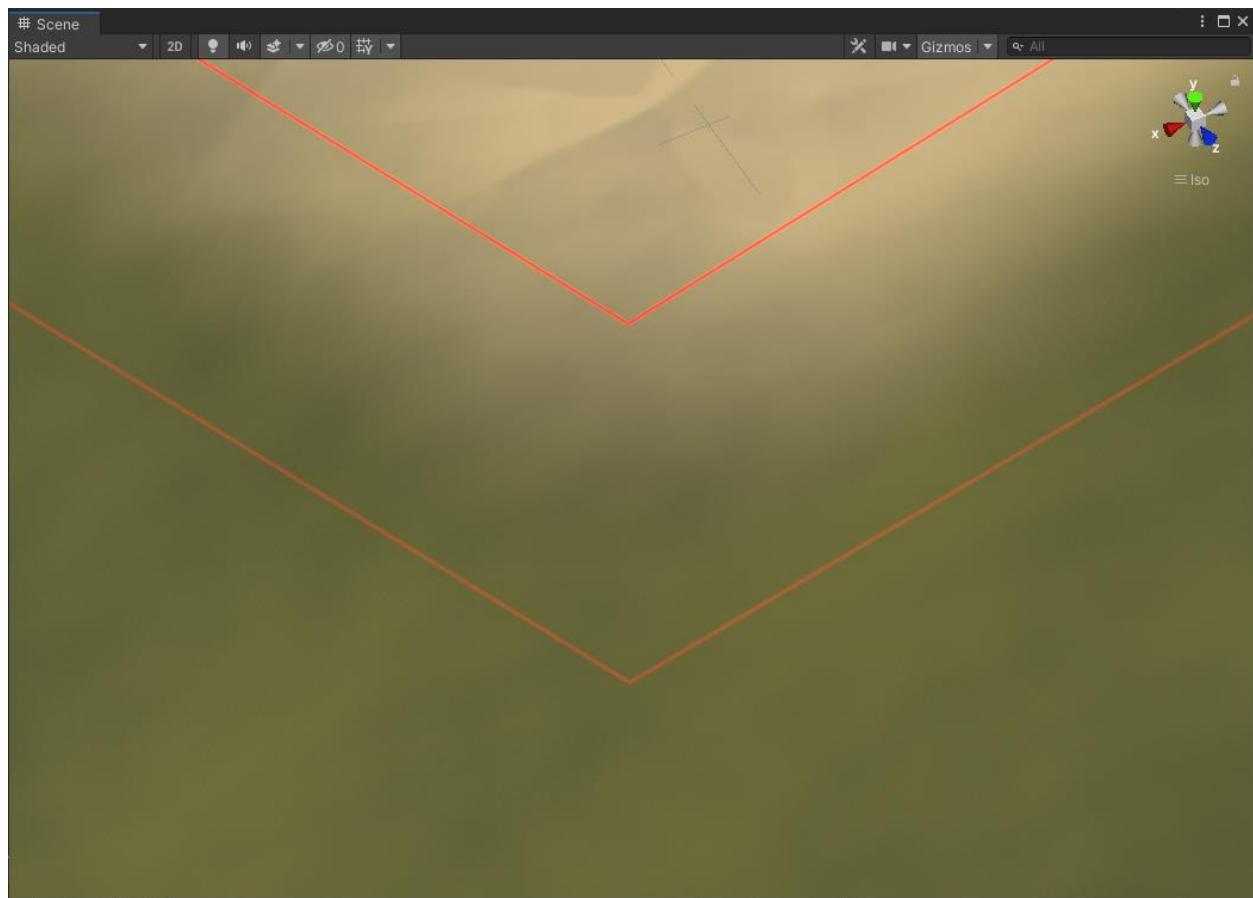
Better transition

Biomes are blended using a biome mask, an internal texture generated when you edit the biome anchors. Initially, the biome mask texture can contain some hard edged areas due to its polygonal nature:



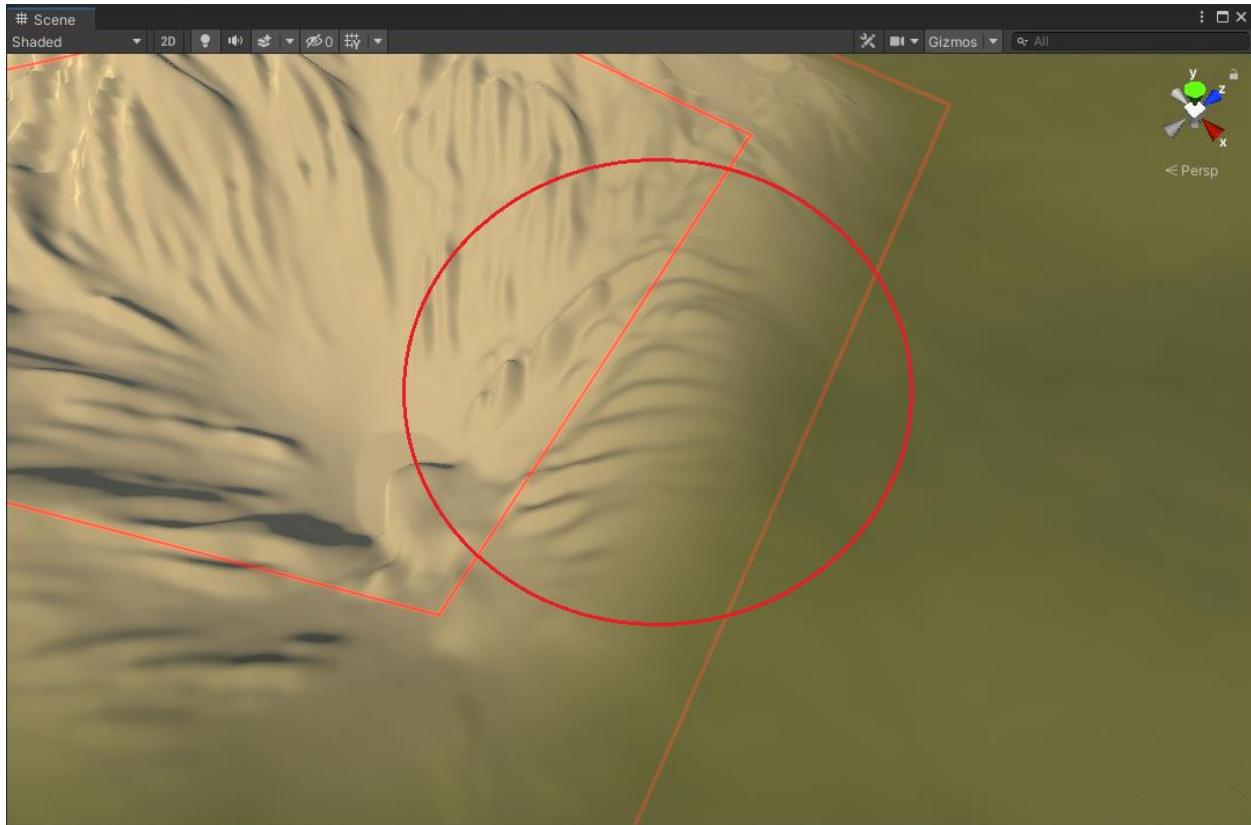
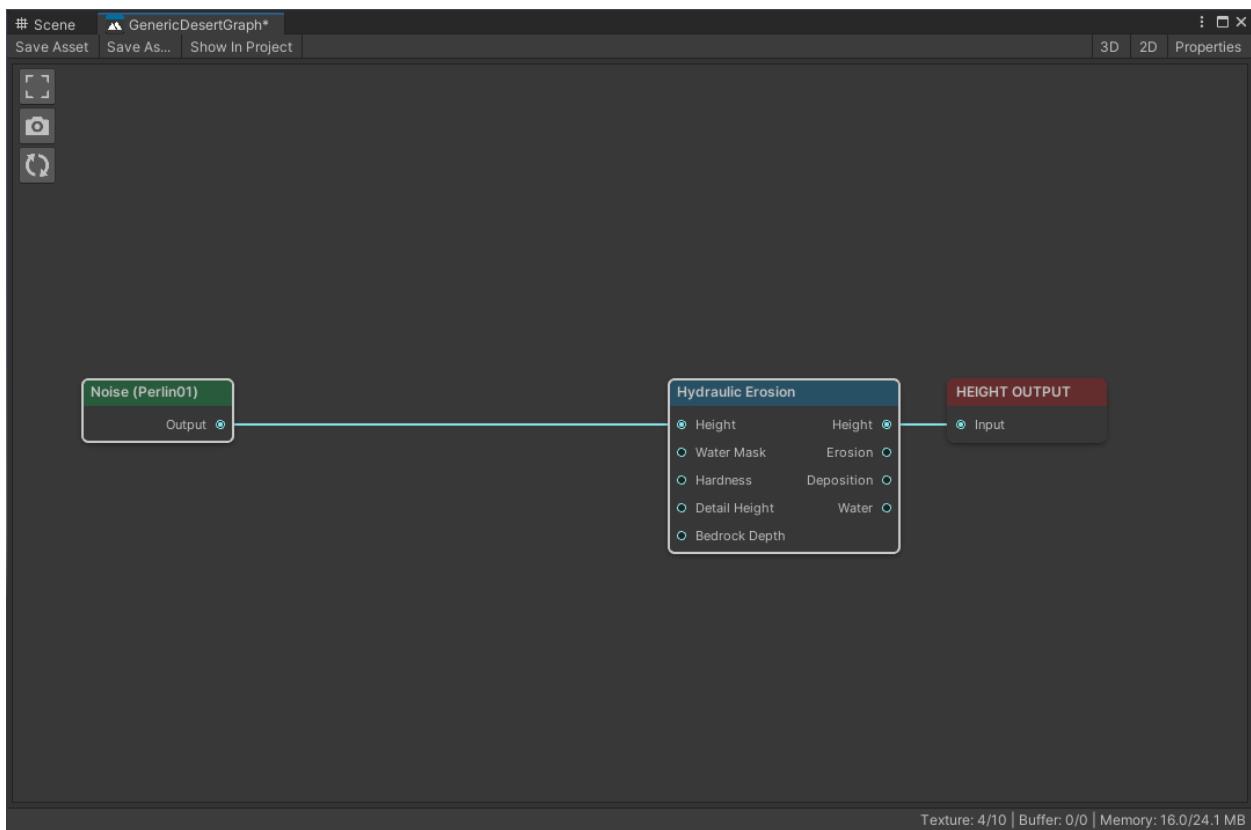
Luckily, Vista also provides the [Biome Mask Graph](#) that lets you polish the mask. Apply the graph to the biome, we have a way better transition:





Advanced transition

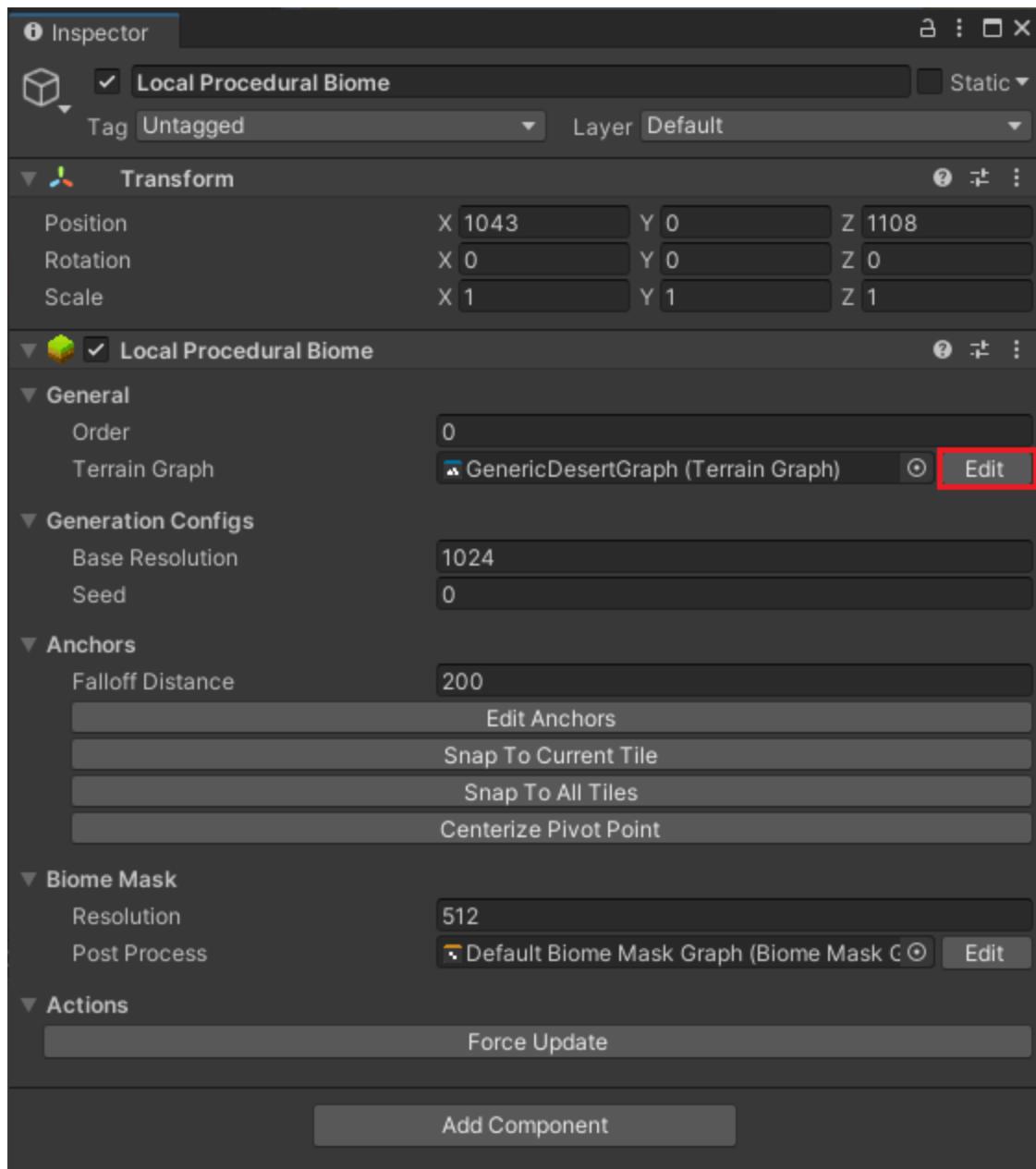
Sometime retouching the biome mask is not enough for a good transition, consider the example below:



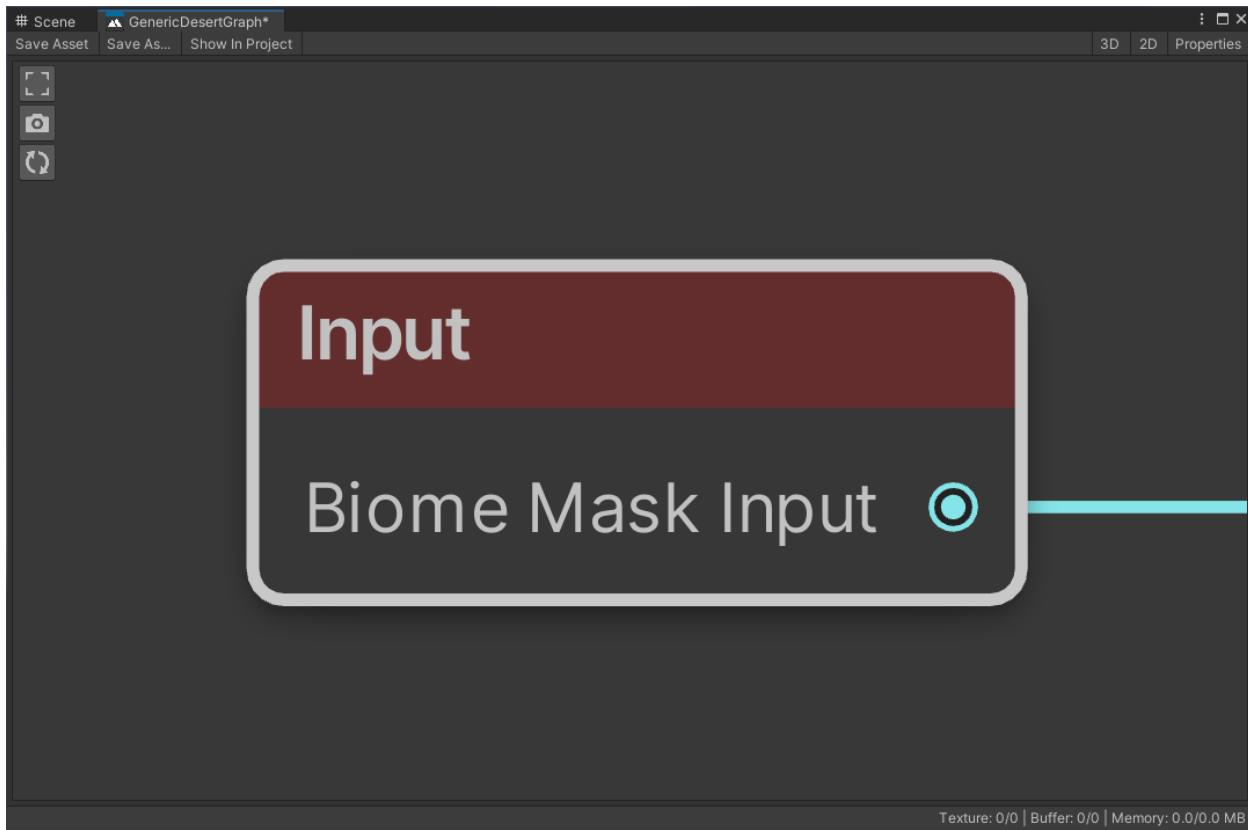
You can see that the erosion looks unnatural, like water is flowing uphill.

A way we can get around this is to take advantage of the biome mask directly in the terrain graph beforehand.

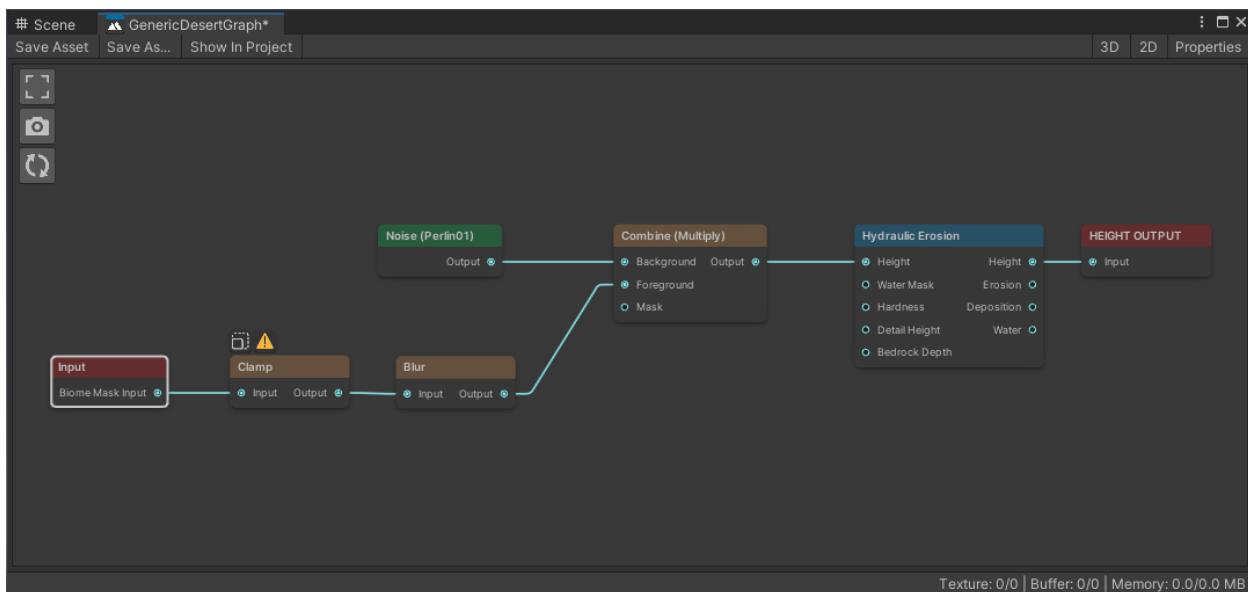
To do this, select the biome instance and click on the Edit button next to the terrain graph slot. It will bring up the graph editor with the biome mask fed as input.



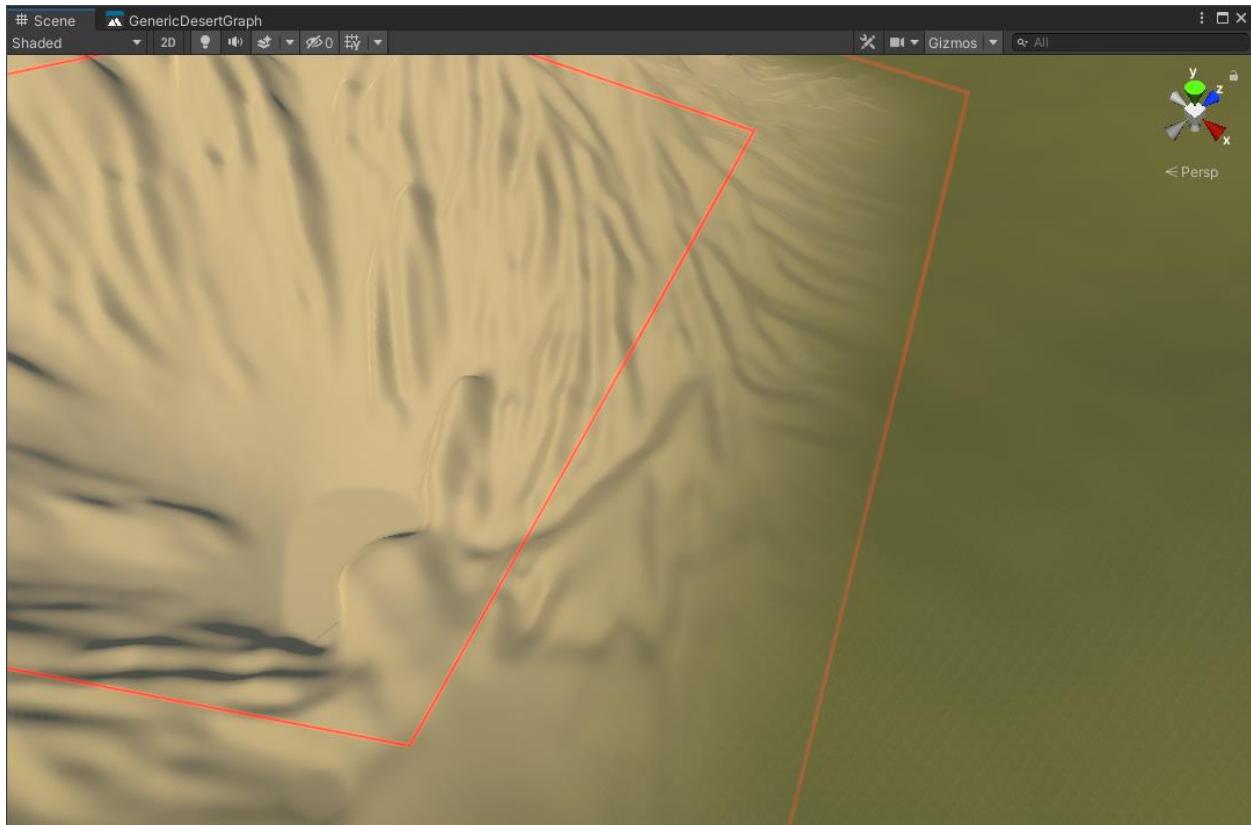
To refer to the biome mask texture, add an Graph Input node, and use “**Biome Mask Input**” (exact name without quotes) as input name. Instead of typing, you can select it from the name selector.



Since the biome mask is provided as raw with no post process, we need to do some quick effect such as blur:



Save the graph and this is what we get, way better:



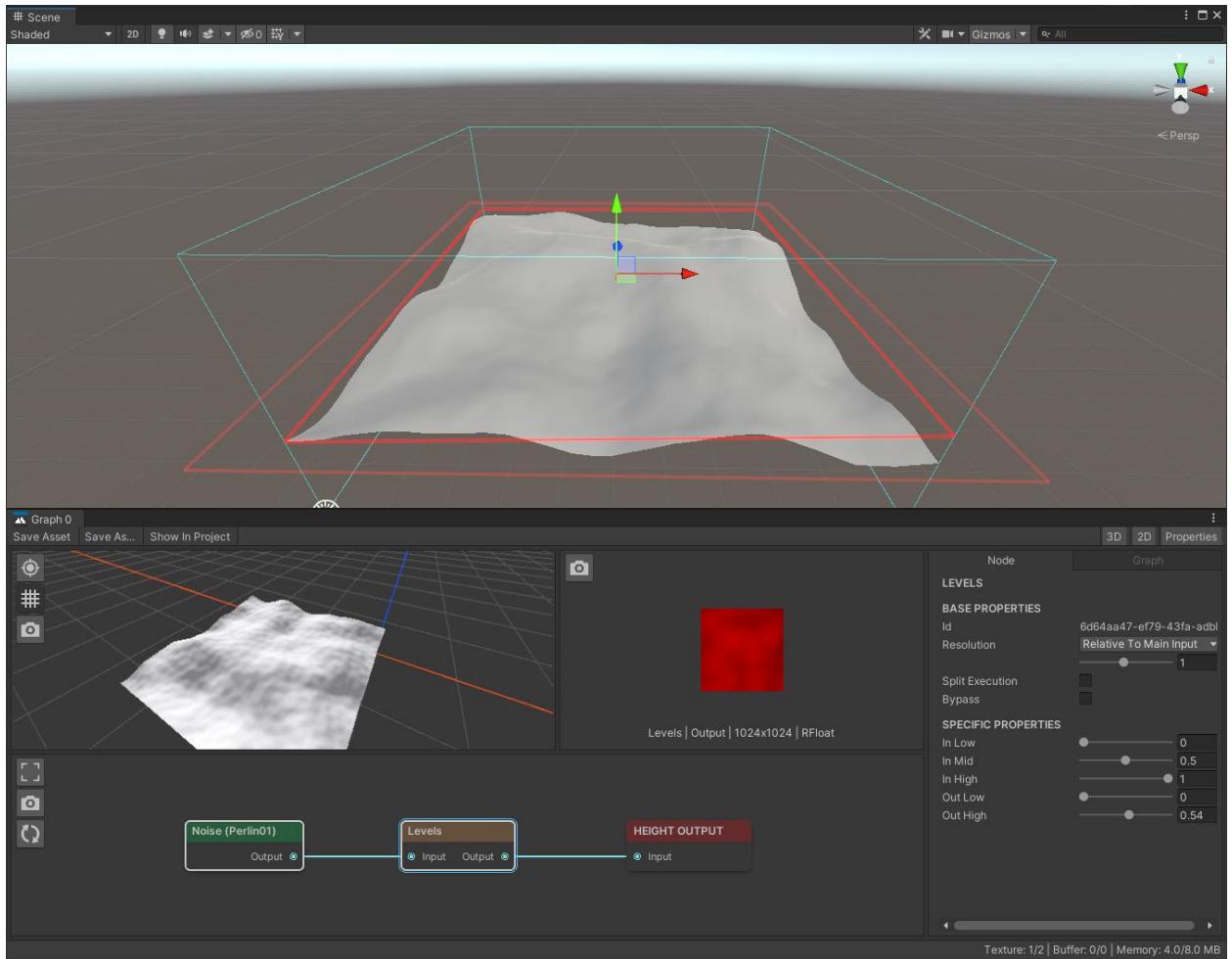
Blend modes

The biome allows you to select which blend mode to use on a specific data type. Please prefer [this section](#) for detailed specification.

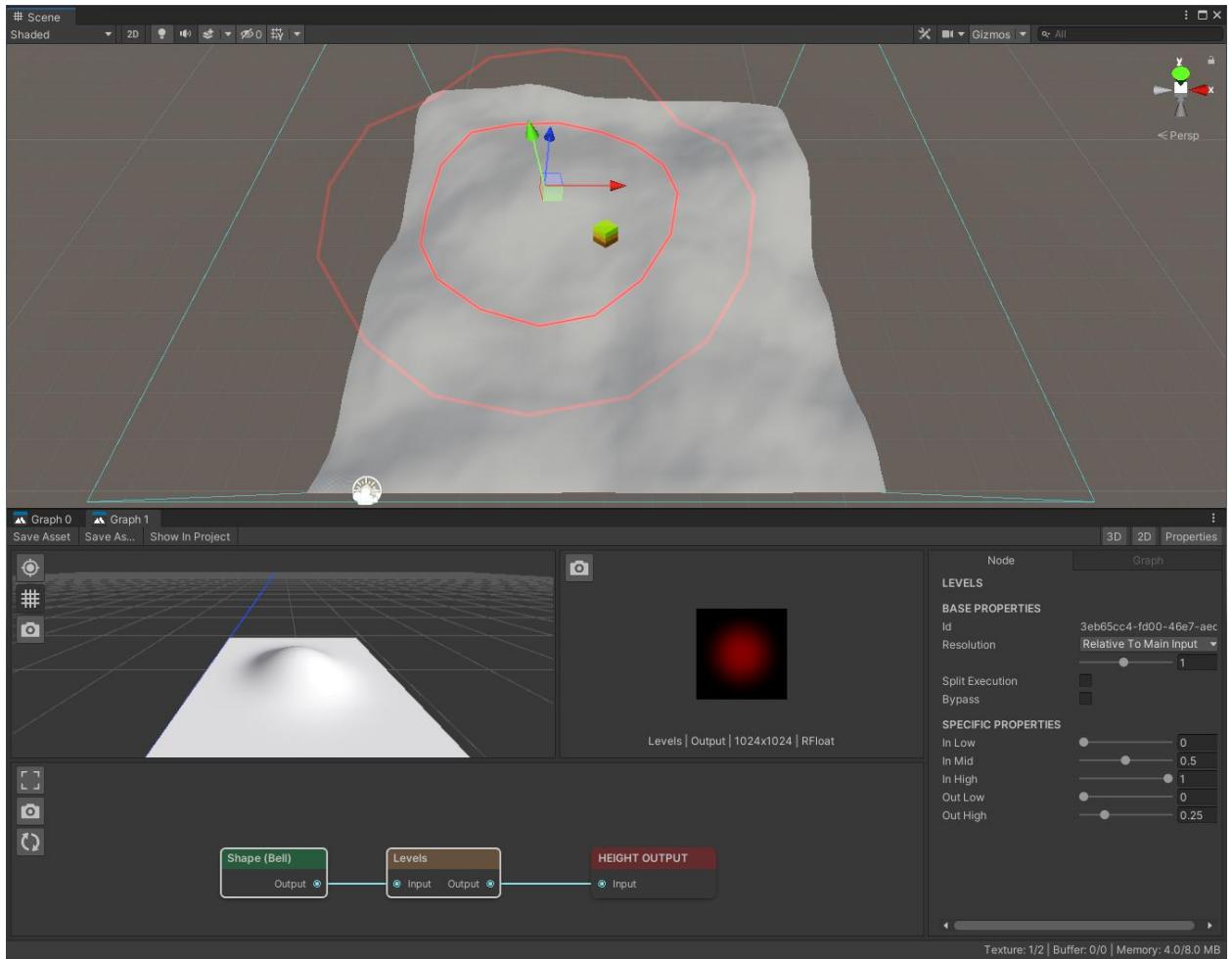
Below are some example of how to use blend mode in specific cases:

Raise or lower the terrains at a region:

First, create a “base” biome that output base geometry using the Graph 0 graph as below:

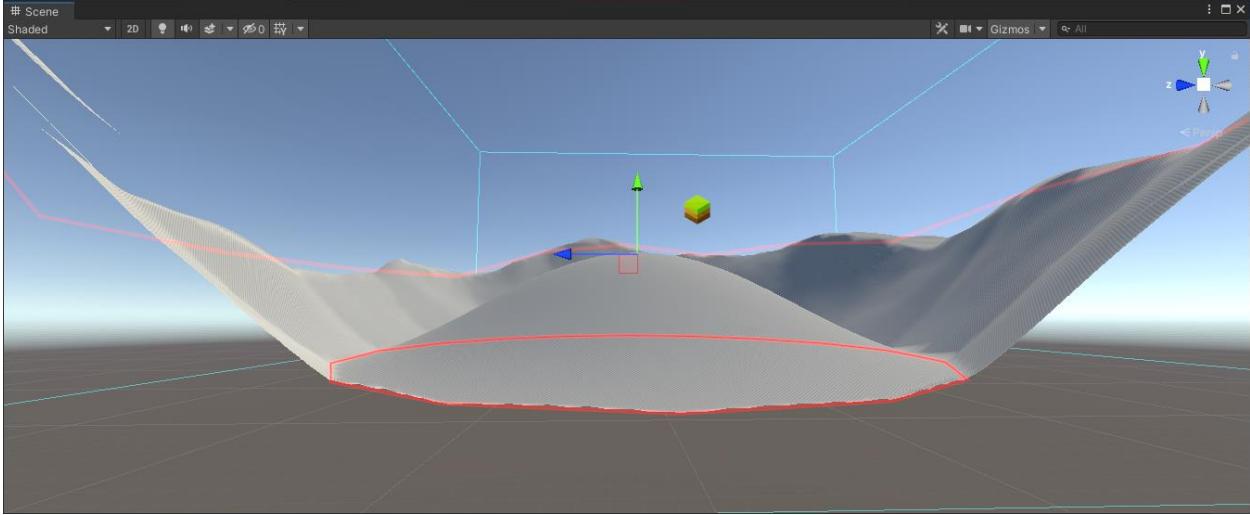


Next, create another biome with circle shape covers a part of the terrain, using the Graph 1 graph like this:



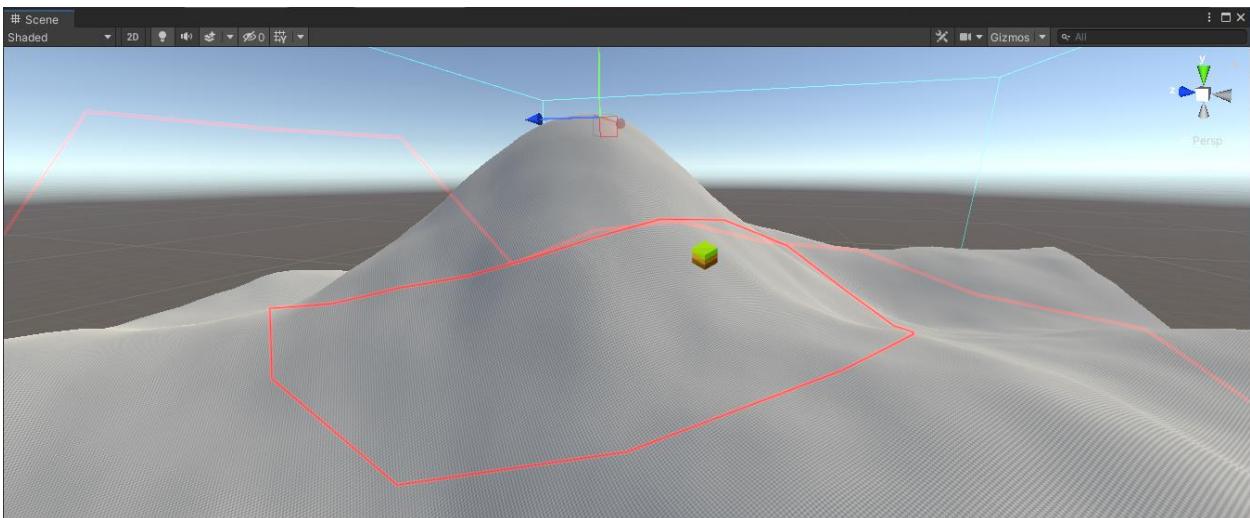
The second biome will just output a bell shape with its value remapped to $[0, 0.25]$, as we will later add/subtract the bell shape to the base terrain.

By default, height map blend mode will be set to Linear, which gradually change from one to another. that doesn't make sense in this case:

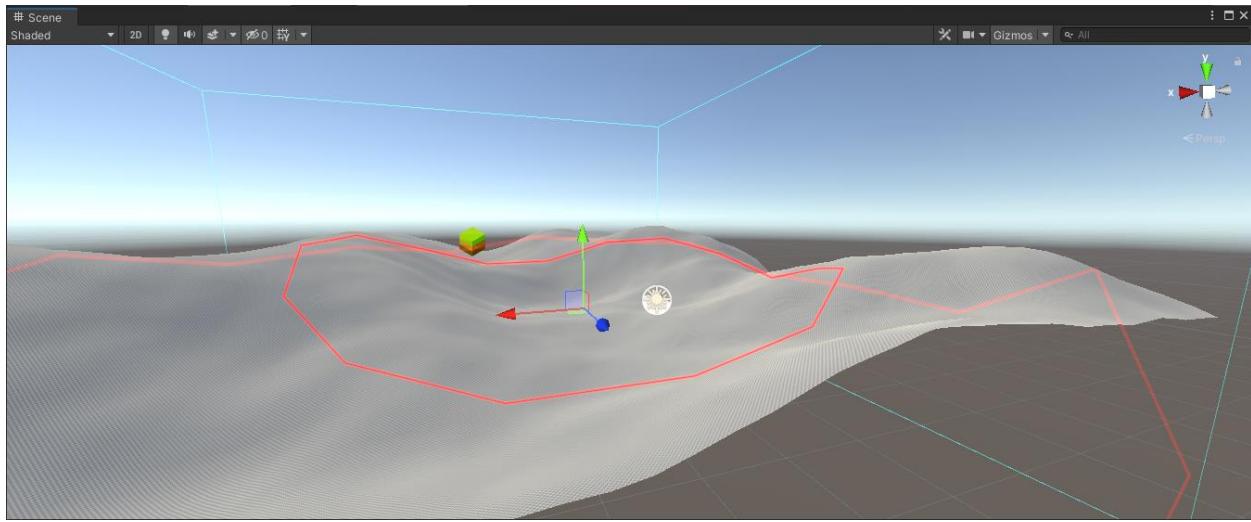


You need to set Height Map blend mode to Addictive or Subtractive for the **second biome**, the result will look like this:

In additive mode, the terrain raised up:

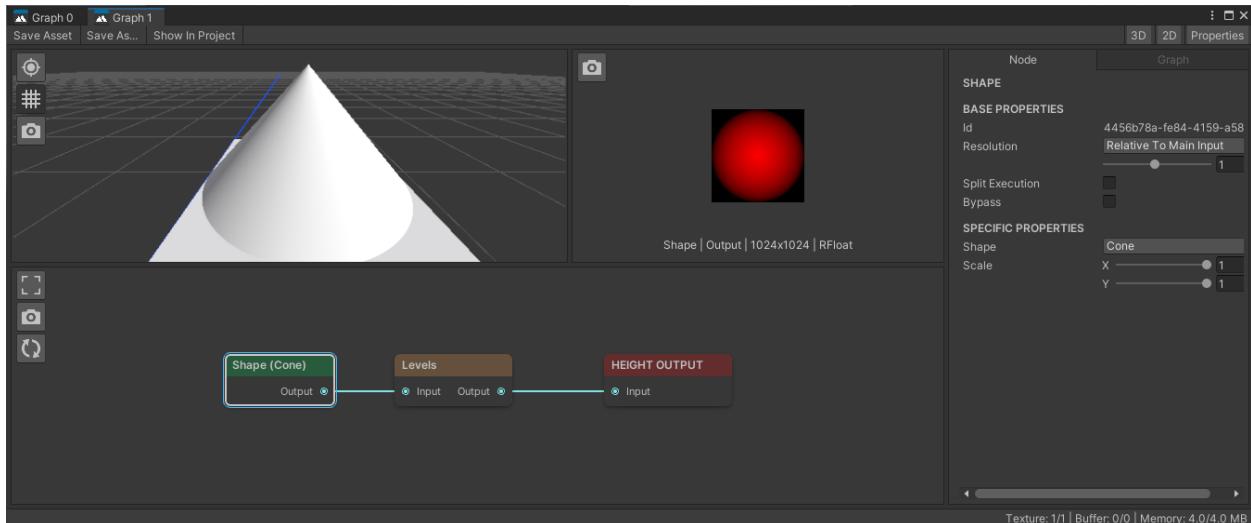


In subtractive mode, the terrain lowered down:

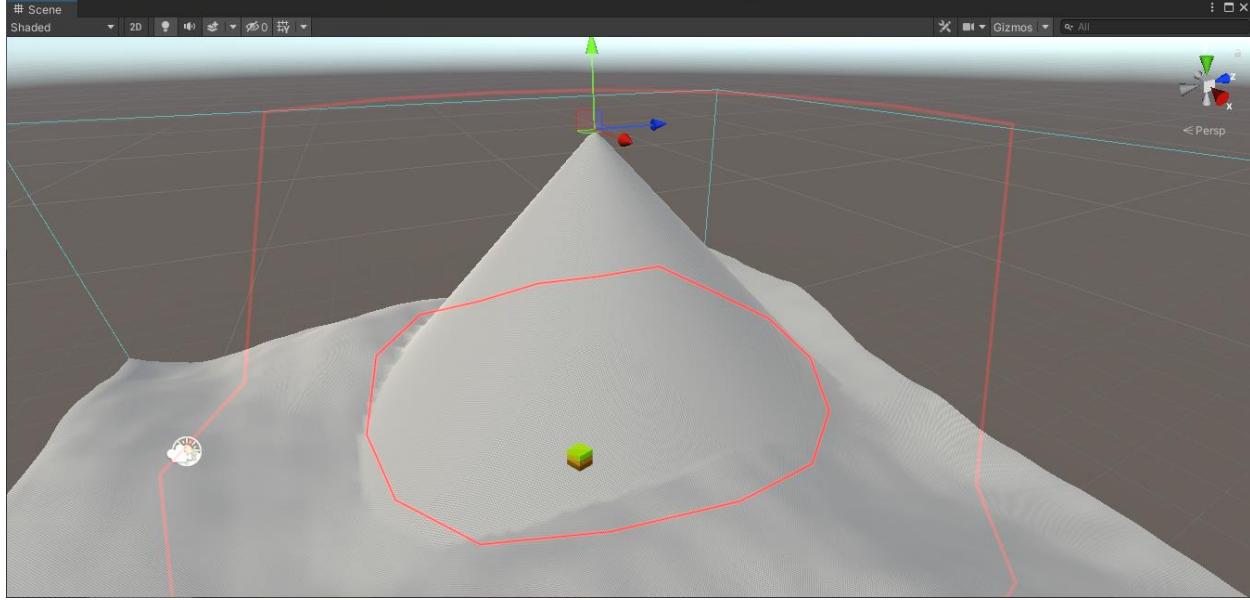


Placing a mountain

Tweak the Graph 1 graph to output a Cone instead of a Bell to see the effect better, also remap it to [0,1]



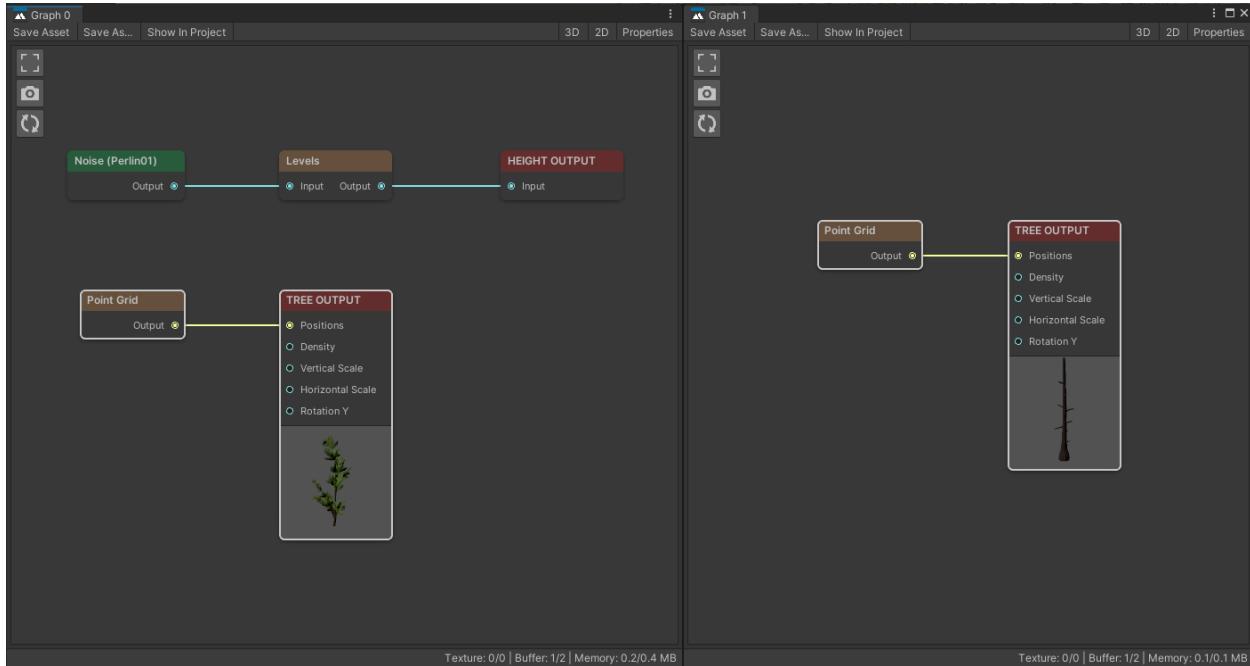
Set the Height Map blend mode on the second biome to Max, then you will see this:



Different from Addictive mode, the Max mode will choose the higher value of the base & the cone, which is suitable for interesting effects such as mountains with sharp transitions.

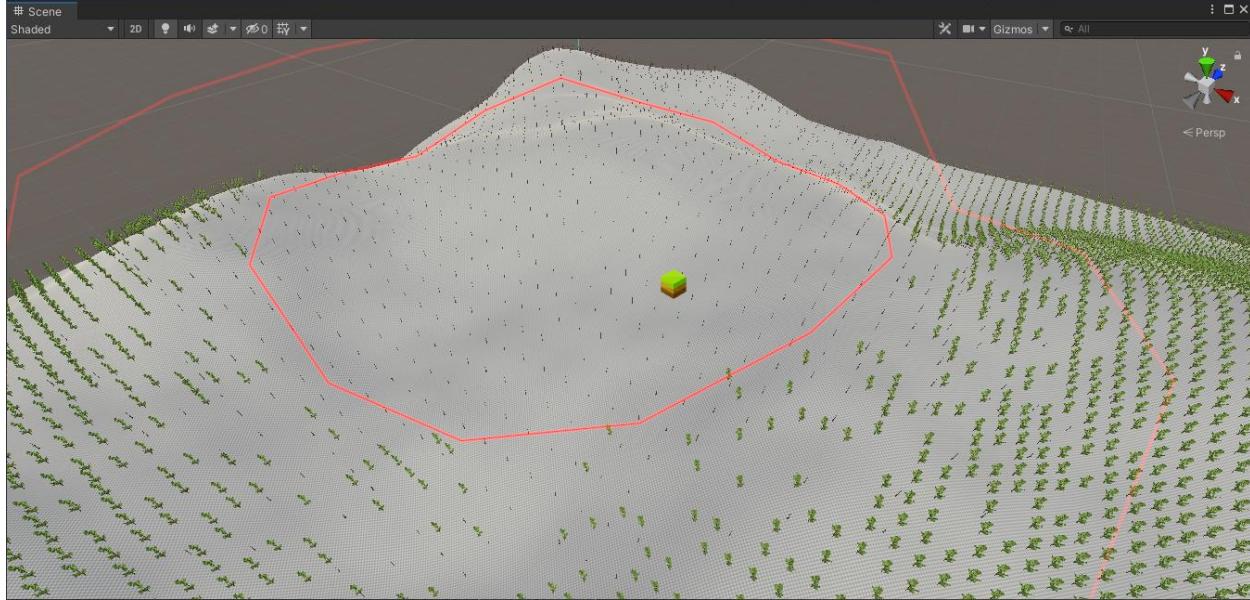
Grow additional tree species at a region

Modify the 2 graph to output some tree like this:



Also remove the Height Output Node for the Graph 1, so it's easier to see what's going on.

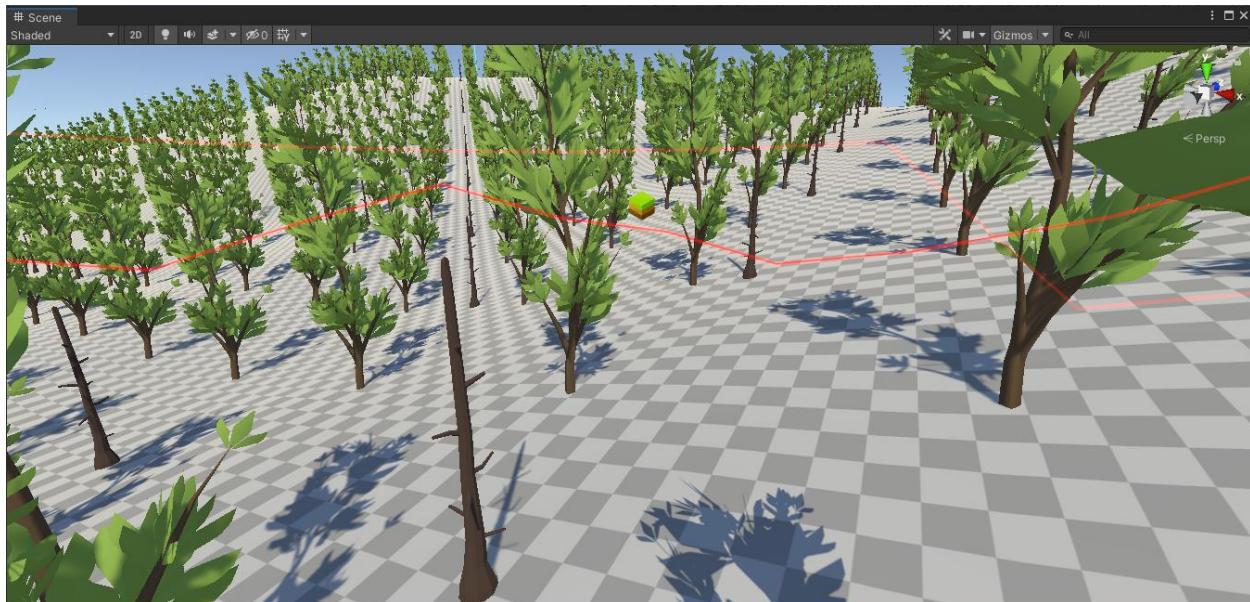
With Instances blending set to Linear (default), you will see this in the scene:



Trees from the below biome (green trees) cannot grow inside the region of the second biome.

In this example, we want to grow some dead trees at that location while keeping the existing trees there. To do so, set the Instance blend mode to Addictive for the **second biome**.

Then the two type of trees will coexist in the same region:



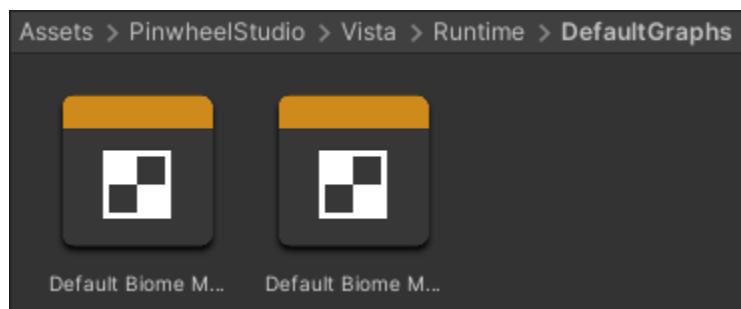
Grow additional grass species at a location

The idea is the same as the tree example above, just set Detail Density blend mode to Addictive for the second biome.

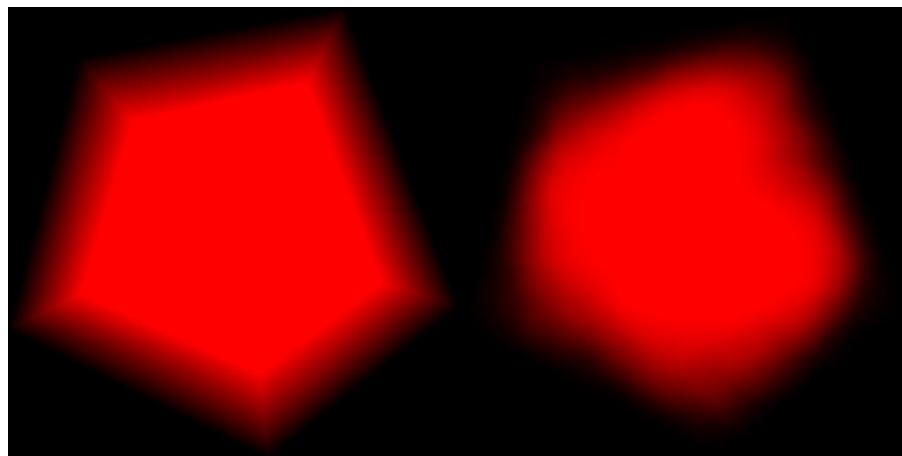
Biome Mask Graph

Biome mask graph is an asset containing rules for mask post processing, it is simpler compared to the terrain graph with less nodes, but quite enough for a fast retouching.

To create a biome mask graph, right click on the Project window, then select **Create>Vista>Biome Mask Graph**.



We provide a **Default Biome Mask Graph** asset for you to get started, located under **.../Vista/Runtime/DefaultGraphs** directory. The default graph will do some smooth and noise blend effect on your biome mask.



To edit the graph, double click on the graph asset. However, it's better to use the "Edit" button in your biome's Inspector to take advantage of contextual inputs.

Additional nodes

The following nodes are added to your library:

- Crack
- Default Value
- Flatten At
- Geometry Mask
- Gradient Map
- Landslide
- Layered Noise
- Slide
- Snow Fall
- Splatter
- Thermal Erosion
- Water Flow
- More in the future

For a detailed explanation of each node, please take a look at [this section](#).

MODULE: PRODUCTIVITY BOOST

Overview

[This module](#) offers some utility features that let you work smarter, faster and stay organized.

Version: 2023.1.0

Requirements: Vista 2023.1.0+

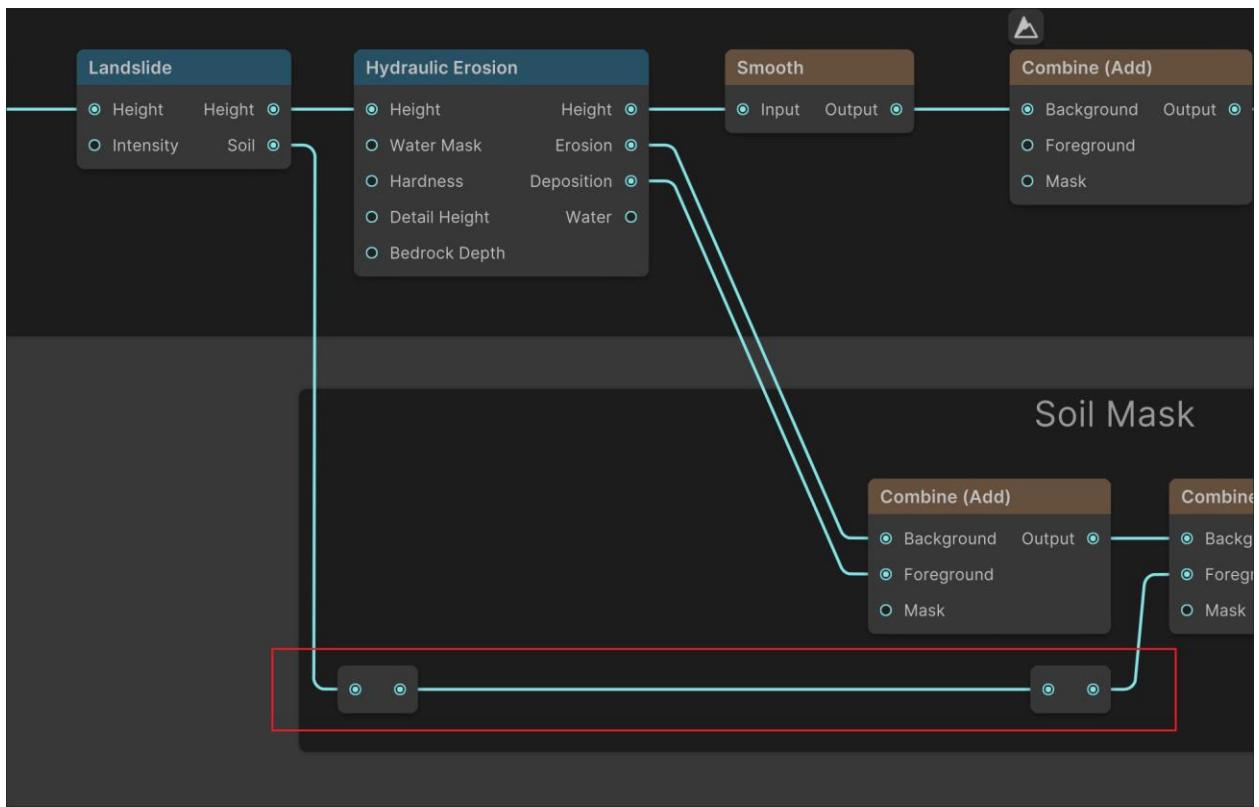
This version introduces breaking changes, please refer to the [Update guide](#) section to instruction.

Change log:

- New: Drop graph assets to the graph view to create sub graph nodes.
- Changes and improvements to be compatible with Vista 2023.

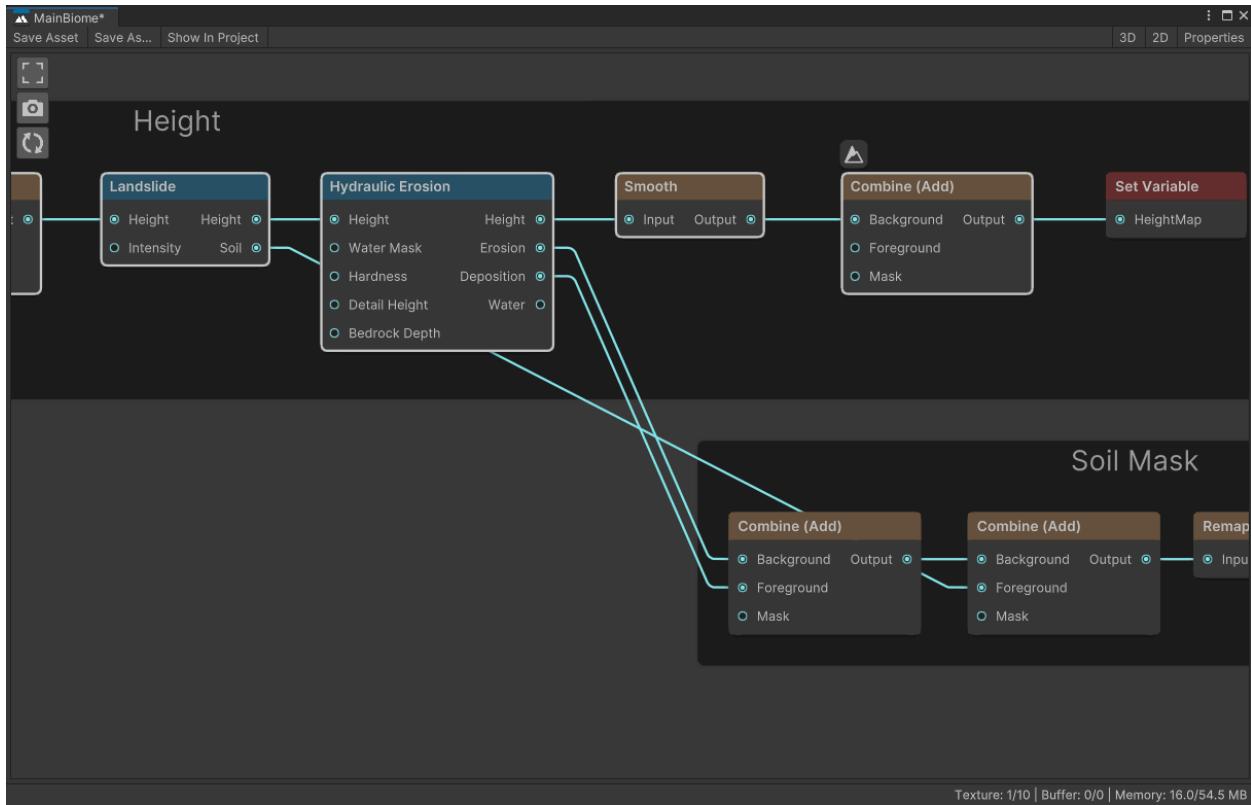
Anchors

Anchors allow you to rewire an edge to keep the graph tidy from messy connection.



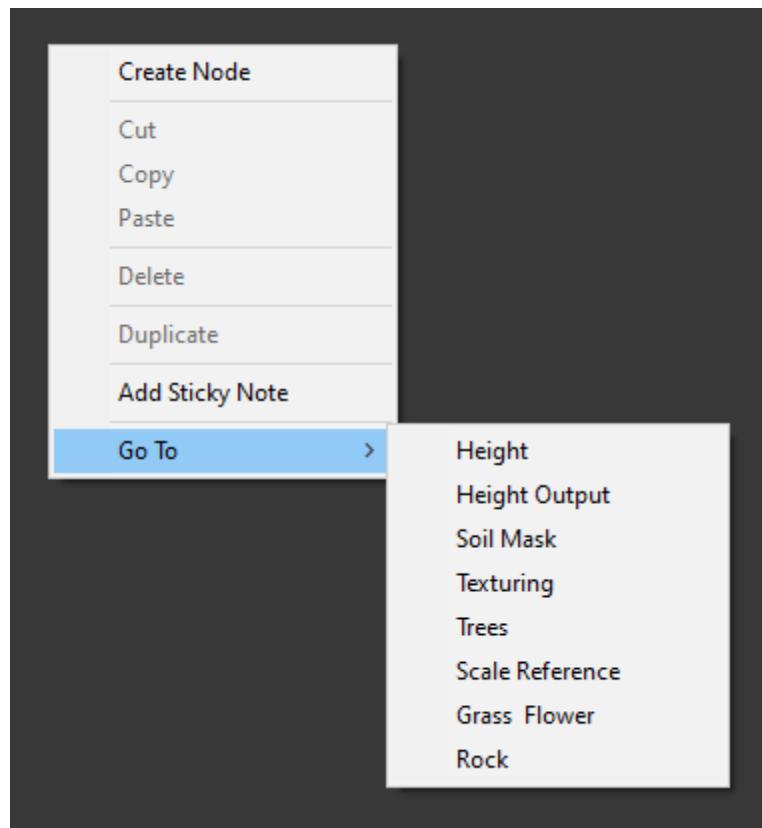
It can also be used to split an edge to multiple ones.

To create anchors, simply double click on an edge at the point you want to add.



Go to

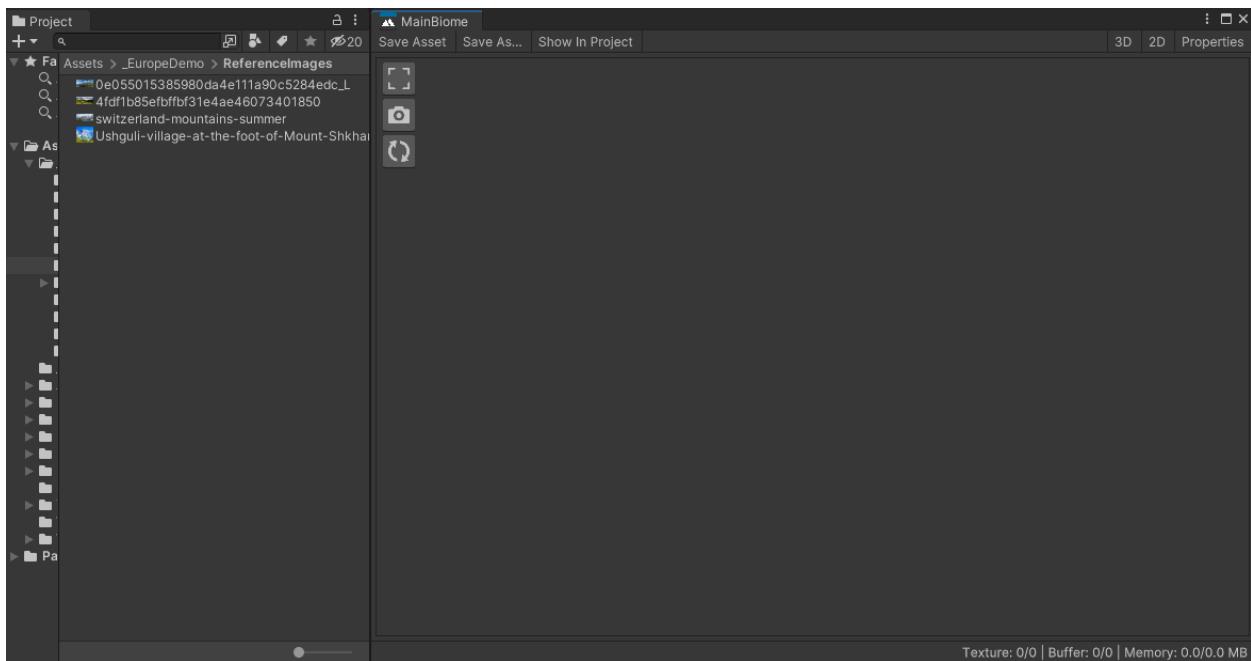
You can jump between [groups](#) in the graph by right clicking on a blank space to bring up the context menu, then select Go To.



Sticky images

You can add reference images and move them around in the graph view like other elements. This is a great way to find inspiration without leaving the editor.

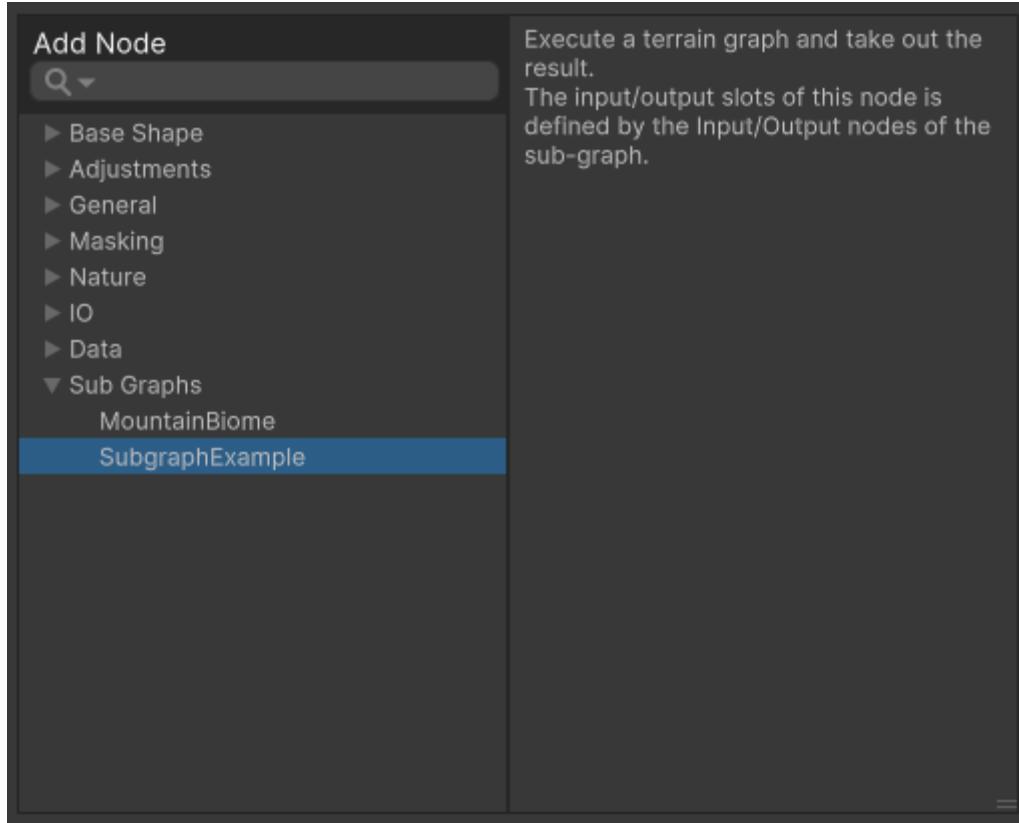
Simply drag and drop the textures from the Project window into the graph view.



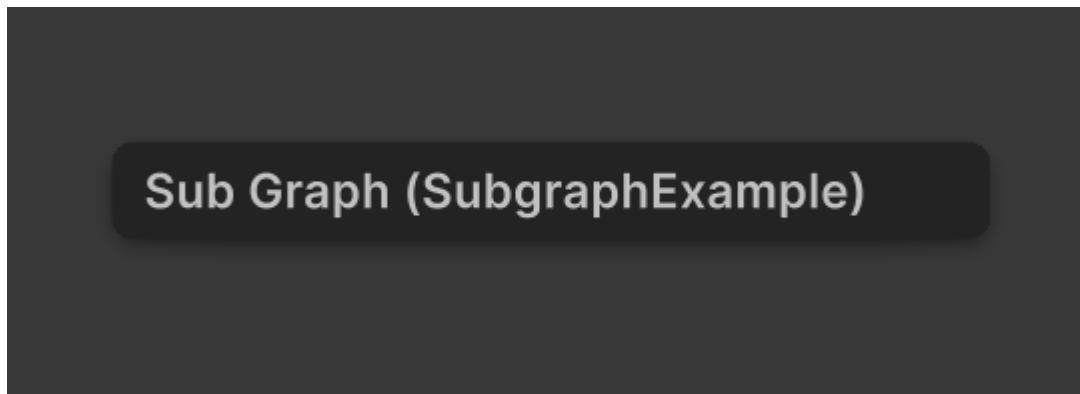
Subgraph

Subgraph allows you to split a large graph into several smaller ones. There is no specific asset type for this, a terrain graph is considered as a subgraph when you embed it into another graph.

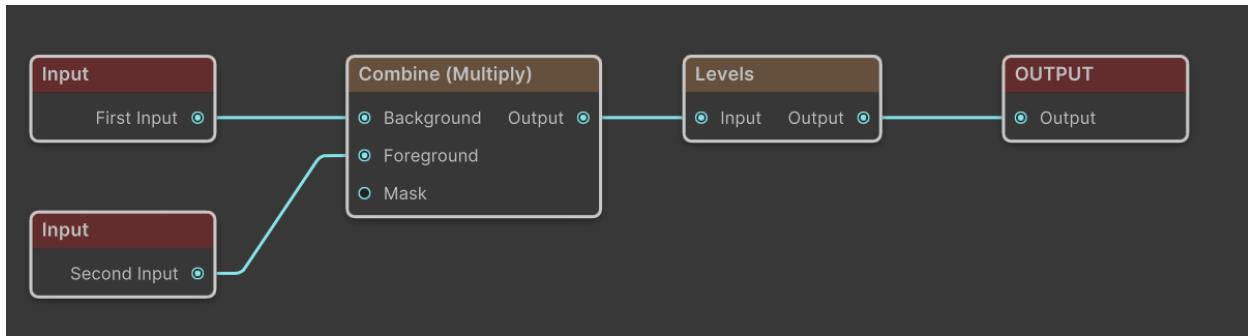
When opening the Add Node window (spacebar), you will see all terrain graph assets will be listed under Sub Graph section:



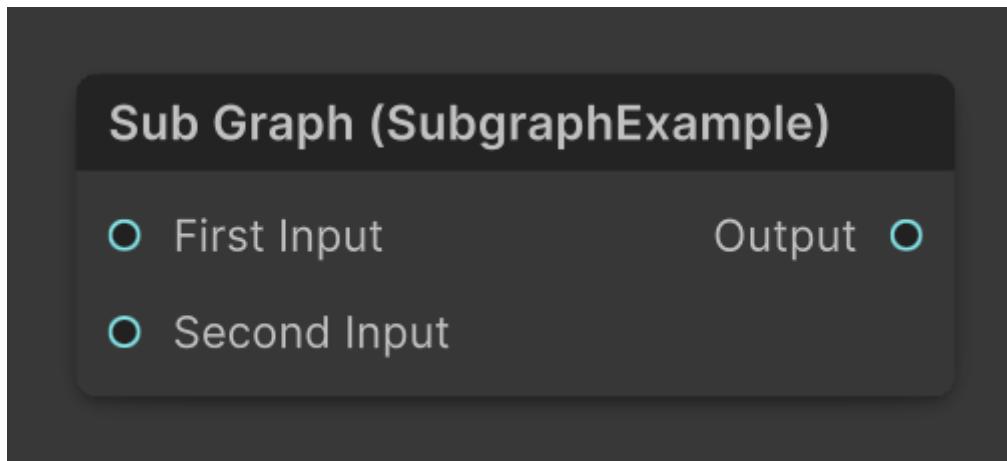
However, the subgraph node does not have any input or output port:



To add some input/output ports to it, you have to open the sub graph, then add some Graph Input & Graph Output nodes, like this:



Save the sub graph, then go back to the sub graph node:



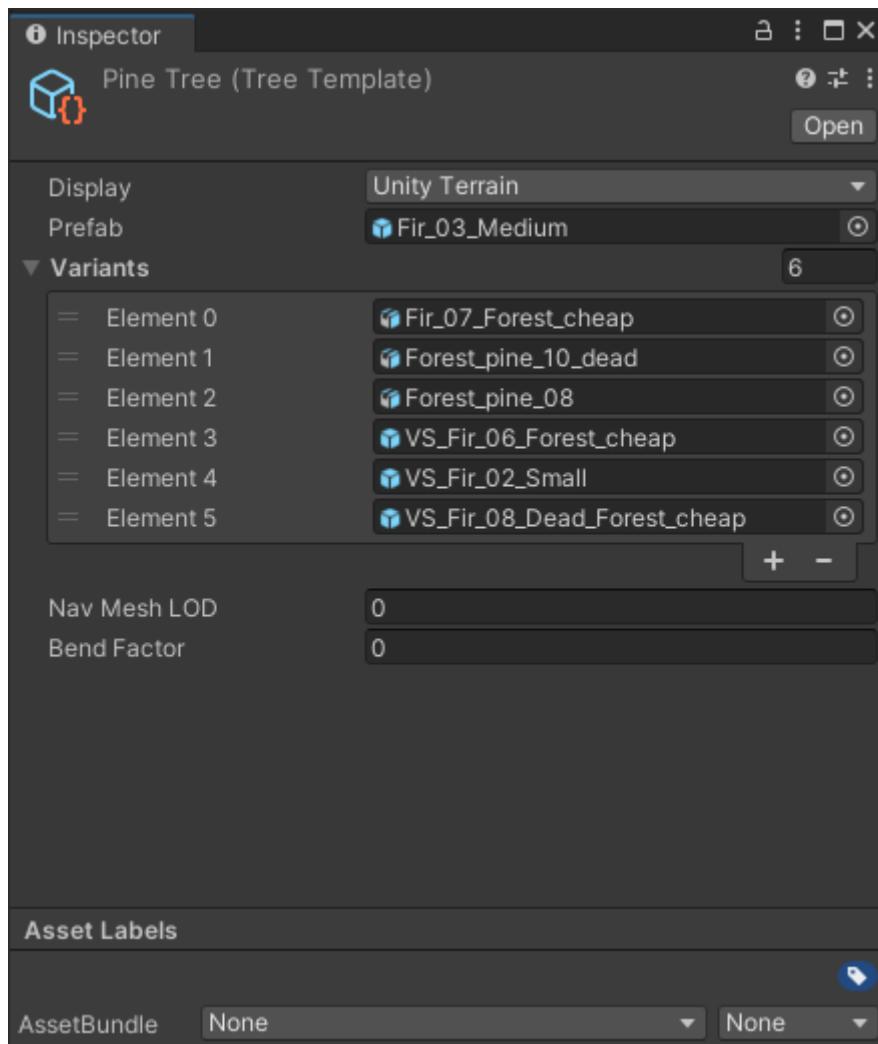
You can see the ports have been added to the node, you can interact with them as usual.

You can also drag and drop a terrain graph asset into the graph view to create a new subgraph node.

Template variants

A single tree prefab scatters all over the scene will produce a boring result. You can work around this by adding some more nodes with little modification to output another tree prefab, but that will slow you down alot.

With template variants, you can output many tree prefabs using a single graph output. Select the [Tree Template](#) asset, then you will see a Variants section to add more look-alike prefab there:



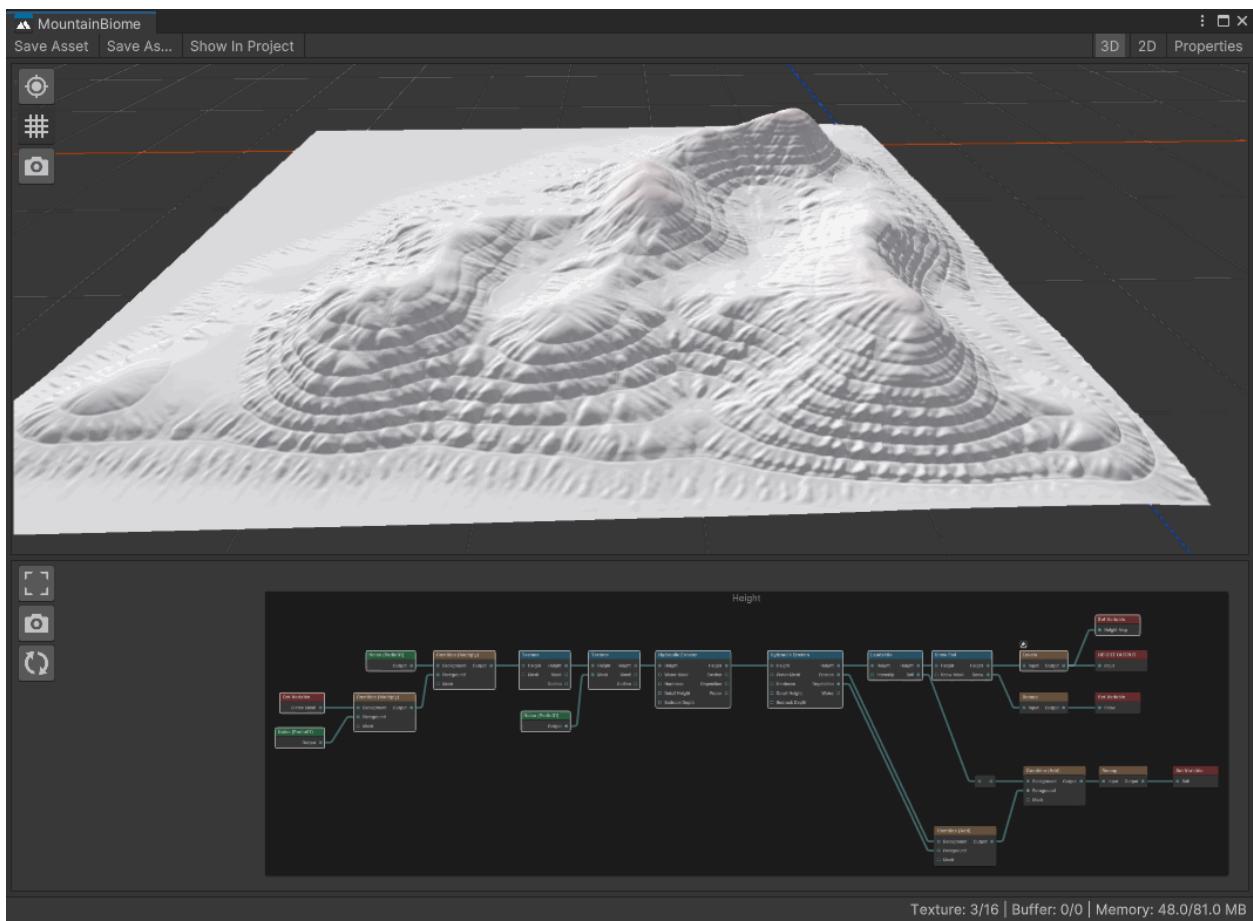
See [Vista - GPU Terrain Generator - Tree variants output](#).

This feature also supports [Detail Template](#) and [Object Template](#).

Permute graph

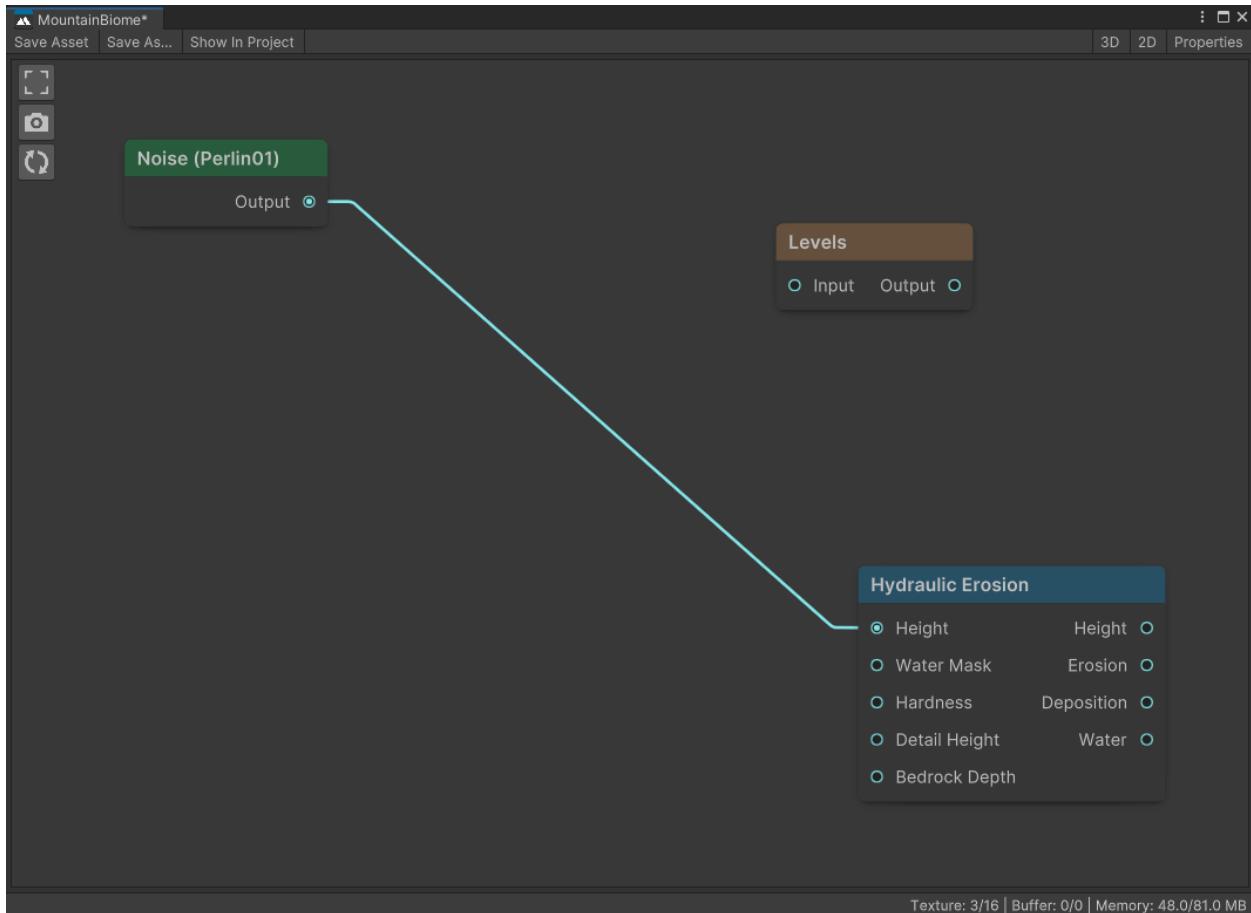
This will change the seed of every node in the graph (if they have a seed) to generate a completely new terrain for you.

In the graph view, simply click on the button.



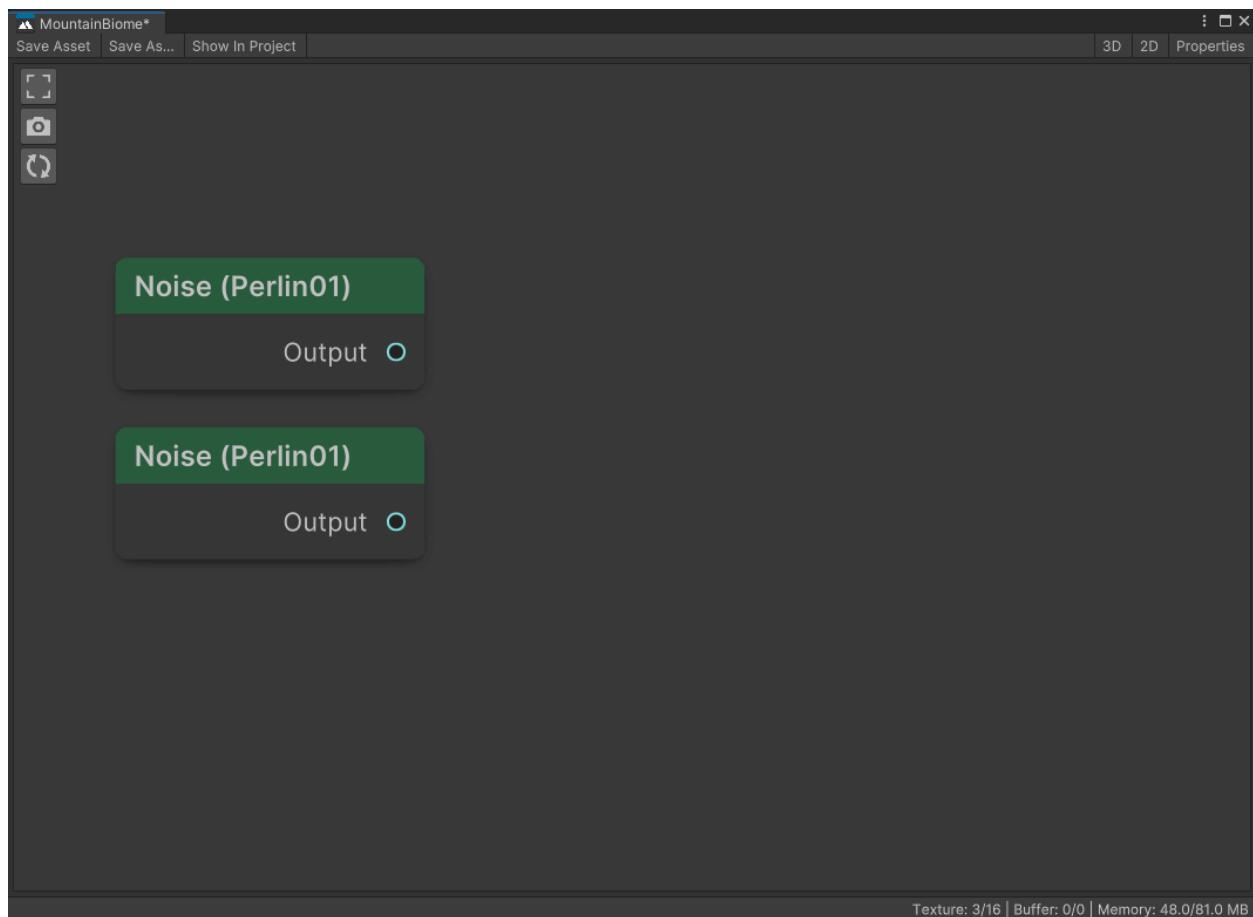
Drop to reconnect

You can drag and drop a node onto an edge to rewire the connection.



Smart search

The node browser will suggest the best suitable nodes for your keyword. You don't need to know the exact node name, just type the operation you want to do such as "add", "mul", or the noise type "perlin", "billow", or the shape name "bell", "cone", etc.



MODULE: MICROSPLAT INTEGRATION

Overview

This module allows you to synchronize MicroSplat's Texture Array Config with graph outputs, as well as outputting MicroSplat specific maps such as stream, snow, etc.

Those features only work in the editor. At runtime they take no action.

Version: 2023.1.0

This version introduces breaking changes, please refer to the [Update guide](#) section to instruction.

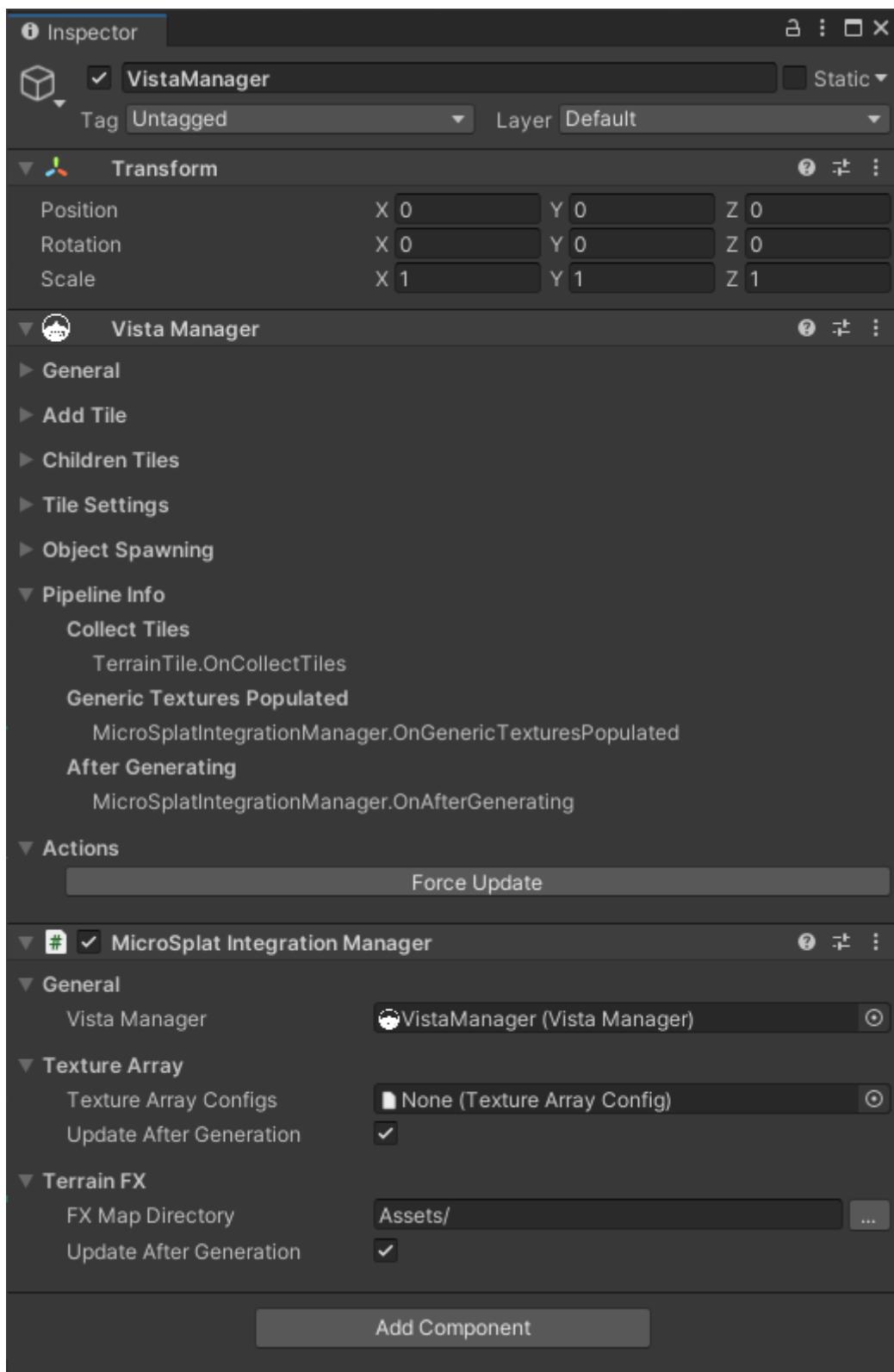
Requirements:

- Vista 2023.1.0+
- Latest version of MicroSplat.

This document assumes that you are familiar with the MicroSplat asset. Please consult its documentation for more detail.

Setting up the Vista Manager

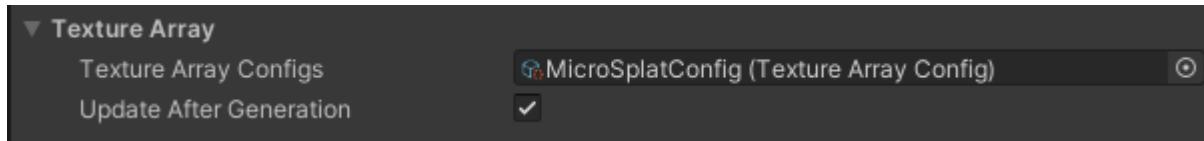
First, select the Vista Manager in the scene, then add a MicroSplat Integration Manager component to it.



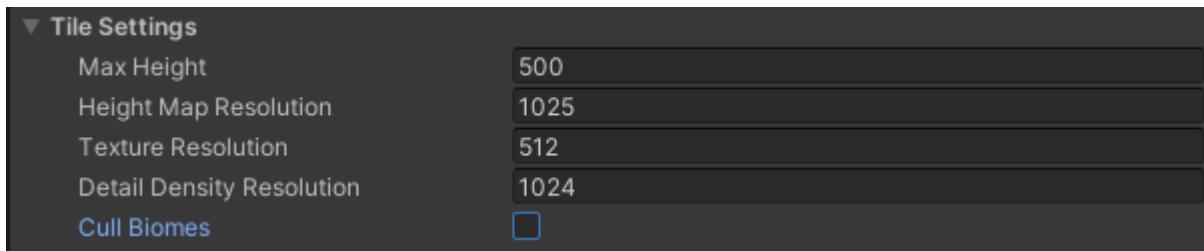
Under the General section, make sure the Vista Manager has been assigned.

Synchronize texture array

Under the Texture Array section, assign your Texture Array Config asset to its slot, and turn on the Update After Generation option.



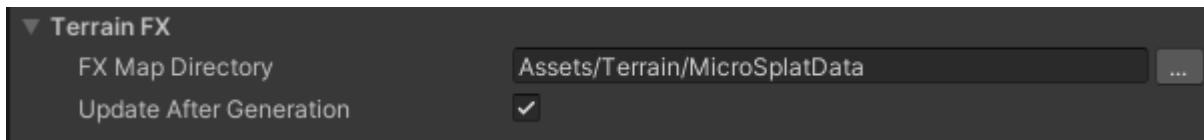
Under the Tile Settings section of the Vista Manager, make sure Cull Biomes is off:



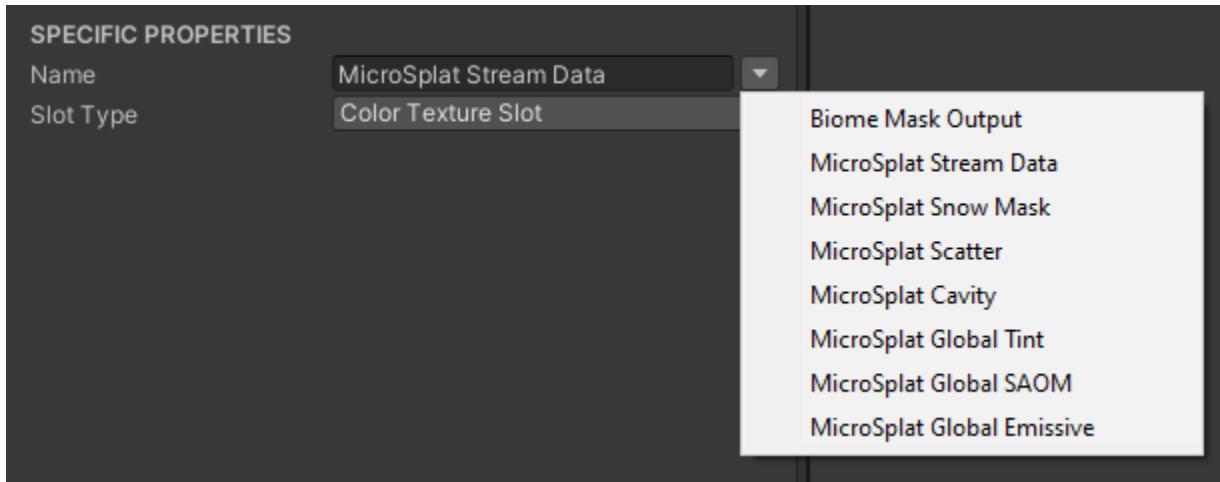
After this point, the Texture Array Config asset will update its texture list after re-generating the terrains. This action might take some time so consider turning the Update After Generation off if you don't make any change to the terrain layer.

Outputting terrain FX maps

Under the Terrain FX section, select the folder you want to store the FX maps, then turn on the Update After Generation option.



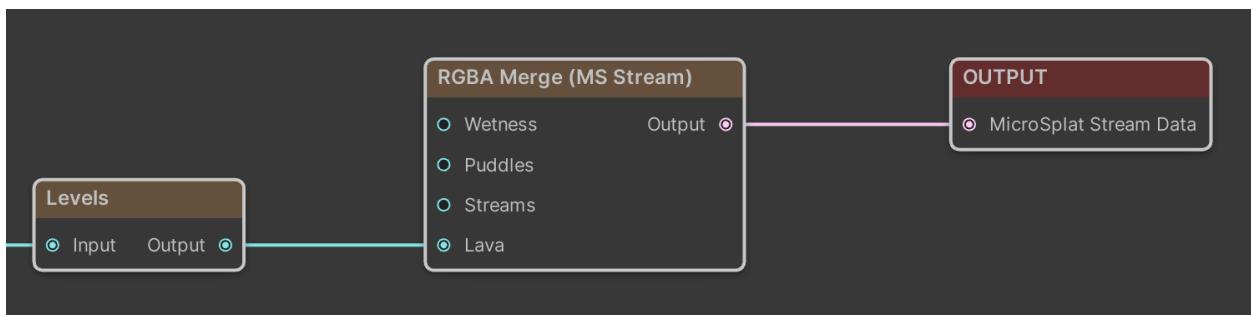
In your graph, add a Graph Output node, then use the name selector in the Properties view to pick the texture you want to output.



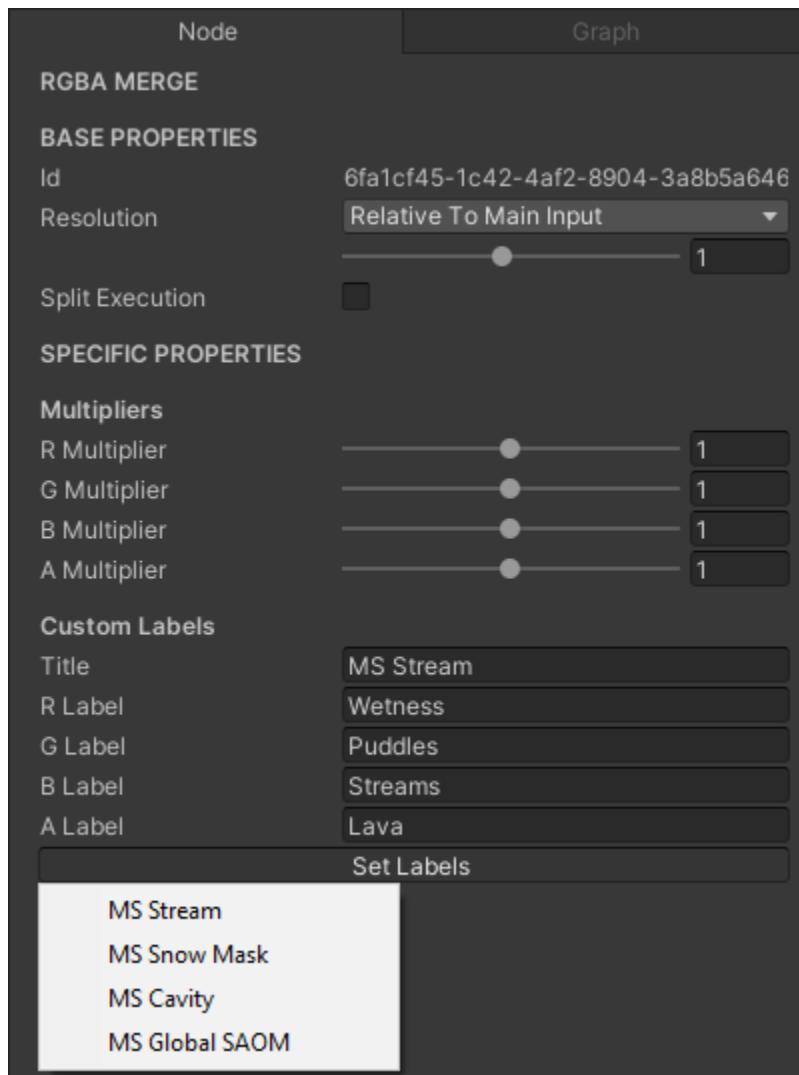
Currently the following maps are supported:

- Stream (wetness, puddles, streams, lava)
- Snow Mask
- Scatter
- Cavity
- Global Tint
- Global SAOM
- Global Emissive

Some maps may pack data in multiple texture channels, you can add a RGBA Merge node to do that:



The RGBA Merge node also allows you to set custom labels, by using the Custom Labels in its Properties view:



After this step, it will export some textures in the directory you selected above, the textures are also assigned to the correct terrain in the scene.

Note: Remember to turn a certain effect on in your MicroSplat shader compiler.

MODULE: HAND PAINTING

Overview

This module adds more functionalities to let you fine-tune your biomes as well as paint custom masks using brush.

Note that paint actions are only available in the editor, but paint results can be included in the build and work at runtime.

Version: 2023.1.0

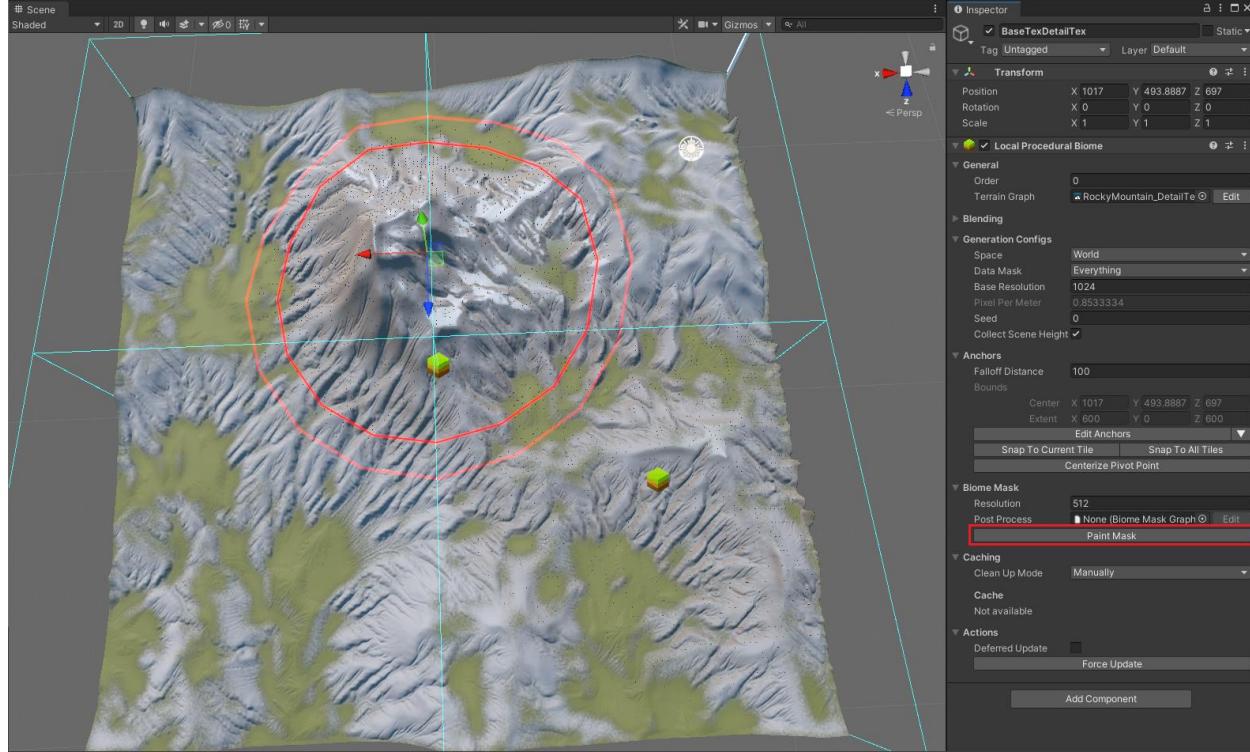
This version introduces breaking changes, please refer to the [Update guide](#) section to instruction.

Requirements: Vista 2023.1.0+

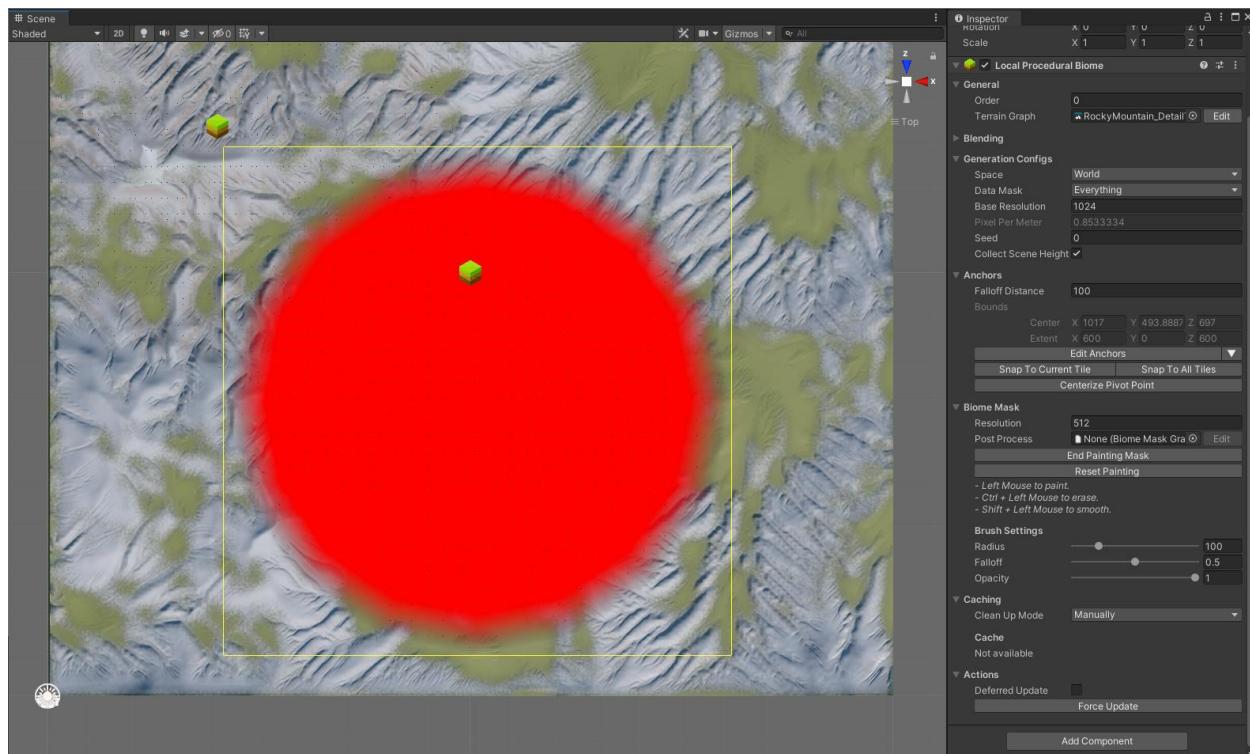
Biome mask painting

The shape of a biome is usually defined by its anchor points. With this module installed, you can add fine details & touches to the biome mask BEFORE processing it with the biome mask graph.

Select the biome, then in the Inspector click on Paint Mask:



Then you should see the Scene View will switch to the top down view. In the Inspector, there will be some brush settings:



- Radius: Radius of the brush, in meters.
- Falloff: Low value gives “hard” brush, while high value gives “softer” brush.
- Opacity: The opacity/intensity of the brush.

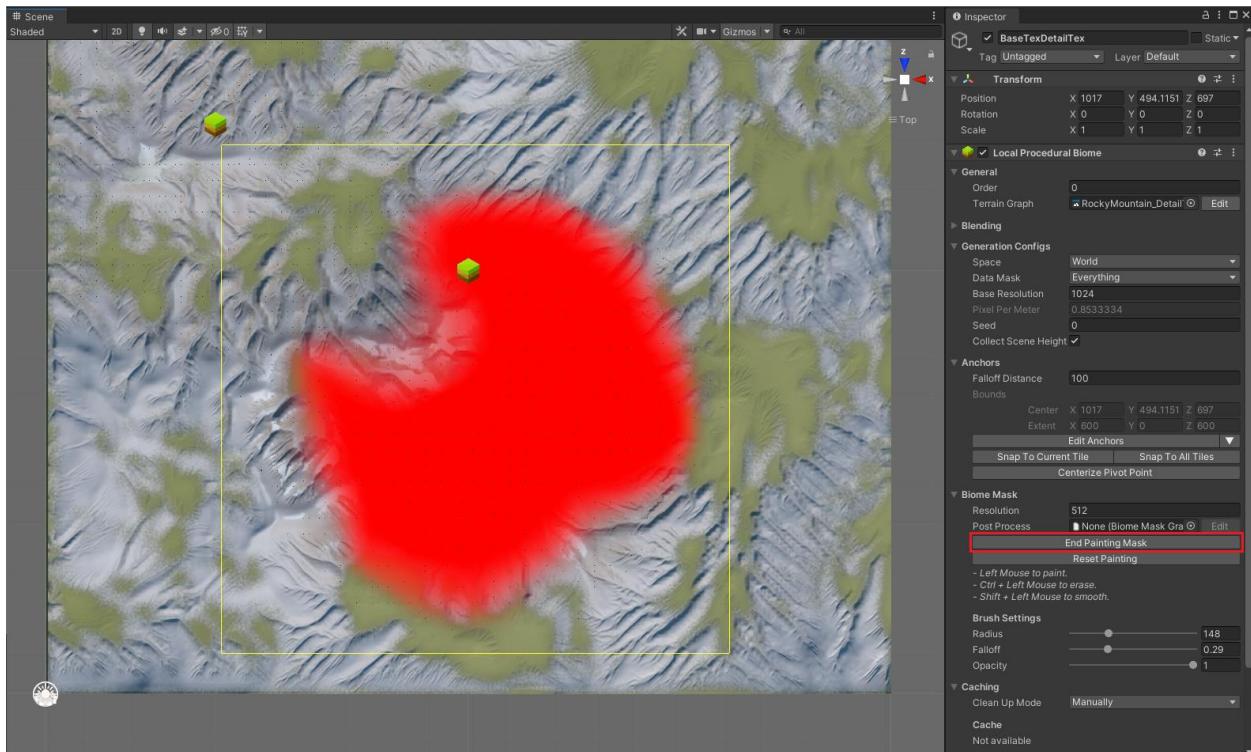
In the scene view, you will see the brush gizmos when your mouse is close to the biome’s boundary (yellow box).

- Use the left mouse to paint.
- Use Ctrl + left mouse to erase.
- Use Shift + left mouse to smooth out.

Then, click on the **End Painting Mask** button in the Inspector to save the mask and re-generate the scene.

Also, you can use the **Reset Painting** to erase all paint data and start over.

Note: Undo is not available at the moment for painting.



Paint Nodes

This module provides some nodes to perform manual painting in the graph editor.

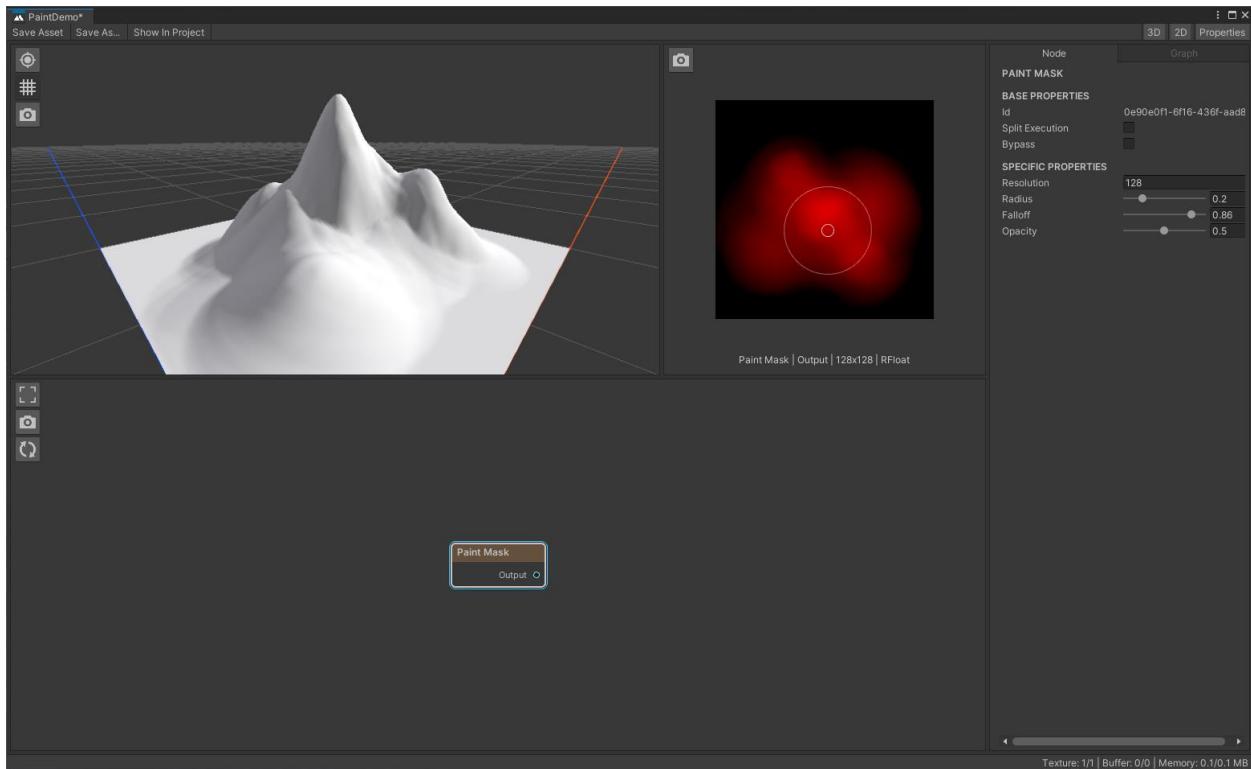
- To paint mask data (RFloat), add a Paint Mask Node.
- To paint color data (RGBA32), add a Paint Color Node.

They have similar properties:

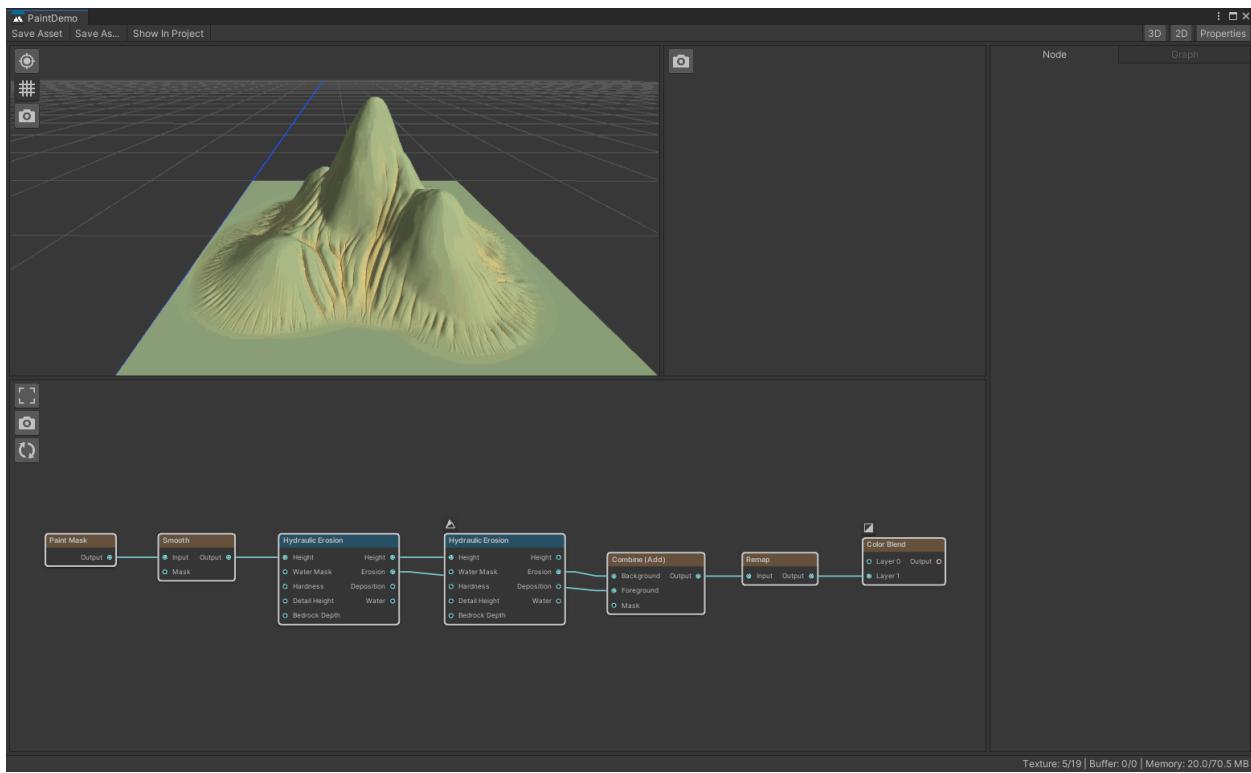
- Resolution: size of the paint canvas in pixels. **Note that the pixel data will be embedded into the graph which may lead to large file size, so choose a smallest number as possible.**
- Radius/Falloff/Opacity/Color: Regular brush settings.

When selecting the node, you can use the 2D Viewport to perform painting:

- Use the left mouse to paint.
- Use Ctrl + left mouse to erase.
- Use Shift + left mouse to smooth out.



The paint nodes are simple, but powerful. Below is an example of how to setup a node chain for painting terrain with erosion and color:



MODULE: SPLINES

Overview

This module allows you to use existing spline solutions to perform actions such as terrain conforming, make ramp, paint road & path, spawn/remove vegetation, etc. Whatever you can imagine with a spline.

Currently the spline tools being supported are:

- Unity's Splines package (`com.unity.splines`), free, requires Unity 2022.1+, installed via the Package Manager.
- Polaris Spline Tool: included in Polaris 2021. This tool itself only works for Polaris terrain, but with this module it also works with Unity terrain or any terrain system.
- [Curvy Spline 8](#) from ToolBuddy.
- [Dreamteck Spline](#) from Dreamteck.
- [Bezier Solution](#) from Yasirkula. In this document we will refer to it as "Yasirkula Spline" to be more specific than the common name "bezier spline".
- Your own spline tool by implementing the `ISplineEvaluator` interface, or deriving from the `SplineEvaluatorBase` class. Please contact via Discord if you need help on this.

Note: This module doesn't provide a spline placement tool, you will need at least one of the tools above.

Version: 2023.1.0

Requirements: Vista 2023.1.0+

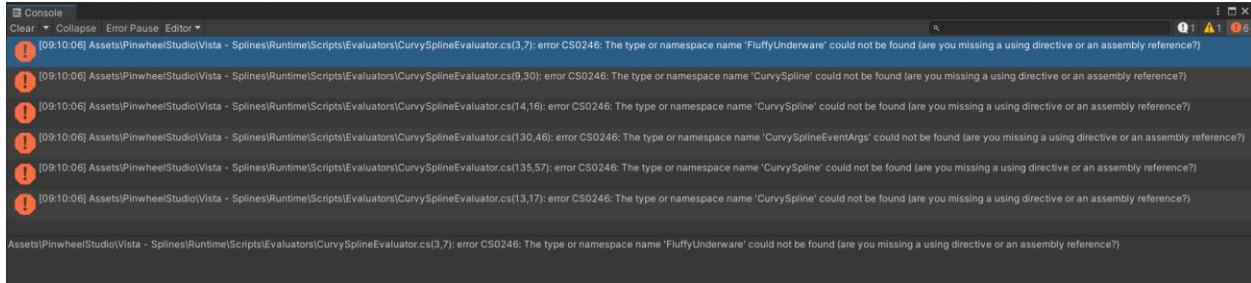
Installing a spline placement tool and creating a spline

As usual, you install a spline placement tool using the Package Manager. Each spline tool has a different way to add a spline point, adjusting tangents, etc. Please refer to the specific tool documentation for detail.

Setting up for Curvy Spline 8

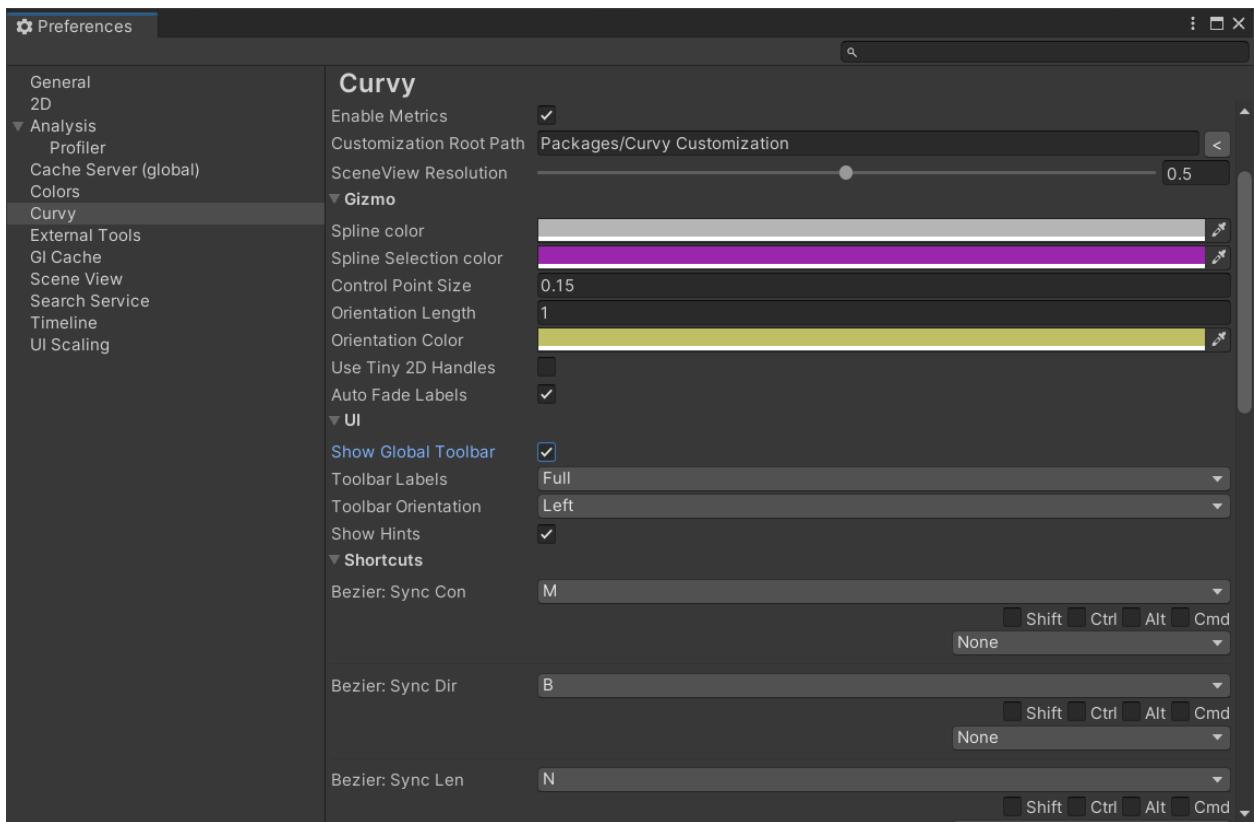
Please read this section if you're using the Curvy Spline 8 asset.

After importing Curvy Spline 8, you will see the following error:

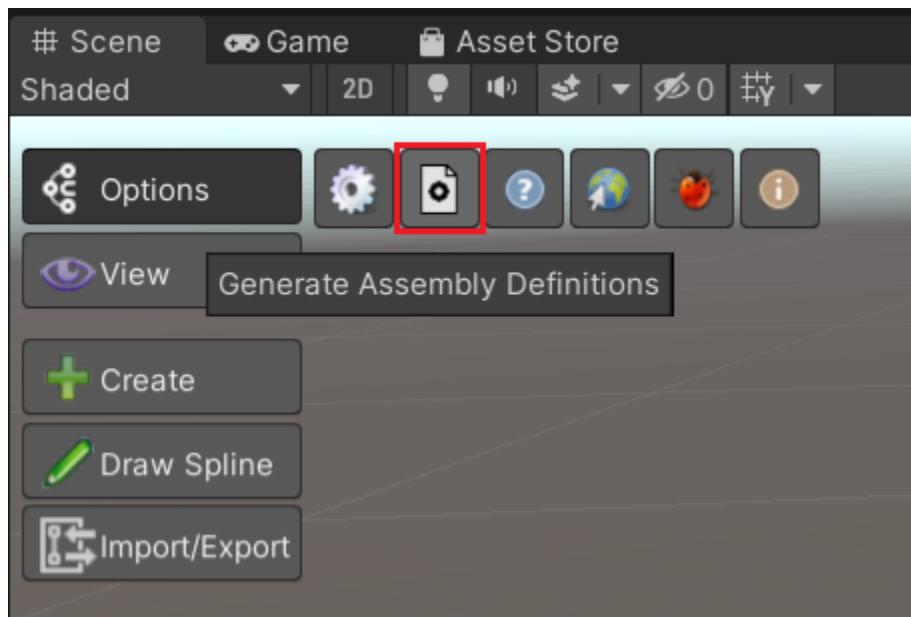


To resolve them, please do the following steps:

1. Go to **Edit>Preferences...>Curvy** and enable **Show Global Toolbar**



2. In the Scene View, select **Option>Generate Assembly Definitions**



3. Let it compile.

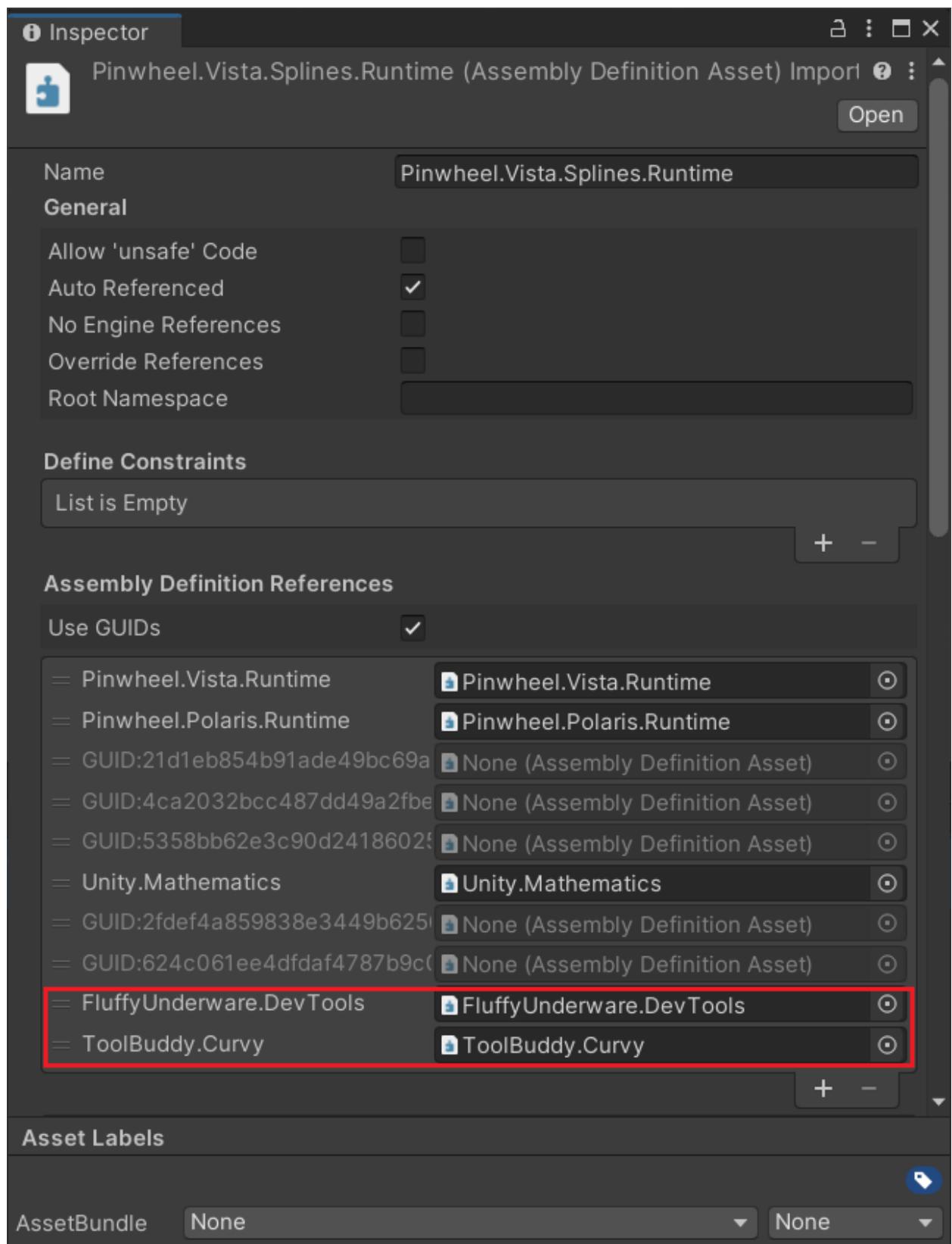
4. Select the assembly definition at **Assets/PinwheelStudio/Vista - Splines/Runtime/Scripts/Pinwheel.Vista.Splines.Runtime.asmdef**

Under the Assembly Definition References, add 2 assemblies:

- FluffyUnderware.DevTools
- ToolBuddy.Curvy

Click

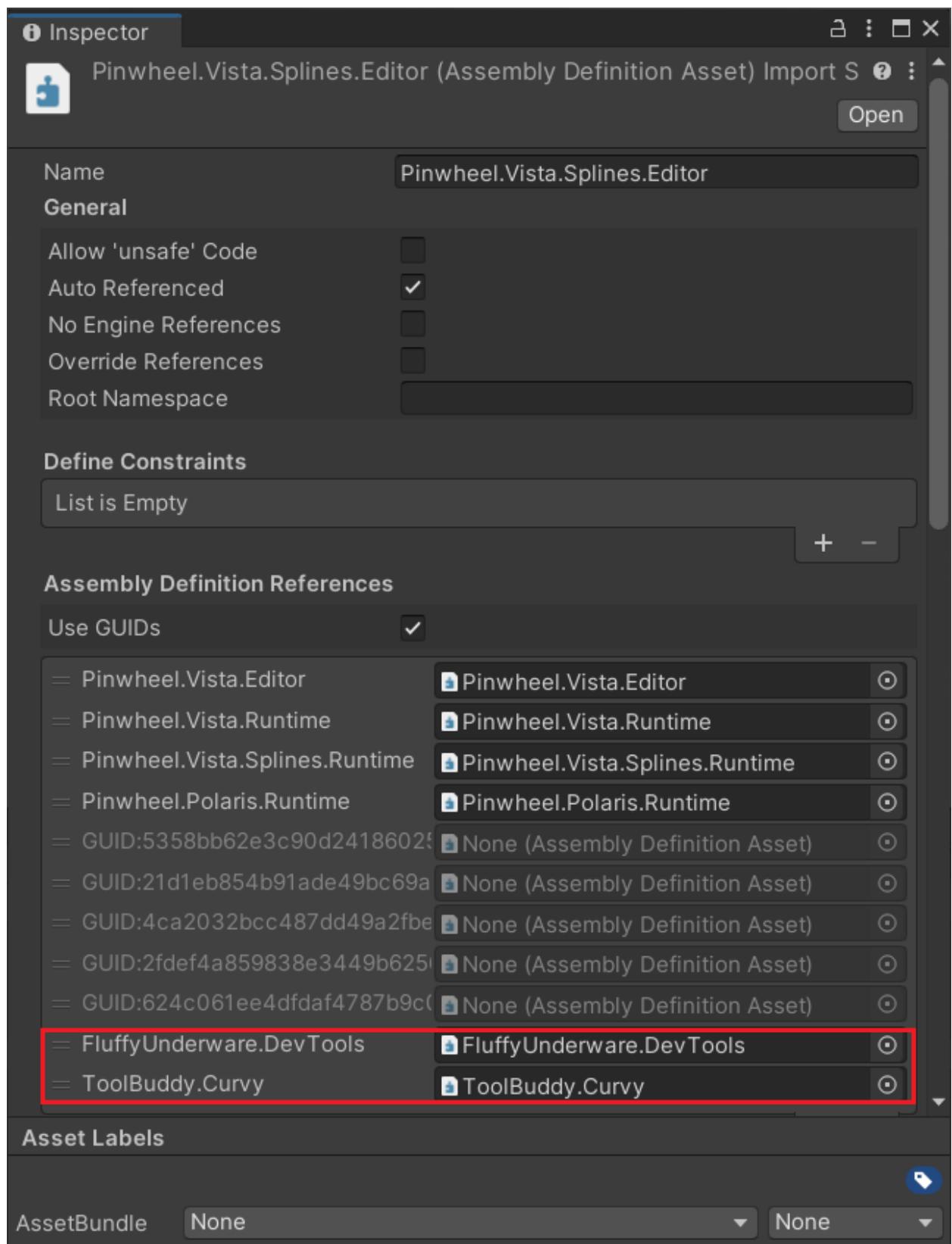
Apply.



5. Select the assembly definition at **Assets/PinwheelStudio/Vista - Splines/Editor/Scripts/Pinwheel.Vista.Splines.Editor.asmdef**
Under the Assembly Definition References, add 2 assemblies:
- FluffyUnderware.DevTools
- ToolBuddy.Curvy

Click

Apply



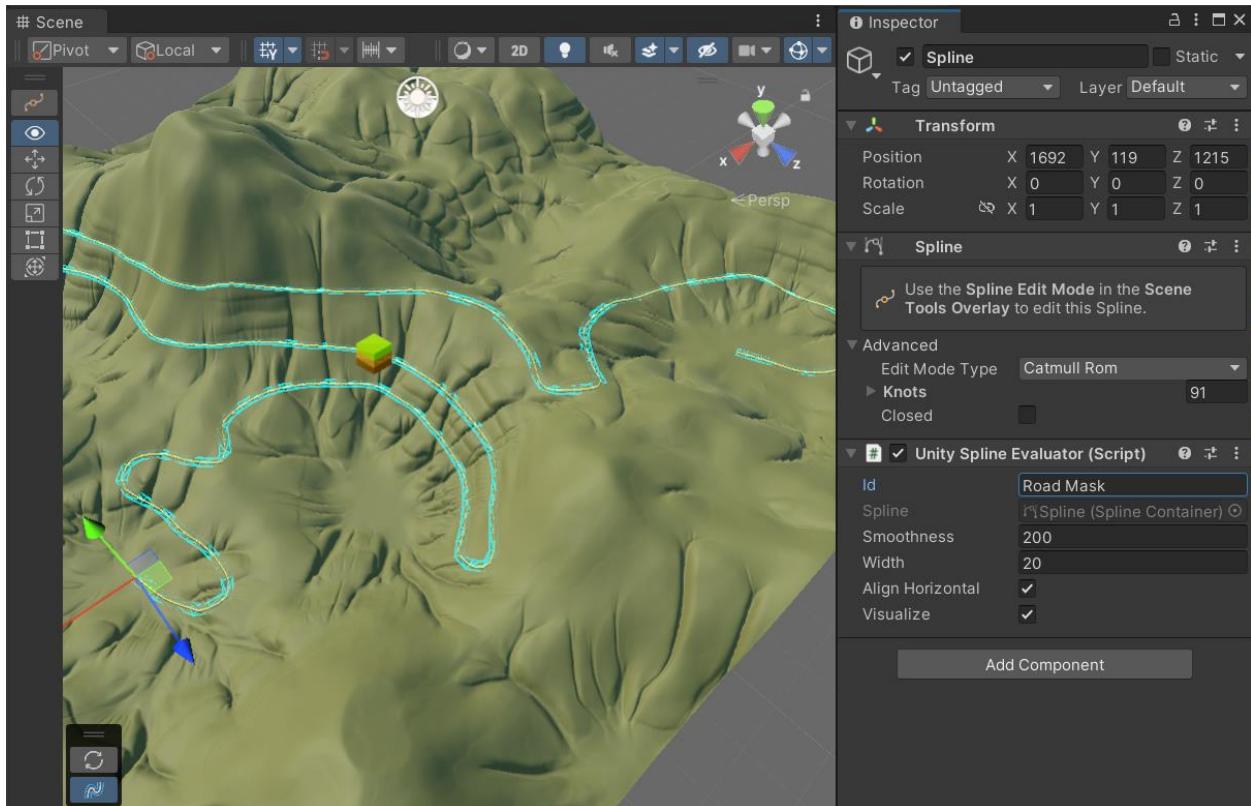
After this step, there should be no script error in the console.

Adding proper Spline Evaluators

For Vista to recognize and read data from the spline object, which comes from many third party tools out there, you need to add proper Spline Evaluator components to the object, depending on which spline tool you are using:

- Unity Splines: add a Unity Spline Evaluator component.
- Polaris Spline Tool: add a Polaris Spline Evaluator component.
- Curvy Spline 8: add a Curvy Spline Evaluator component.
- Dreamteck Spline: add a Dreamteck Spline Evaluator component.
- Yasirkula Spline: add a Yasirkula Spline Evaluator component.

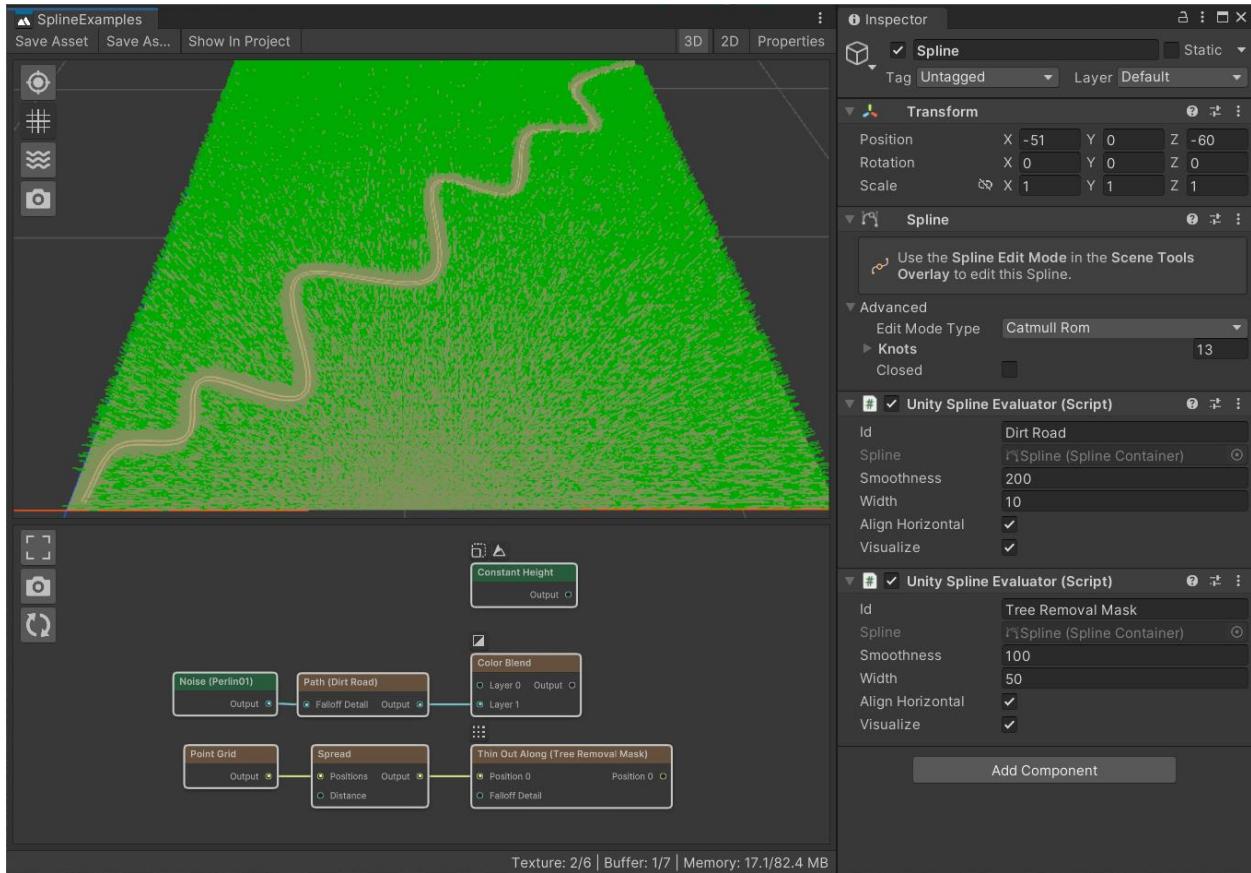
From now on, we will use Unity Splines for example.



A Spline Evaluator will have the following properties:

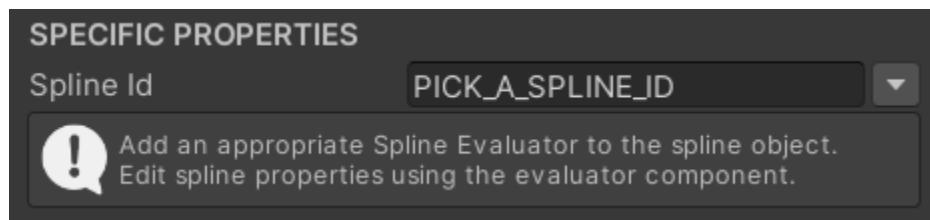
- Id: An id (should be unique) assigned to the evaluator. This id will then be used to refer to the spline inside the graph editor. It's best to use an id that describes how the spline will be used, such as "Road Mask" or "Vegetation Removal Mask", etc. **If id is empty, the spline will be ignored from all operations, including reference from the graph editor, draw gizmos, etc.**
- Spline: The component that holds the spline and provides spline evaluation methods. This will be assigned automatically in most cases.
- Smoothness: Decide the detail level of the spline, higher value will generate more vertices internally. For most spline tools, this value is "per-spline object", but for Polaris spline it is "per-segment". Because of this, the Polaris spline will use a much smaller value.
- Width: Width of the spline, aka: how far the spline will expand from any point along the normal vector at that point.
- Align Horizontal: This will try to "rotate" the spline mesh so it aligns with the horizontal plane, useful when making a ramp where you don't want a very steep angle when the spline takes a sharp turn. Note that this only affects the internal spline mesh, it doesn't modify the rotation value of spline points.
- Visualize: A global value indicates that it should render the spline mesh in the Scene View or not. On Unity 2021.2+, there is also a button in the Scene View Overlay for this.

You can add multiple Spline Evaluator components to the same spline object to perform different actions using the same spline. Below is an example of painting a dirt road and removing tree instances along its way with a greater distance.



Refer to the spline in the graph editor

All spline-related nodes will have a Spline Id field, you can manually type in or use the selector by clicking on the arrow button next to the text field.



Spline related nodes

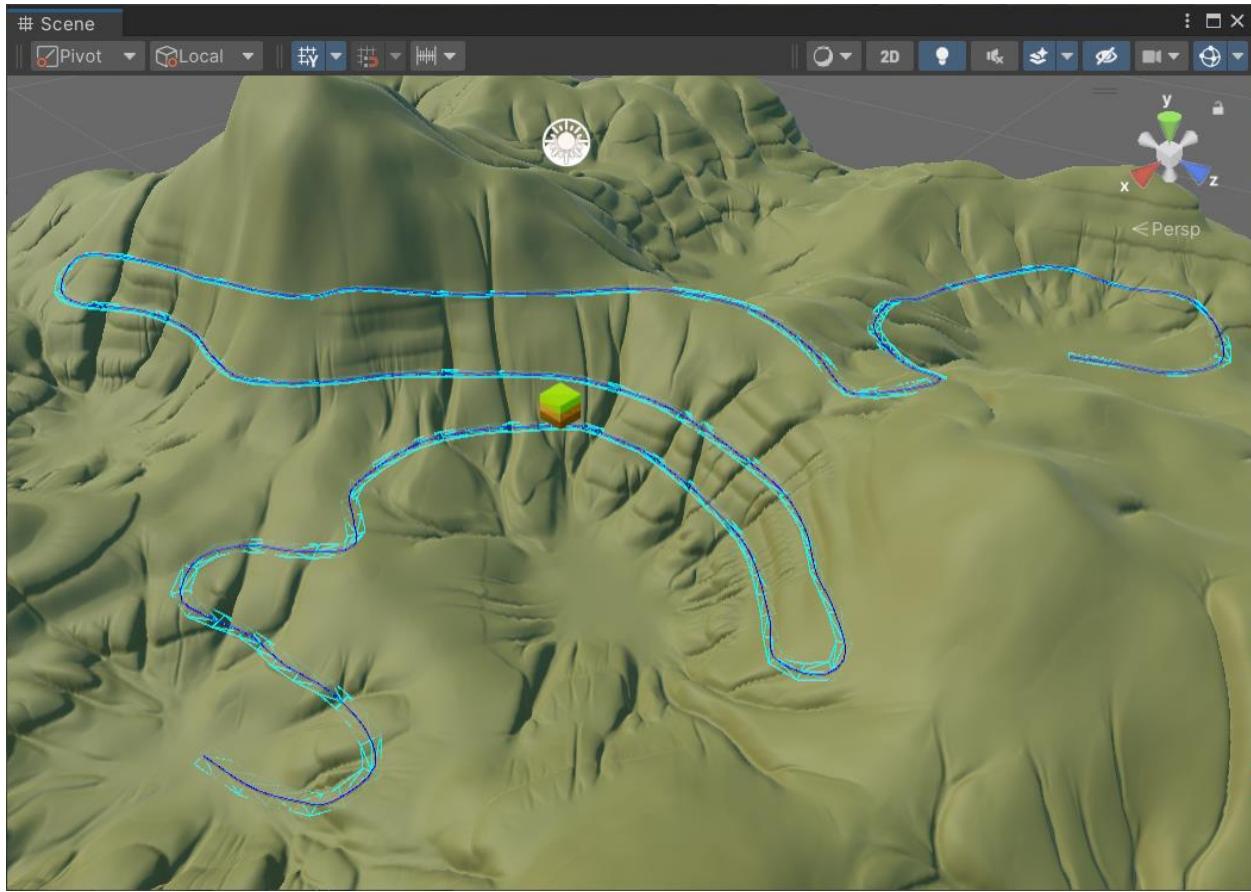
Below is the list of nodes packed in this module:

- Spline Extract: Extract the spline information such as mask, region, height map, anchors and points along the path to do something. You will only use this node in special cases as there are other nodes for common jobs.
- Ramp: Conform the terrain height to the spline mesh, useful for mountain pass, matching the terrain height to the road mesh, etc. Note that this node doesn't generate any road mesh.
- Path: Paint a path along the spline as a mask. The mask can be later used with a Weight Blend or Color Blend node for texturing.
- Thin Out Along: Remove vegetation instances along the spline, or the opposite, remove instances that are not on the spline.

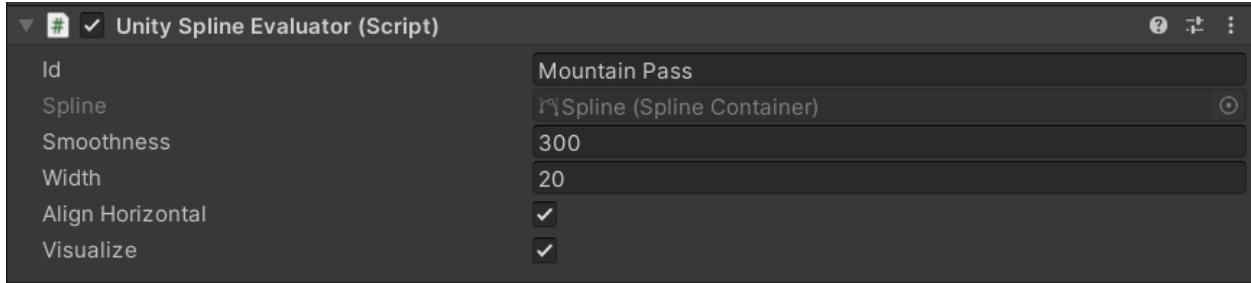
Examples

Mountain pass

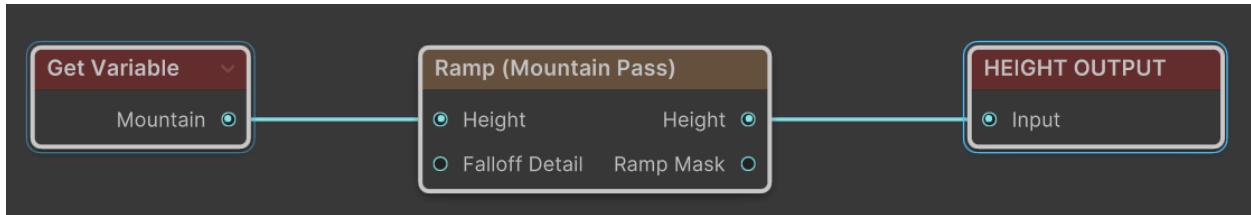
Assuming that you already have a good mountain height map. Add a spline that go around with a reasonable elevation change like this:



Then the evaluator:

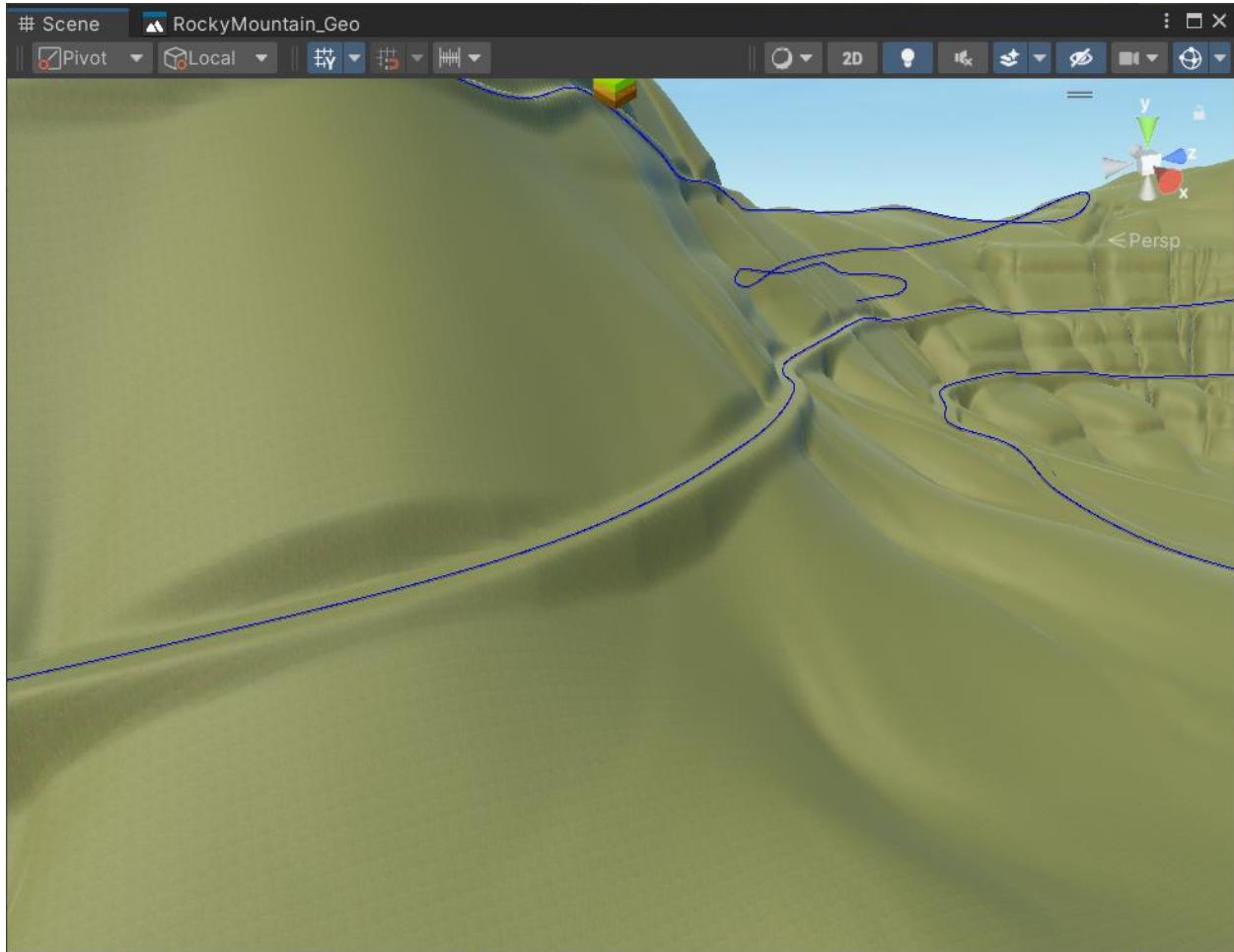


In the graph:



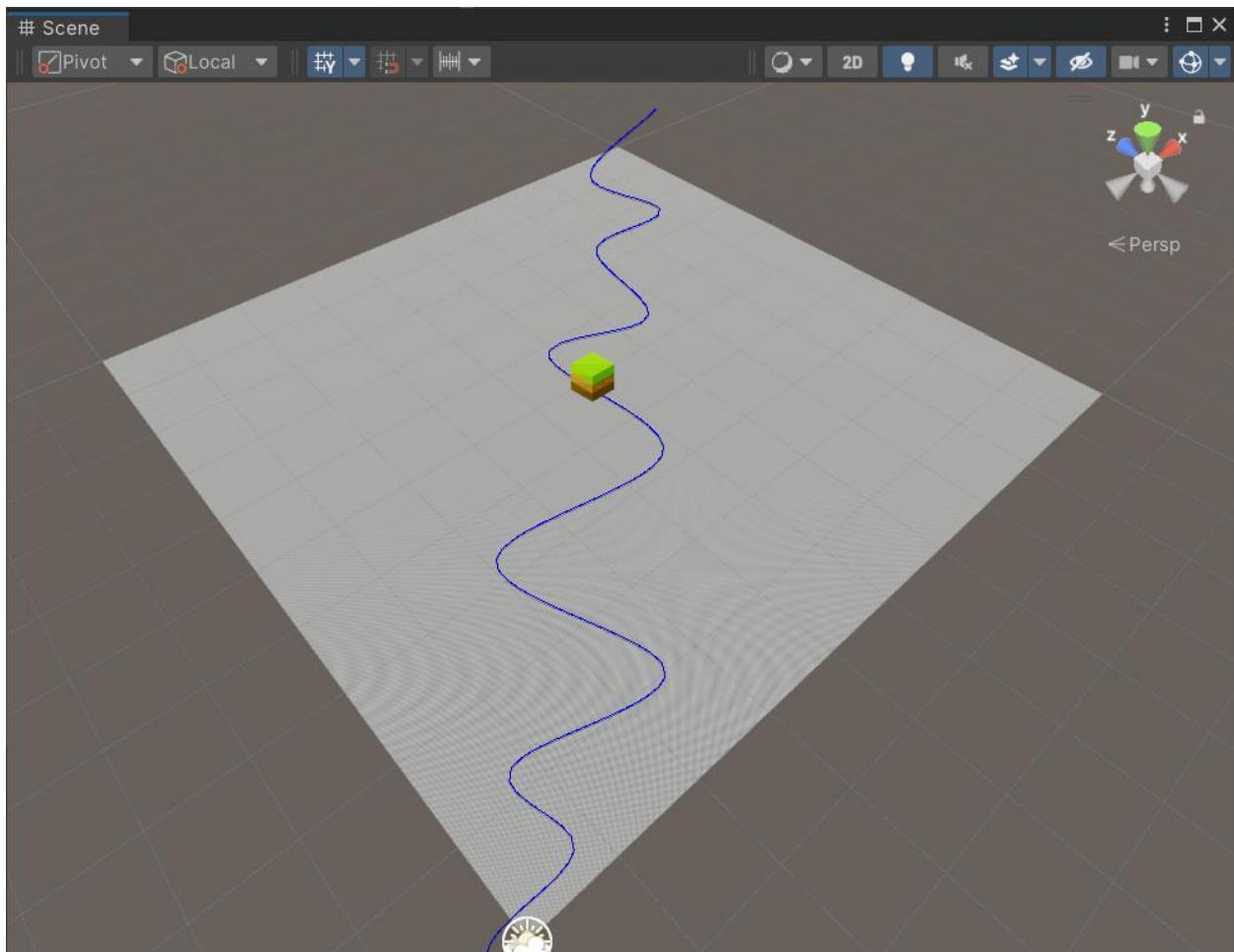
The Mountain variable is something you generate before this step, then register with the Set Variable node.

Then the result:

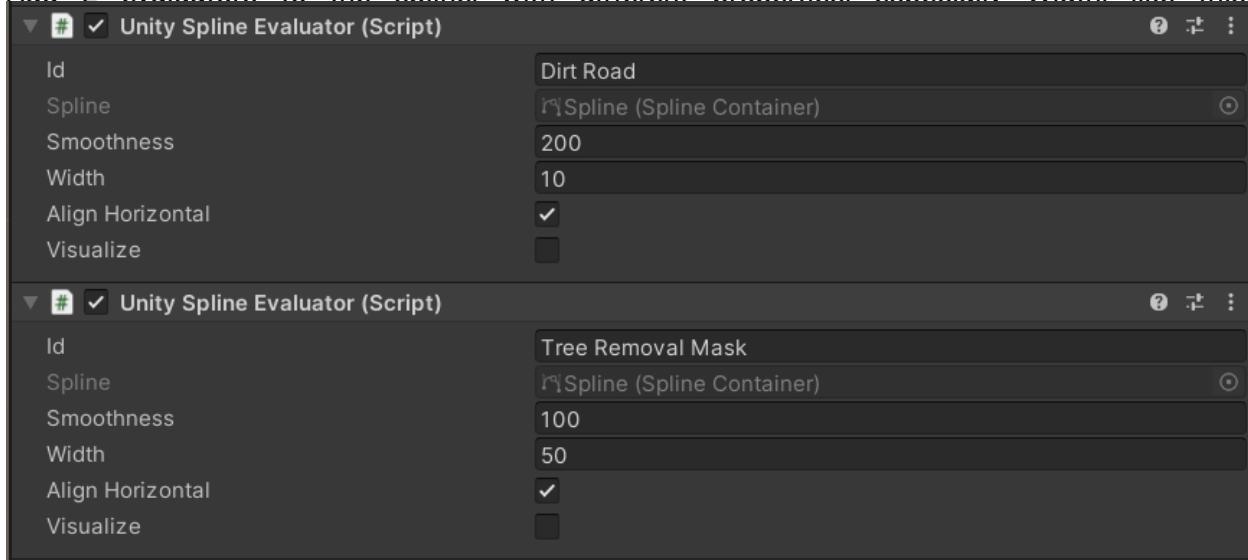


Paint a dirt road and remove trees along its path

Make a simple terrain and a spline like this:



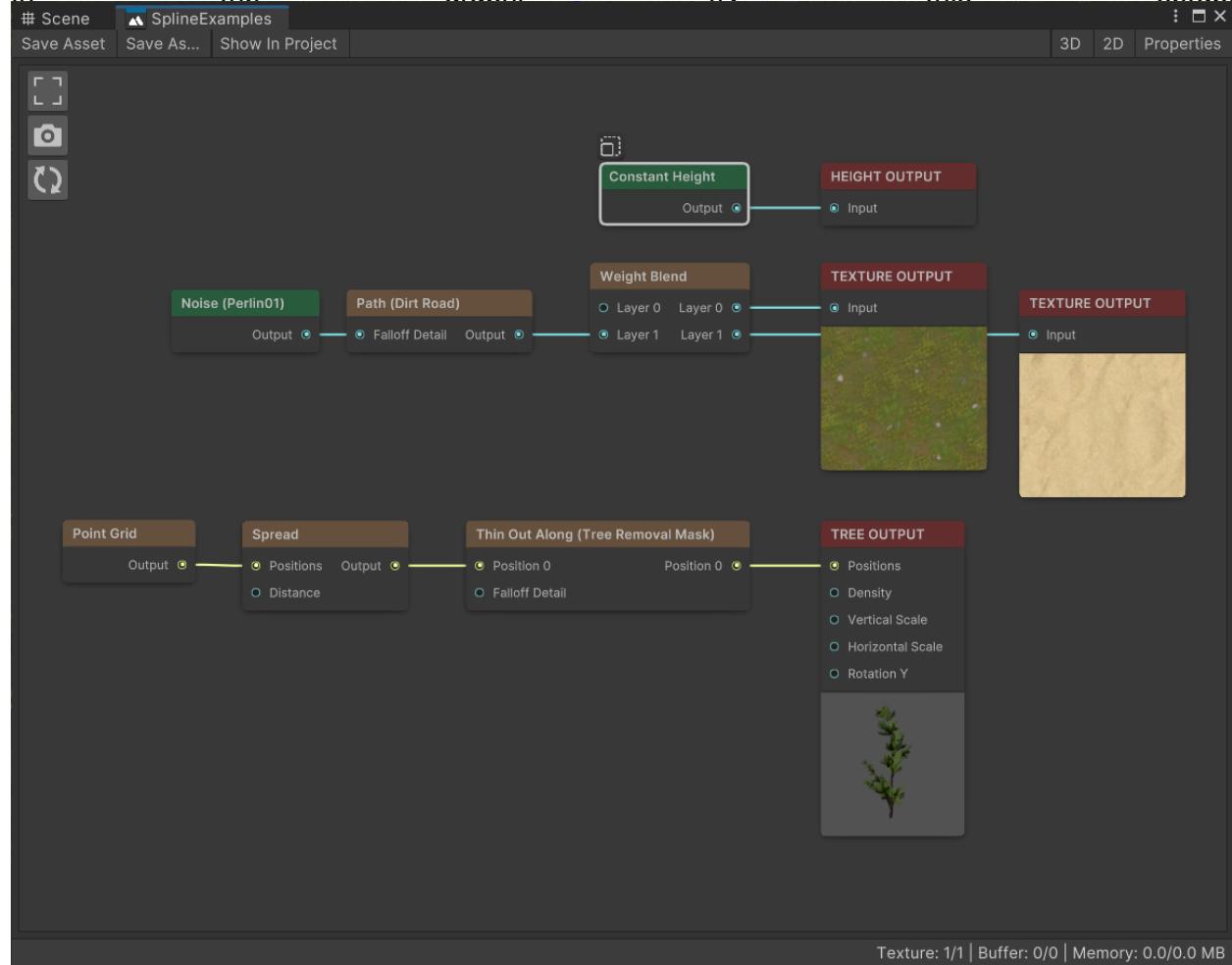
Add 2 evaluators to the spline with different properties, especially Width, like this:



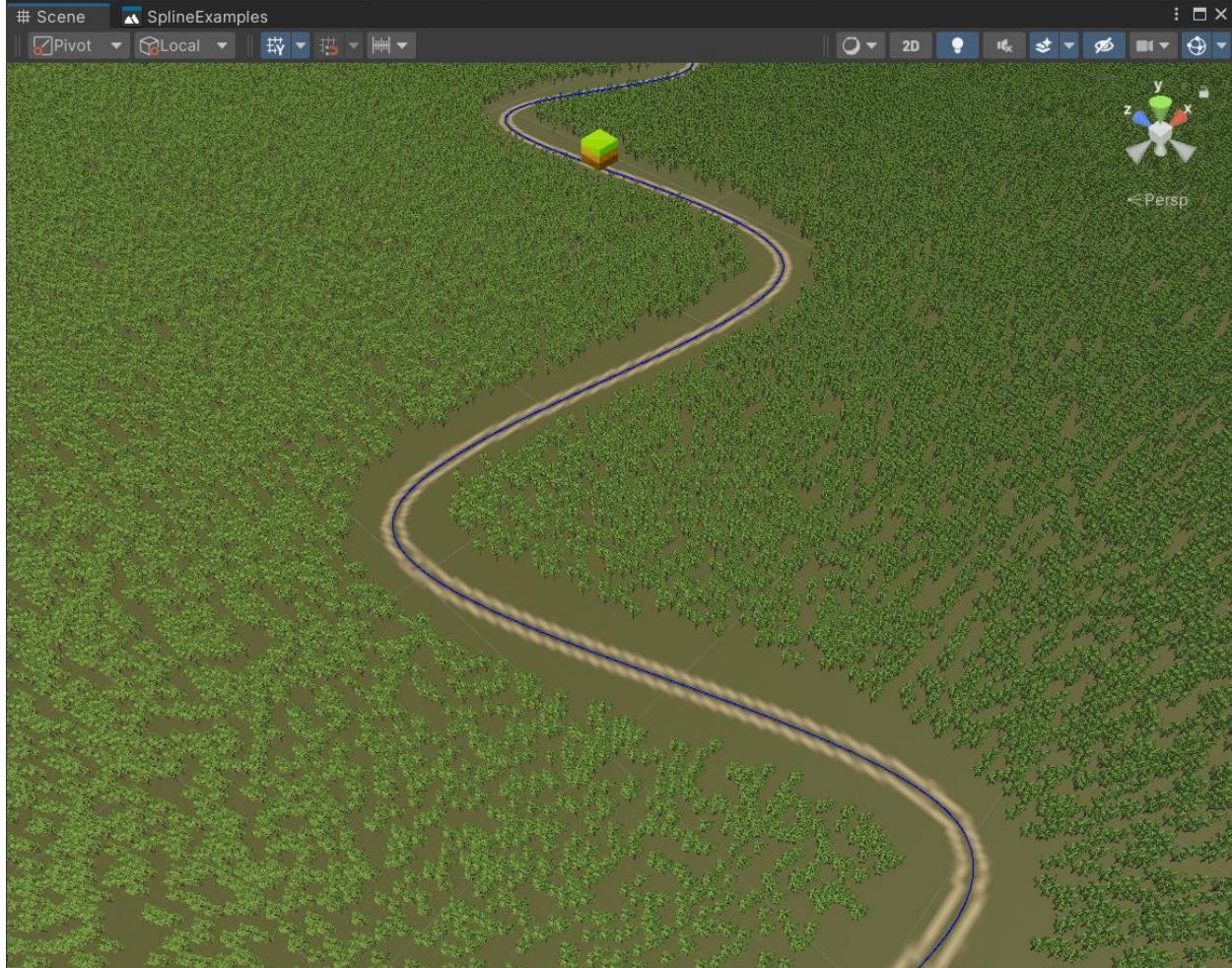
The first evaluator will be used for painting the path, while the second is for removing tree instances.

Setup a biome just to cover the terrain, and a graph for it.

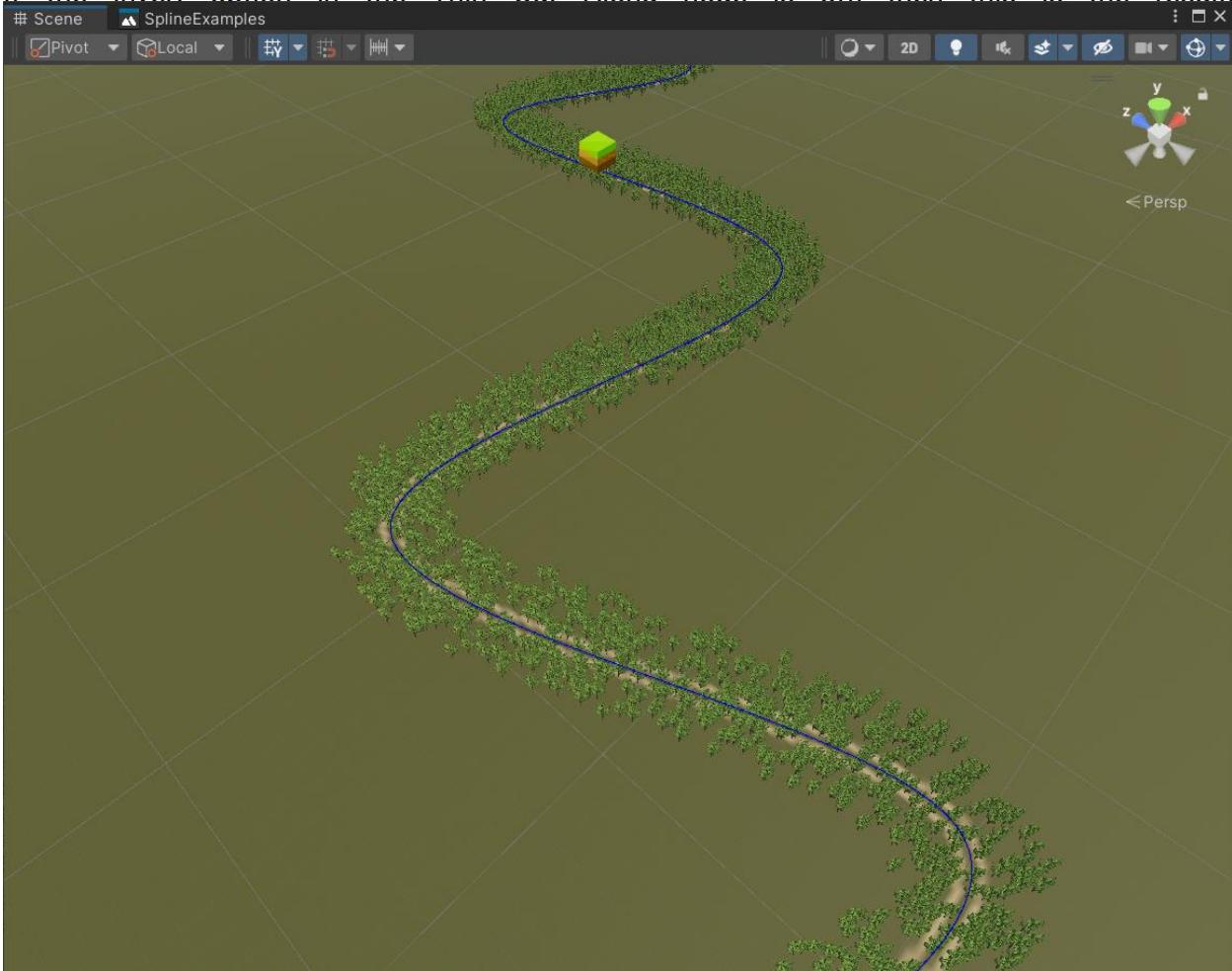
In the graph, try this setup:



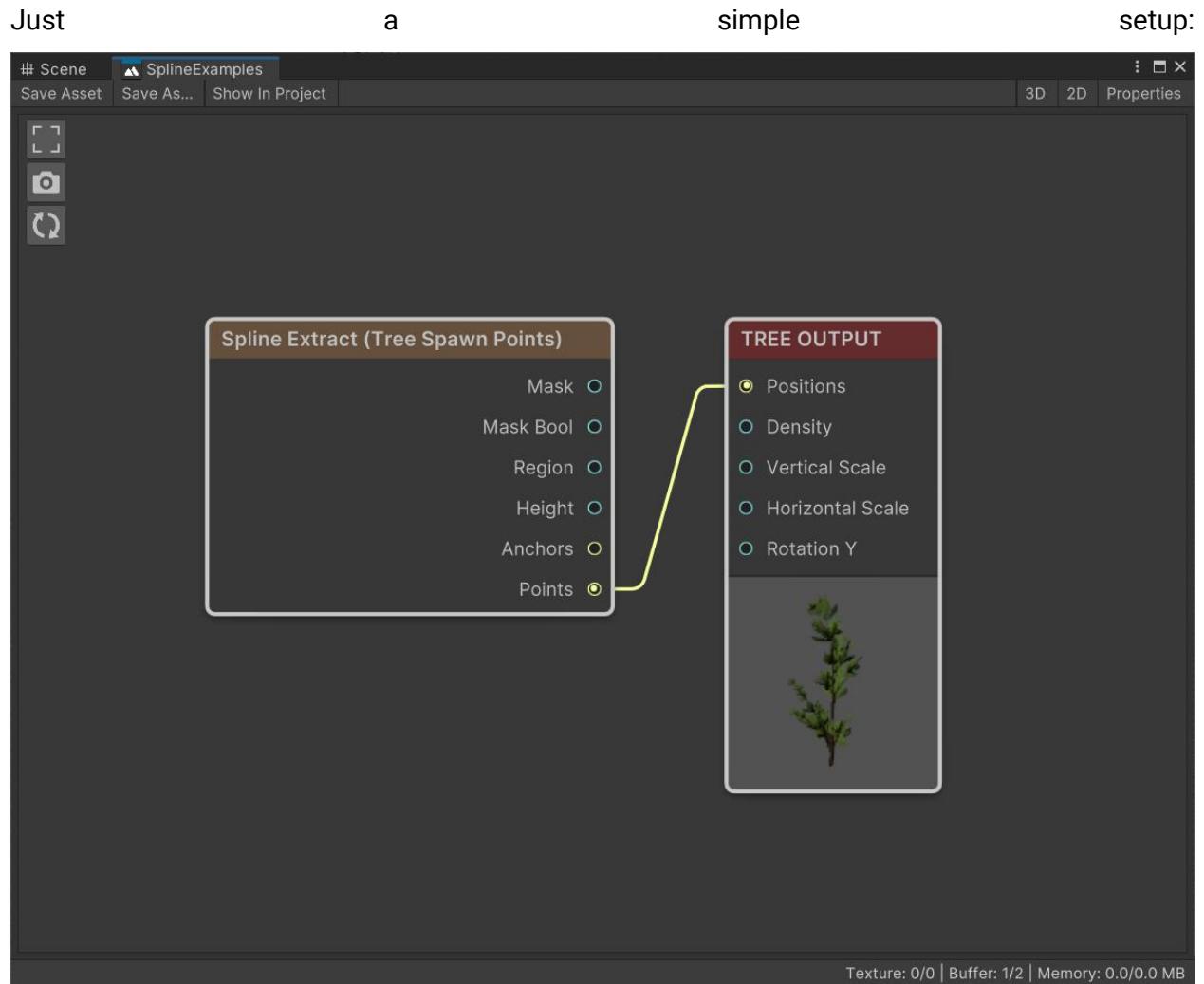
Save the graph, then this is the result (Billboard Distance was max out)



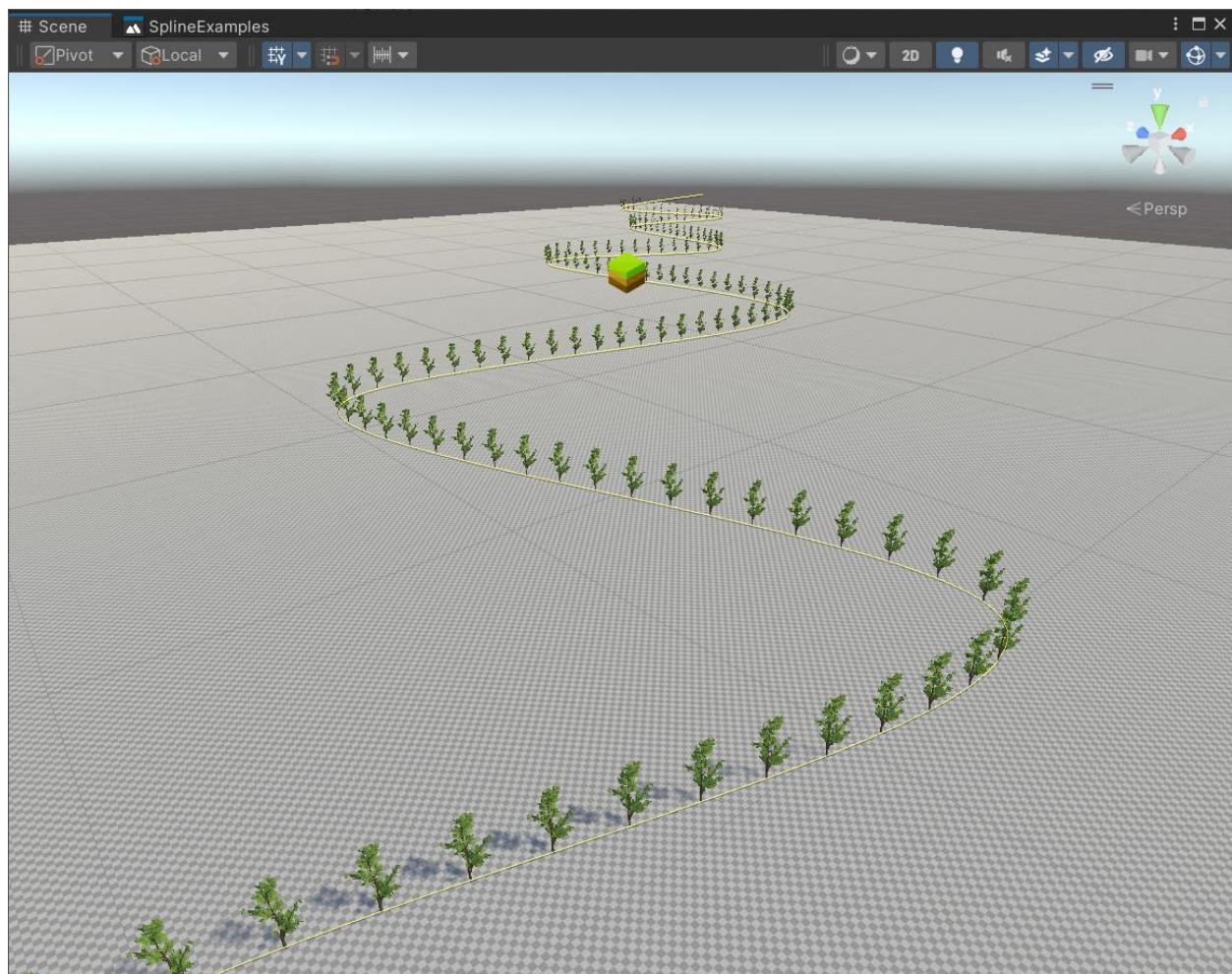
If the Invert option in the Thin Out Along node is off, then this is the result:



Spawn trees along the spline

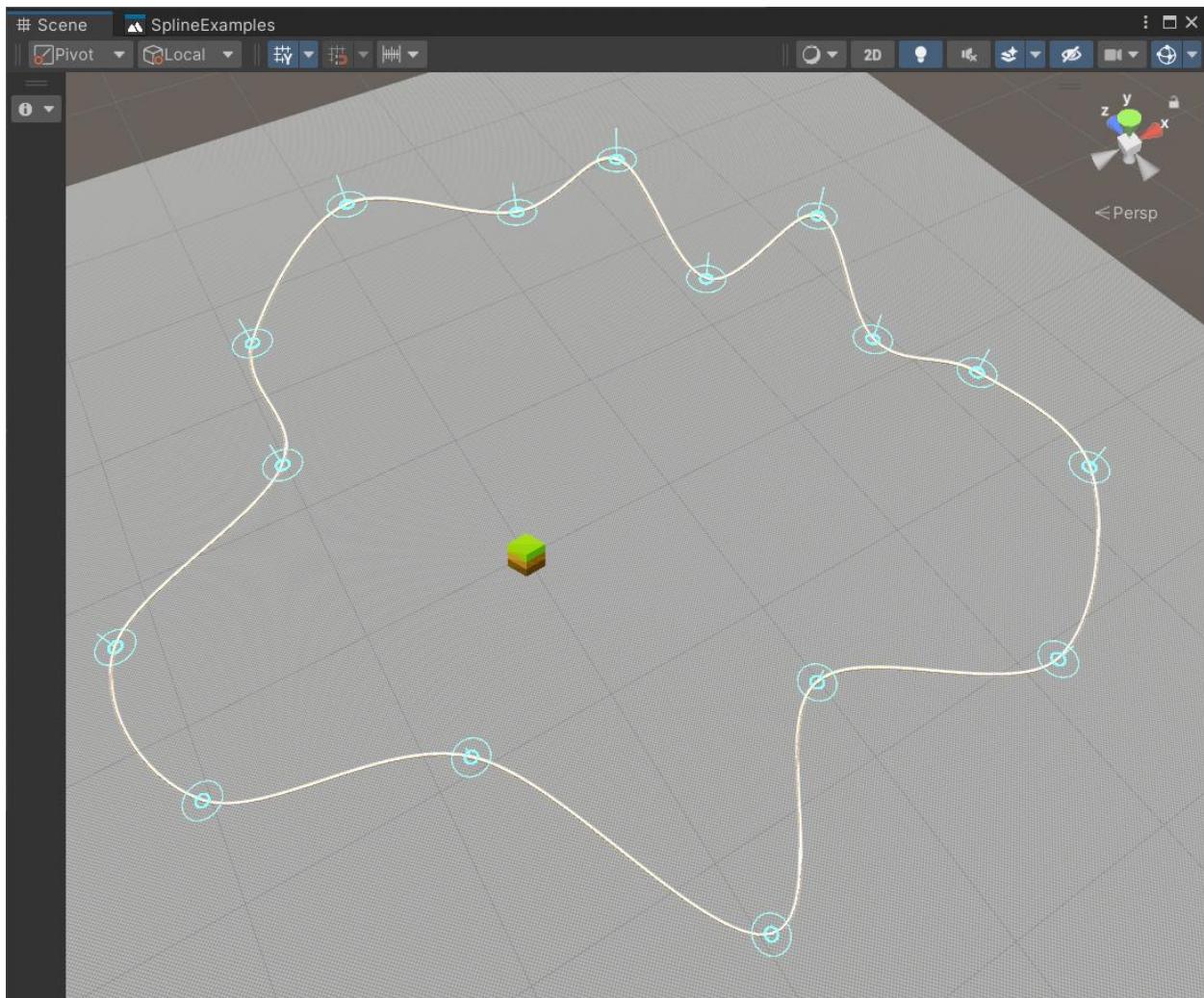


And the result:



Spawn trees inside the spline loop

Create a "Closed" spline like this:

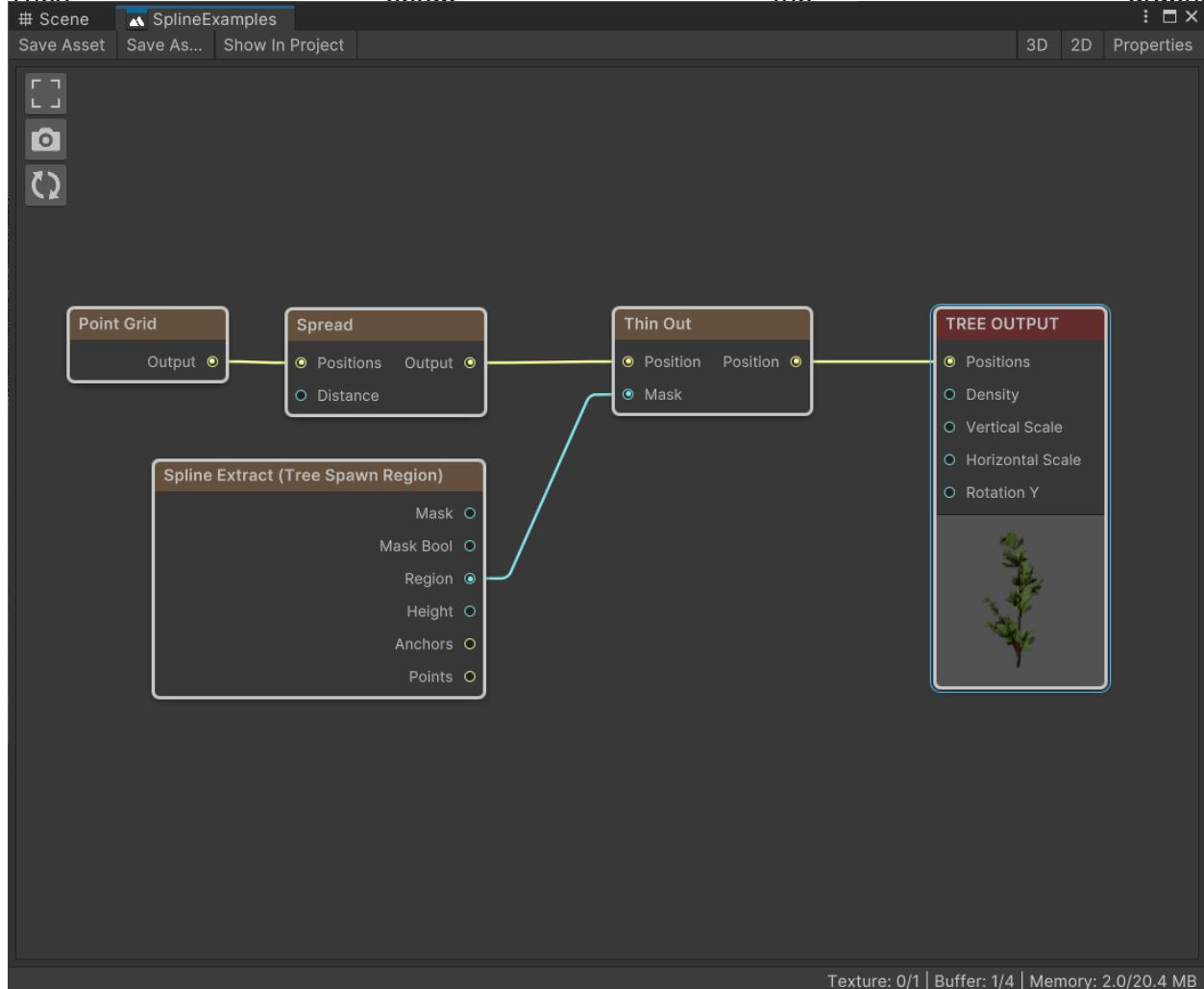


Then

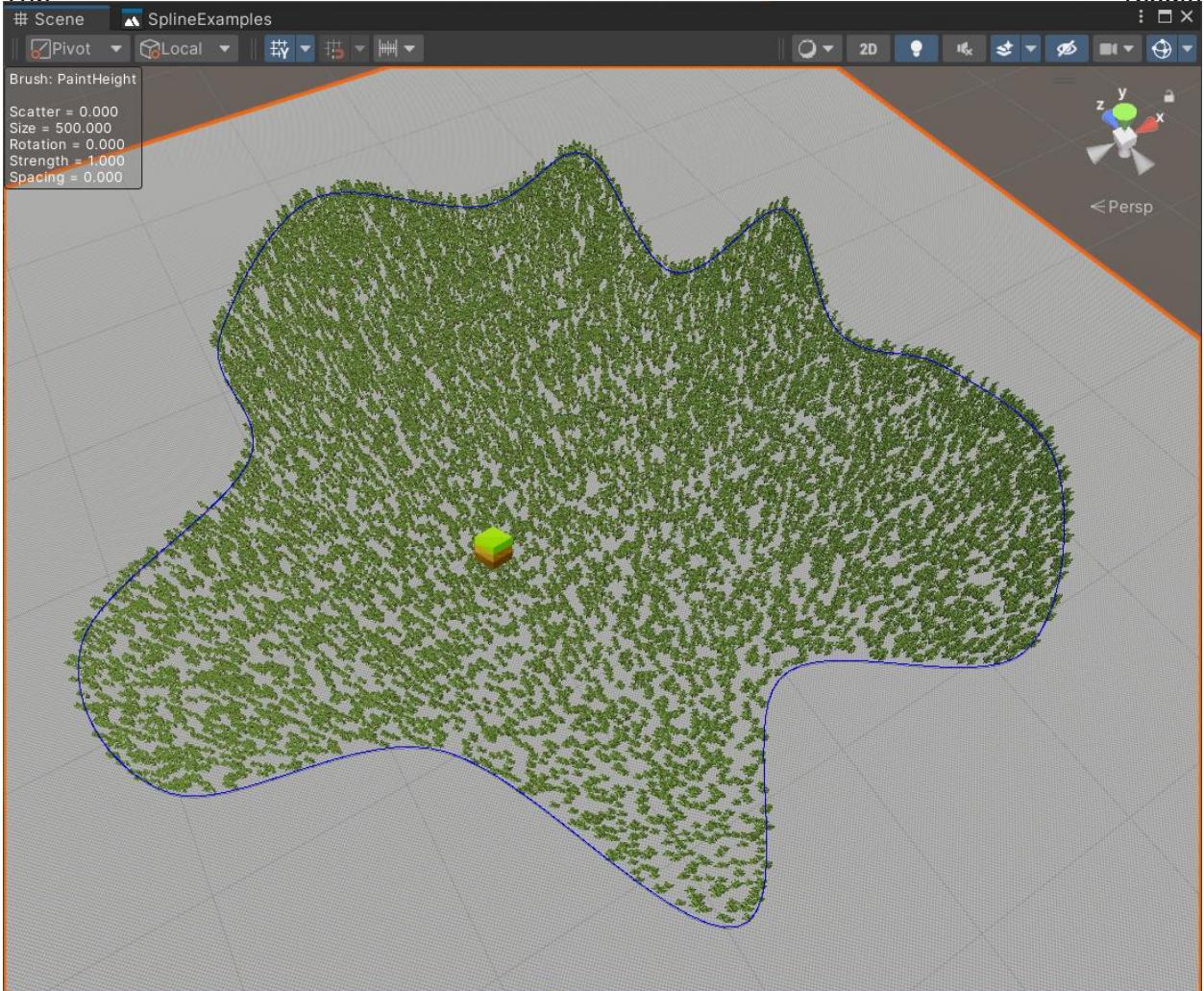
setup

the

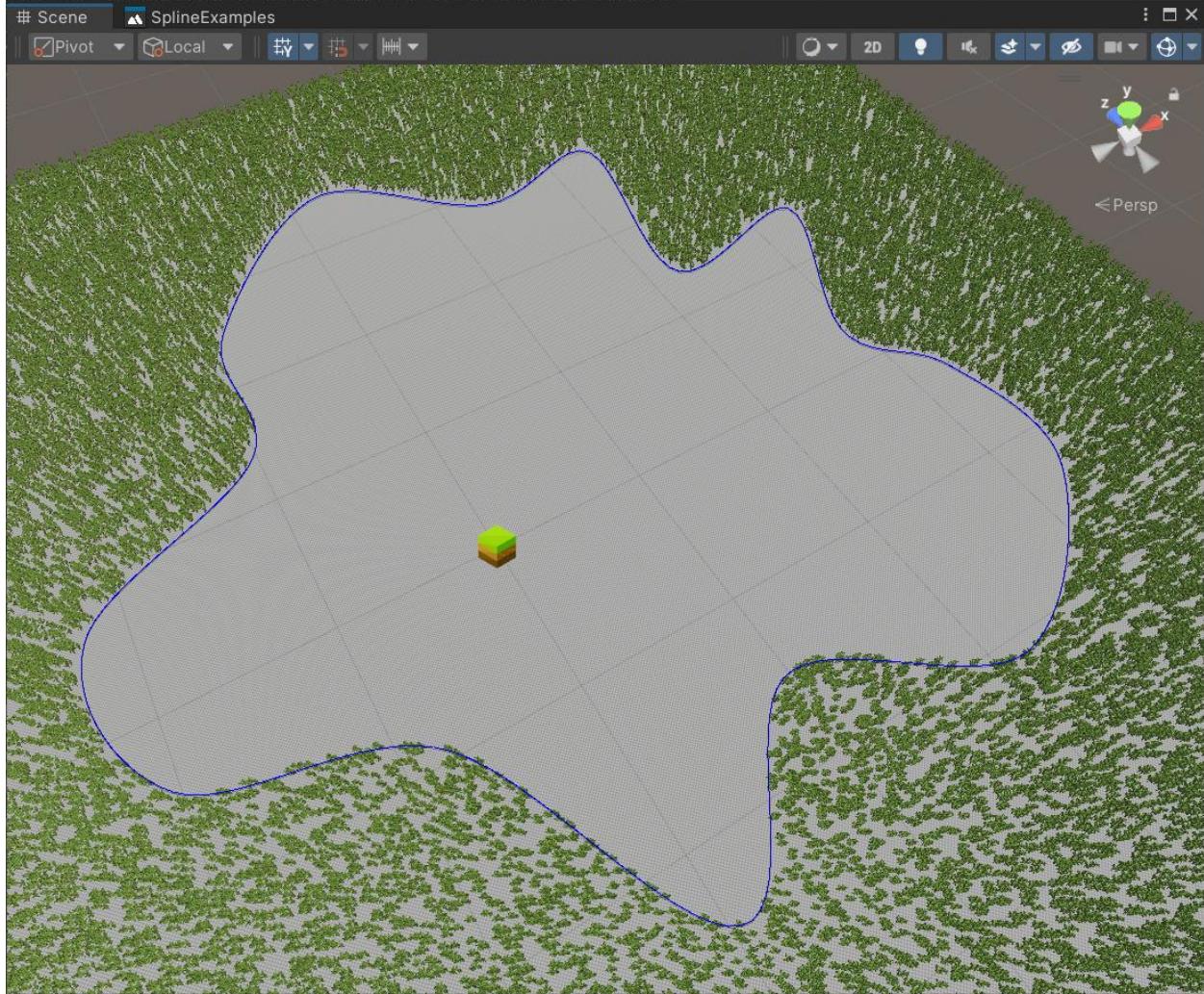
graph:



The



When the mask is inverted with Math node:



TUTORIALS ON TOPICS

Overview

This section is NOT for fresh users. It assumes you have basic knowledge of Vista workflow, such as setting up the [Vista Manager](#), [Local Procedural Biome](#), [Graph Editor](#), [Terrain Graph](#), etc.

Please be sure to take a quick read through the topics above this before moving on.

More interesting Terrace effect

Multi-level-of-detail erosion

Highlighting the erosion soil

Grass growing with Concave

Highlighting sharp edges with Convex

FAQ

Q: What are the benefits of using Vista in my project?

A: The benefits it can bring to your project is huge:

1. Quick: You can quickly create your terrain, both for prototyping and production, in just a couple of nodes. The library contains many nodes that are useful for most scenarios, including natural simulation. All run on the GPU.
2. Biome system: You can place unlimited biomes across the scene, anywhere, in any shape, those will get blended nicely. This will give you total control over one of the most important things: the game story. (*Requires [Big World module](#)*)
3. Highly extensible: You can inject custom code at some checkpoints, write custom nodes, adding support for custom terrain type (such as mesh based Polaris terrain), custom graph output, etc. The possibility is endless.
4. Modules: There are many modules provided beside the core toolset. Those modules are designed for a subset of users, not all of them, so you will buy only when you need to.
5. Affordable pricing: Because of the modular design, the money you have to spend is lower and only pay for what you need, not the whole bulky and expensive package.
6. Fast iteration: Many of the terrain tools out there are standalone apps, they require switching back and forth between 2 windows. Vista works directly inside the Unity editor, thus the iteration time is a lot faster that reduces production cost.
7. Customer service: We maintain an [email address](#) and a [Discord](#) server for customer support, pre-sale and post-sale. This way you can be confident to choose Vista.

Q: What are the key differences between Vista and other terrain generators on the Asset Store?

- Vista uses GPU for its calculation, a lot faster, but cannot be multithreaded, just frame-splitting is possible.

- Vista is NOT bound to Unity terrain (not Unity terrain exclusive), its data is generic, which can be applied to many heightfield terrain systems (currently Unity terrain AND Polaris terrain).
- Vista has many more simulations.
- Vista uses a "local" approach, which means you decide the biome position by hand -> more control, story driven, but magically infinite terrain is a problem.
- Vista is open for extension: 3rd party assets integration, custom output, code callbacks, etc. is easy to do.
- Vista can generate at runtime, but you will need some setup for the build.
- Vista uses a similar graph editor as Shader Graph.

Q: Why Vista but not other assets?

There are many terrain creation tools on the market, they all have pros and cons:

- Some are standalone apps, which will slow down the iteration between them and the Unity project.
- Some run on the CPU which is slow.
- Some only work on a single terrain.
- Some don't have biomes, or support biomes in a way that is not flexible and hard to use.

Well, Vista solves them all.

Furthermore, Vista UI was made using the same API as Shader Graph, that helps it to look, feel and behave like a Unity built-in tool, users will spend less time learning it.

Q: Grass and detail has some artifacts, missing grass at some random locations

A: Sometimes happens on Unity terrain. Mostly due to Detail Density is too high that causes instances to be stripped off. Lower the terrain Detail Density value will fix it.

Q: Trees output not working, no marker visualization in the 3D Viewport

A: The Point Grid node is mysteriously broken on some machines running DX11 without error messages. There is no fix just yet. However, you can switch to Vulkan graphic API to continue to work on the graph (Go to Edit>Project Settings>Player>Other Settings, uncheck Auto Graphic API for Windows, add Vulkan, bring Vulkan to the top of the list).