



# GAIA

By Procedural Worlds

Gaia is a system that enables the creation  
of gorgeous looking Unity terrains.

Version 1.6.1

# Contents

Installation.....	3
Quick Start .....	4
What is Gaia ?.....	4
Gaia Systems and Components .....	6
Gaia Manager .....	7
Work Flow .....	7
Session Management .....	9
Session Workflow .....	12
Session Settings.....	12
Resource Management .....	16
Gaia Defaults .....	16
Work Flow .....	17
Settings.....	18
Gaia Resources.....	21
Work Flow .....	25
Visualiser – Visualise and Edit Spawn Criteria .....	31
General Settings .....	34
Texture Prototype Settings.....	35
Detail Prototype Settings.....	40
Tree Prototype Settings.....	47
Game Object Prototype Settings .....	53
Stamping System .....	60
Work Flow .....	61
Stamper Settings .....	73
Spawning System.....	79
Work Flow .....	82
Texture Spawners .....	91
Coverage Detail Spawners.....	100
Clustered Detail Spawners.....	104
Tree Spawners.....	107
Game Object Spawners .....	110
Spawner Settings.....	117
Scanning System.....	128

Extension System (GX).....	129
Installed Extensions .....	129
Compatible Extensions.....	131
Extension Exporter .....	132
Extensions API For 3 <sup>rd</sup> Party Developers.....	134
Utilities.....	137
Support.....	139

# Installation

Installing Gaia will create the following folder structure:

## Gaia:

**3<sup>rd</sup> Party Samplers:** 3<sup>rd</sup> party sample assets; you can delete this

**Data:** Defaults and Resources data files stored here

**Documentation:** Gaia documentation

**Masks:** Sample masks used by spawners

**Materials:** Materials used for mesh previews and lighting

**Prefabs:** Spawner extension prefabs

**Scripts:** Gaia source code

**Stamps:** Gaia stamps, classified by stamp type

## Important Notes

- Gaia relies on being able to locate stamps in a fixed location, so the Gaia directory should never be moved.
- Gaia expects all terrain tiles to be square, and to share the same textures, details, trees, and resolutions between each terrain tile. If these conditions are not met, Gaia may not operate in the way expected.
- Gaia expects all stamps and image masks to be square. If these conditions are not met, Gaia will not operate in the way expected.
- Gaia can only work on scenes which have only one active terrain tile at a time. If you have a scene with many terrain tiles you can work around this by temporarily deactivating them, use Gaia, and then reactivating them when you are done. Multiple active tile support will be added in version 2.0.
- Gaia creates standard unity scenes and is subject to the usual performance constraints. Grass and trees can have huge impacts on performance, so you need to balance their density against your performance requirements.

## Removing Gaia

Gaia can be removed from your project when you have finished using it to create your scenes, but make sure you re-parent any spawned game objects first so you don't delete them when you delete Gaia!

## Quick Start

See the QuickStart guide for a quick overview on how to get started with Gaia.

## What is Gaia ?

Gaia is a system that enables rapid and precise creation of gorgeous looking terrains and scenes.

With Gaia you will:

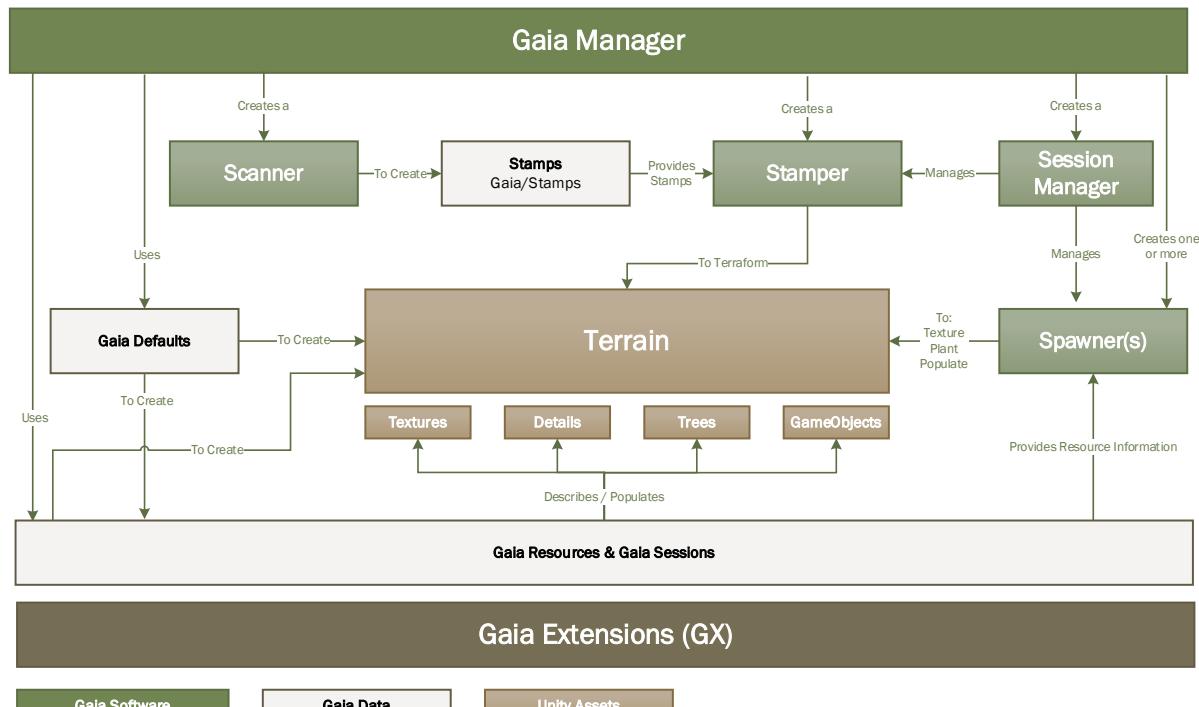
- Define and manage your resources
- Create and terraform your terrain by stamping mountains, hills, lakes and other features
- Leverage your resources to texture, plant, and populate your terrain procedurally using spawners.
- Leverage 3<sup>rd</sup> party tools and extensions to make your environments better via the Gaia eXtension system (GX).
- Backup, restore and share your environments with the Session Manager.

Gaia is different because:

- **It's standard** – There are no special shaders or other tricks that could cause compatibility problems (you can safely delete Gaia after you have created your scene);
- **It's stamp based** – Terrains are created using stamps, which is a visual, intuitive and precise way of getting exactly the feature you want at exactly the location you want;
- **It's extensible** – The Gaia eXtension system allows you to easily leverage 3<sup>rd</sup> party assets. Additionally, the Spawner Rule Extension system allows you to customise how Gaia spawns, and the Scanning system allows you to create more stamps based on terrains, windows 16 bit raw files, meshes, or even images and textures from the internet;

- **It's replicable, adaptable and rule based** – All aspects of terrain texturing, planting and object placement is done procedurally and is seed based, which allows it to adapt to new terrains, and makes it perfect for networked games – the session management system records every step you take and allows you to play it back at will;
- **It's malleable** – The spawning system can be extended via code allowing you to enhance or over-ride its behaviour to suit your own game and own work flow.

# Gaia Systems and Components



Gaia is made up of the following main systems:

- **Gaia Manager** – manages your workflow;
- **Session system** – manages session recording and playback;
- **Resources system** – manages your textures, details, trees and game objects;
- **Scanning system** – creates stamps from terrains, 16 bit windows raw files, 3D meshes and 2D textures;
- **Stamping system** – uses stamps to terraform the terrain;
- **Spawning system** – procedurally textures, plants and places resources;
- **Screenshot system** – allows you to take screenshots of your environments;
- **Camera FX and lighting** – allows you to apply unity camera fx and lighting;
- **Extension system** – create and use extensions that allow you to integrate with and leverage 3<sup>rd</sup> party assets;

- **Utilities** – handy utilities to save time.
  - Show Scanner – Shows the scanner component
  - Show Visualiser – Shows the visualiser component
  - Show Terrain Height Adjuster – Shows the terrain height adjuster
  - Show Texture Splatmap Mask Exporter – allows you to export texture splatmap masks
  - Show Grass Splatmap Mask Exporter – allows you to export grass splatmap masks
  - Show Waterflow Mask Exporter – allows you to calculate and export water flow masks
  - Show Terrain Normal Exporter – allows you to export the terrain normals
  - Show Terrain OBJ Exporter – allows you to export the terrain as an OBJ mesh file
  - Terrain Shoreline Exporter – allows you to export the terrain shoreline to use as a mask.
  - Terrain Heightmap Exporter – allows you export terrain height map as a PNG file
  - Show Extension Exporter – allows you to create and export Gaia extensions

## Gaia Manager

The Gaia Manager drives your Gaia experience by constructing and configuring the components you use.

As you step through the manager work flow, the manager will first create a session for you – that will record every step you take (to allow you to share it or pay it back later), and then construct new components for you.

The session based approach ensures that the Gaia components are correctly configured and that a consistent sea level, default and resources set is used across your scene. You can tailor these over time as you become more familiar with the system.

Objects created by the Gaia Manager are added as children to the Gaia object or the Gaia Environment in the scene.

## Work Flow

1. Open the Gaia Window by selecting Window -> Gaia -> Show Gaia Manager (or by pressing Ctrl G).



2. Select the defaults file that you want to use, or new to create a new Defaults file.
3. Select the resources file that you want to use, or click New to create a new Defaults file.
4. Select the Operation mode you wish to operate in.

**Standard** – Set up an entire scene with a few clicks;

**Advanced** – Set up a scene, with detailed access to specific components;

**Utilities** – Quick access to common utilities.

**GX** – Quick access to Gaia extensions.

5. Perform the operations you want by clicking the relevant buttons.

#### Note:

The manager requires you to select your Gaia Defaults and Gaia Resources files, or to create new ones.

The Default's file describes the defaults that will be used to create new Terrains.

The Resources file describes the resources such as Textures, Details, Trees and Game Objects that Gaia will use, and the criteria by which they can be spawned into the scene.

Defaults and Resources files are scriptable objects and scene independent so if you change them in one scene you will affect all scenes that use them. You can use as many resources files as you like (even multiple references to the same physical resource), and may choose to have a different set of resources and spawners for each different area.

If you wish to use different settings then select the file and hit Ctrl – D to duplicate it, and make your changes. The currently selected defaults and resources files will be passed into any new components you create.

These files are described in the Resource Management section.

## Session Management

All terrain creation, stamping and spawning work within Gaia is recorded in a session by the session manager.

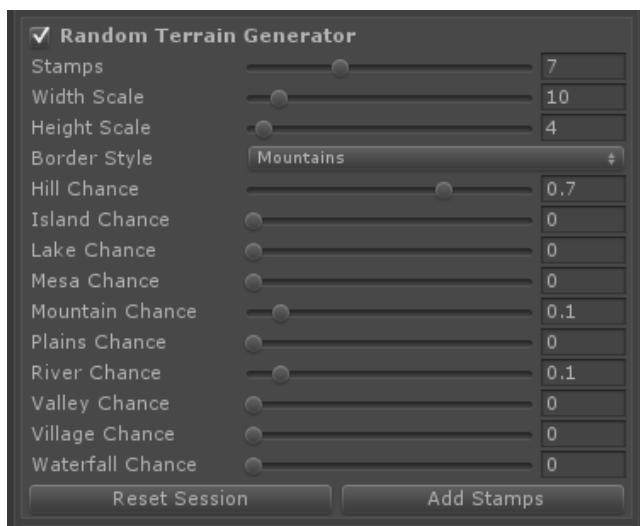
The session manager manages sea levels, and based on them records operations, enables, disables, deletes, applies and plays them individually into a scene, or as a group, and can also generate entirely new scenes.



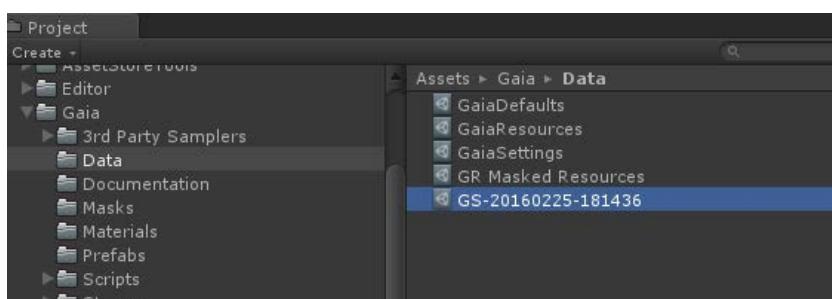
The purpose of the session is to record what you did, operation by operation, so that you can replay it later, undo bad operations, back it up, or perhaps share it with someone else.

If you share it with someone else, and they have the same assets in their project then they will create an exact replica of your scene when they play it back. Because session files are small (about 200k without a preview image, and 400k with one), this is a very convenient way to share and collaborate on ideas and scenes.

The session manager can also randomly generate new terrains with the Random Terrain Generator. You tell the generator what type of features you would like in your scene, and the relative proportion that feature type will be used relative to other feature types, and it will dynamically generate new scenes.



In the example above the session manager will add 7 new stamps, with an approximate width of 10, and an approximate height of 4, and surround the scene with mountains. The width and scale are the same as you would use in a stamper, so choose values that make sense with your scene.



Technically, the session is a scriptable object starting with GS-<filename>. The naming standard for session files is GS-YYYYMMDD-HHMMSS, and is stored by default in the Gaia / Data directory.

While you can browse your sessions manually, the best way to browse for them is via the Session Manager as it unpacks and interprets the embedded session data.



When the session is played back, the defaults and resources files that were embedded in the session file when it was created are exported into a session directory underneath `GaiaSessions`, and given a unique name based on the session name and the original defaults and resource file names. This ensures that the defaults and resource files are unique.

In addition to this, because the referenced asset files can reside in different directories in the target environment than they were in the source environment, the assets referenced in the resourced files are re-associated with their matching physical assets. This is done by file name, so if you have multiple duplicates of the same file name then you will need to re-associate them with the correct original in your target project. This is a one-time operation per project.

When a session is UNLOCKED you can record new operations into it, and also modify its contents, however if a session is LOCKED you can't. It is always good practice to lock a session when you have finished with it so that you do not inadvertently change it.

## Session Workflow

You can either create a new session, edit an existing one or replay an old one.

The Session Manager is automatically created whenever you use Gaia to create a terrain, stamp a terrain or spawn a terrain, or can be created manually via the Advanced menu.

### New Session Workflow:

To create a new session you can do one of:

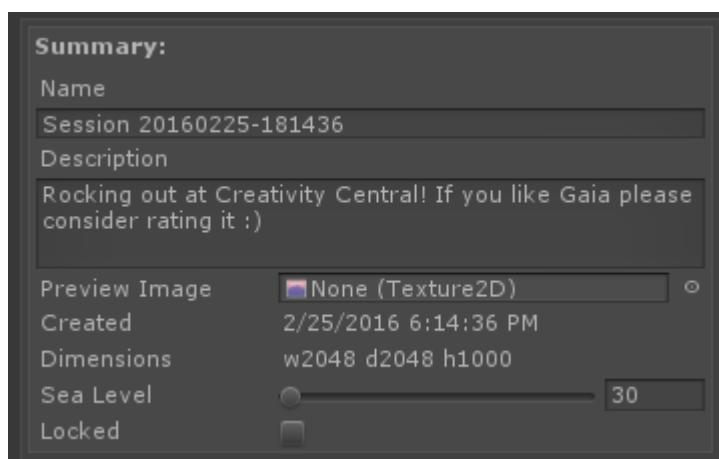
- In Standard Menu, Create Terrain and Show Stamper
- In Standard Menu, Create Spawners
- In Advanced Menu, Show Session Manager
- In Advanced Menu, Create Terrain or Show Stamper
- In Advanced Menu, Create any Spawner

Then every operation that you perform with Gaia will be recorded into the session, displayed in the Operations section of the Session Manager, and will be available for later playback.

Note: You can only add or change operations in a session file if it is unlocked.

## Session Settings

### Session Summary



The session summary allows you to name and describe your session, add a preview image, change the sea level, lock or unlock the session, and shows some session metrics.

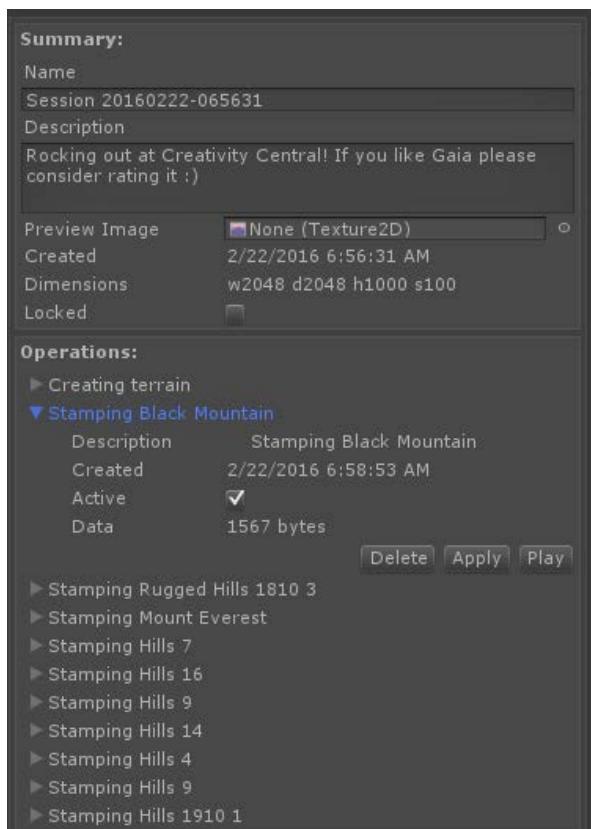
When a session is initially created the sea level is taken from the currently active or last used defaults file. You can easily change the sea level here any new stamps or spawns will adhere to that sea level.

Unlike the previous version of Gaia, the sea level in a session will be applied to every stamp and spawn as they are then applied into the scene. A nice outcome of this is that you can change the sea level as you like and then replay the session.

When a session is locked, no further changes can be made to it. It is a good idea to lock sessions when you are done in order to prevent accidental modification.

When you add an image to a session it will be converted to a smaller image and then embedded into the session. This adds about 200kb to the size of the session but is a great way to visually remember what the session was about.

## Session Operations



Each operation performed by Gaia is added to the session and shown in the Operations section. Operations are only added if the session is unlocked.

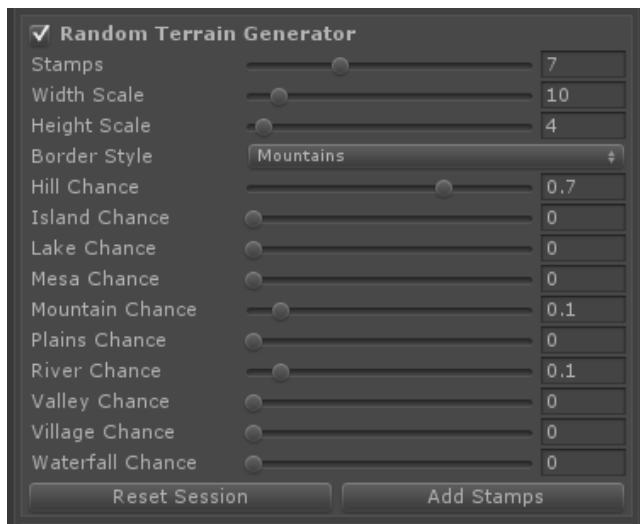
You can mark an operation as inactive – which will cause that operation to be skipped when the session is played back.

You can Delete an operation.

You can Apply the operation. Applying an operation takes the operations configuration and applies it to the relevant stamper or spawner. You can then look at the stamper or spawner to see how it has been set up. This is a great way to learn new techniques when receiving sessions from other people.

You can Play the operation. Playing an operation will Apply the operation to the relevant stamper or spawner and then Stamp or Spawn it.

## Random Terrain Generator Workflow:



To create a new random terrain, or enhance the existing terrain with additional random stamps click on the Random Terrain Generator tab in an unlocked session.

Choose the number of stamps, the approximate stamp width and height scale (you can base this on the typical scale that you already use to place your stamps), and the relative frequency of the terrain types you would like to apply to the terrain.

Then hit Add Stamps to add the relevant stamps to the terrain. The session manager will randomly add and position stamps in the relative proportions of the stamp chances selected. When you are done with this you can hit Play Session to play the stamps back.

If you are unhappy with the result then use the Terrain Helper / Flatten terrain button to flatten the terrain, the Reset button to remove the old stamps, and then the Add Stamps button to generate a new set of stamps. You can then hit the Play Session button play it back.

You can hit Add Stamps as many times as you like to add more stamps to the session.

## Terrain Helper

The terrain helper option allows you to modify your terrain independently of the session – so the things you do there will not be recorded into the session.

This allows you to experiment with adding, inactivating and deleting steps in your session and then playing your session to see the modified outcome.

## **Session Controller**

The session controller session allows you to play your session as a whole, and also to export the assets that were embedded within the session.

By default any assets that were embedded within the session will be exported during playback so that Gaia can access and use them. You can find them in GaiaSessions / Session Name.

Note: This is very useful because it offers you the opportunity to switch resources in and out of the relevant resources files and then to see how they affect the resulting scenes.

# Resource Management

Gaia can adapt to any terrain and the rules that derive this are encapsulated in the Gaia Defaults and Gaia Resources files, as well as in the Spawners.

## Gaia Defaults

*Gaia Defaults* are used by Gaia Manager to create new Gaia Resources files and new Terrains.

The intention is have one place in which you configure general settings and they include things like sea level, beach height, terrain size and height, detail, tree, wind and resolution.

You can create multiple defaults files, one for each different type and style of terrain you want to create. To use them select the defaults file in Gaia Manager and then follow normal manager workflow.

The majority of these settings are passed straight through to the Terrain object, and for an introduction to how these settings affect your terrain have a look at: [http://wiki.unity3d.com/index.php/Terrain\\_tutorial](http://wiki.unity3d.com/index.php/Terrain_tutorial). They can have a significant impact on the performance of your project, so treat them with care.

Defaults files are stored in the Gaia \ Data directory and are independent of any scene.



Location of Gaia Defaults files.

## Work Flow

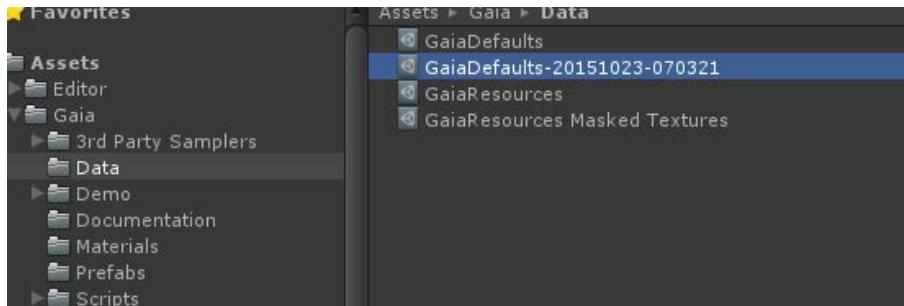
### Create a New Defaults File



Open the Gaia Manager.

Then click the New button next to Defaults.

This will create a new Defaults file in the Gaia \ Data directory and select it in Gaia Manager.



The naming standard is GD-YYYYMMDD-HHMMSS.

### Duplicate a Defaults File

Select the defaults file you want to duplicate.

Hit Ctrl-D to duplicate it. Or alternatively select the Edit -> Duplicate menu.

## Settings

### General Settings

#### Sea Level

The default sea level in meters of any derived Gaia Resources file. This value is passed into the session and then into the spawners and they all work off this one value.

#### Beach Height

The default beach height in meters of any derived Gaia Resources file.

#### Base Terrain

Basic terrain settings. Used by the Gaia Manager when creating new terrains.

##### Pixel Error

The accuracy of the mapping between the terrain maps (heightmap, textures, etc) and the generated terrain; higher values indicate lower accuracy but lower rendering overhead.

##### Base Map Distance

The maximum distance at which terrain textures will be displayed at full resolution. Beyond this distance, a lower resolution composite image will be used for efficiency.

##### Cast Shadows

Whether or not the terrain casts shadows.

##### Material

The material used to render the terrain. This should use a suitable shader, for example Nature/Terrain/Diffuse. The default terrain shader is used if no material is supplied.

##### Physics Material

The Physic Material used for the terrain surface to specify its friction and bounce.

### Tree & Detail Objects

Basic terrain settings controlling tree and detail objects. These settings may have different effect on Unity trees, and SpeedTree trees. The settings are merely passed through by Gaia.

#### Draw

Draw trees, grass & details.

#### Detail Distance

The distance (from camera) beyond which details will be culled.

#### Detail Density

**The number of detail/grass objects in a given unit of area. The value can be set lower to reduce rendering overhead.**

#### *Tree Distance*

**The distance (from camera) beyond which trees will be culled.**

#### *Billboard Start*

**The distance (from camera) at which 3D tree objects will be replaced by billboard images.**

#### *Fade Length*

**Distance over which trees will transition between 3D objects and billboards.**

#### *Max Mesh Trees*

**The maximum number of visible trees that will be represented as solid 3D meshes. Beyond this limit, trees will be replaced with billboards.**

### **Wind Settings**

Settings which control how wind affects details (grass) on a terrain.

#### *Speed*

**The speed of the wind as it blows grass.**

#### *Size*

**The size of the “ripples” on grassy areas as the wind blows over them.**

#### *Bending*

**The degree to which grass objects are bent over by the wind.**

#### *Grass Tint*

**Overall tint applied to grass objects.**

## Resolution Settings

Settings which control the resolution of the underling terrain components. These will be overridden by Gaia to ensure they are optimal settings for the terrain. Note: These settings are per tile settings.

### Terrain Size

The size of terrain tile in X & Z axis (in world units). Gaia limits terrain tiles to being square. If you need a rectangular terrain then position two square terrains side by side.

### Terrain Height

The height of the terrain in world unit meters.

### Heightmap Resolution

Pixel resolution of the terrain's heightmap (should be a power of two plus one e.g.  $513 = 512 + 1$ ). Higher resolutions allow for more detailed terrain features, at the cost of poorer performance.

### Detail Resolution

Resolution of the map that determines the separate patches of details/grass. Higher resolution gives smaller and more detailed patches.

### Detail Resolution Per Patch

Length/width of the square of patches rendered with a single draw call.

### Control Texture Resolution

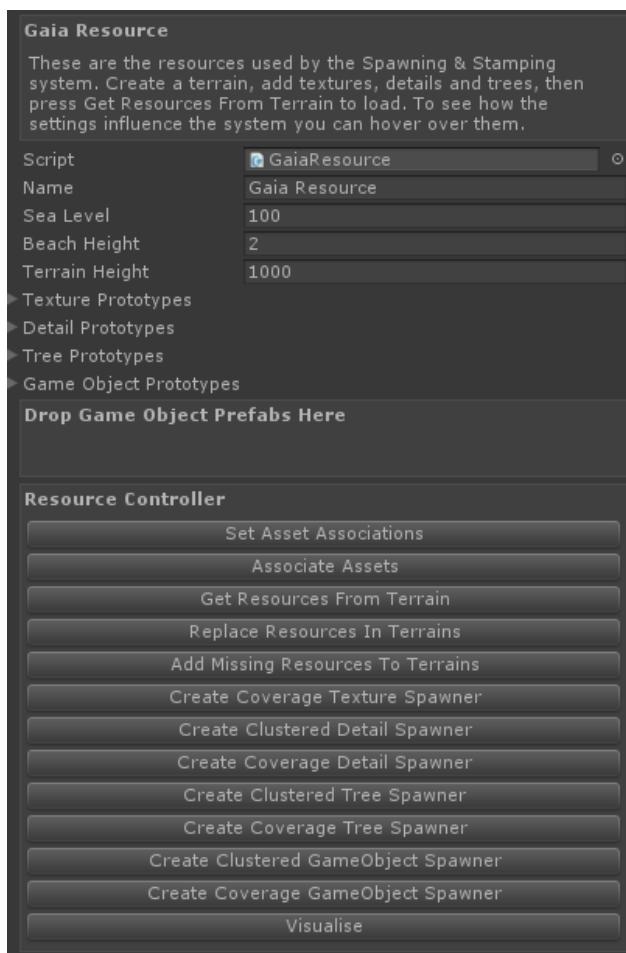
Resolution of the “splatmap” that controls the blending of the different terrain textures. Higher resolutions consume more memory, but provide more accurate texturing.

### Base Map Size

Resolution of the composite texture used on the terrain when viewed from a distance greater than the base map distance (see above).

## Gaia Resources

Gaia Resources are the assets used to configure textures, details and trees in new Terrains, and the settings used by the Spawning system that guide where and how those assets can be applied.



The four types of resources managed by Gaia are:

- Texture resources. Used to texture the terrain;
- Detail resources. Used to plant terrain details such as grass;
- Tree resources. Used to plant terrain trees;
- Game Object resources. Used to populate the terrain.

The process of using a resource in your scene is called Spawning and is performed by a Spawner. The spawner chooses a location and asks the resource to assess that location and assign a fitness value to it. A fitness value is a number in the range between zero, which is a terrible fit, and one which is a perfect fit. The fitness value is calculated based on the content of the resources DNA and Spawn Criteria. It also influences the strength and size of the resultant Spawn.

The structure of the resources scriptable object consists of general physical settings, and then a section for each resource type. Each resource type consists of a description of the physical attributes of the resource, then the DNA of the resource (where applicable), then the Spawn Criteria for the resource, and finally the Spawner Extensions.

The physical attributes are used to physically identify and configure these resources within the scene. The DNA is used to control the physical dimensions and dimension ranges, as well as how closely the resource can be placed to other resources. The Spawn Criteria is used to assess the fitness of the underlying terrain.

Resource files are stored in the Gaia \ Data directory and are independent of any scene.



Location of Gaia Resources files.

#### Spawn Criteria & Fitness:

Let's imagine how a coastal tree might evaluate a terrain?

First let's consider height. Undersea locations are bad, and would have a fitness of zero, locations in the range of say 5 – 30 meters are good, and have fitness of one, and fitness would gradually decline to zero as you went higher.

Then let's consider slope. The tree likes flat ground but could not grow on cliffs, so flat ground would be assigned a fitness of one and then subside to zero the steeper the slope becomes.

Then let's consider Proximity. We might say that this tree can only grow near game objects tagged as 'Tree Magnets'. Not a great tag name but you get the idea.

Finally let's consider texture. We have a city that is situated near the ocean, and we know we are near the city when we detect a cobble stone texture. We could select only those locations that do not have that texture.

The combination of height, slope, proximity and texture based criteria all contribute to the overall fitness of a location and this influences the likely hood that a spawner will spawn that resource there.

### *Spawn Extensions:*

Spawner extensions enable the spawning you to take advantage of the existing infrastructure provided by Gaia and then to extend it.

They need to be added to prefabs, and any number of them can be added or stacked onto a resource.

To create a Spawn Rule Extension derive your class from `SpawnRuleExtension` in Gaia / Scripts / `SpawnRuleExtension.cs`.

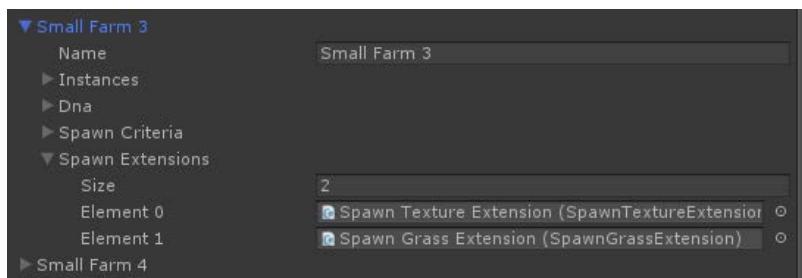
The over-ridable methods exposed are:

- Initialise – called by the spawner before a spawn iteration is called
- AffectTextures – return true if your extension affects textures
- AffectsDetails – return true if your extension affects details
- GetFitness – override and update fitness if your extension determines fitness – example – you may want to implement custom logic to evaluate the location passed in
- OverridesSpawn – return true if you want to disable standard resource spawning and implement your own
- Spawn – override to provide custom spawning behaviour
- PostSpan – override to provide addition behaviour after the resource has spawned.

The DNA has a parameter `m_ExtParam` which can be passed through to the extension to provide instance specific configuration. The same parameter has also been provided to each instance as well on game object prototypes.

To use an extension create an instance of it, configure it, and then save it as a prefab. These prefabs can then be added to the prototypes.

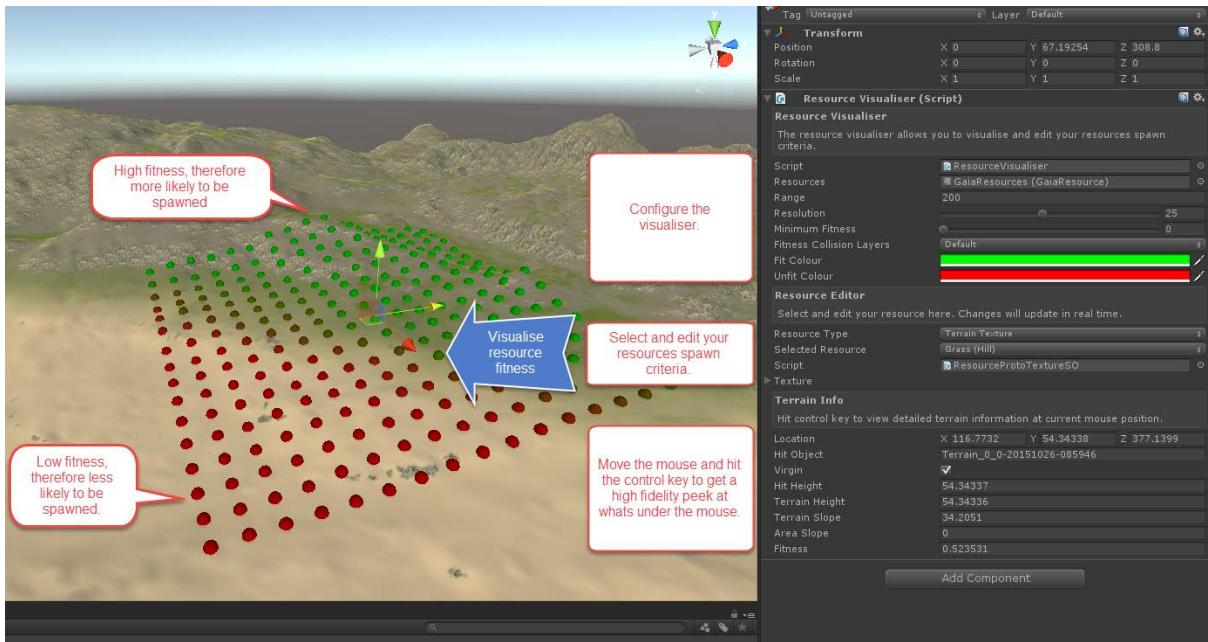
Two sample extensions have been provided in `Gaia/Scripts/SpawningSystem/SpawnExtensions`, and these are used on the default `GameObject` resources.



## Use the Visualiser!!

While you can edit your spawn criteria manually, the Visualiser makes this process much easier as it allows you to edit and visualise the impact of spawn criteria settings in real time.

To access the visualiser select Show Visualiser in the Utilities section of Gaia Manager. To learn more about the visualiser check out the Visualiser section of the manual.



## Sea Level:

Sea Level is the absolute physical height of the sea or water table in your scene, and all spawn criteria heights are defined relative to this.

Sea Level is important because nature inherently does things differently above and below the water, and Gaia needs to support this.

While sea level is defined as an absolute height in the scene, Gaia uses height relative to this in its spawn criteria. For example a spawn criteria height of +5m means '5 meters above sea level', whereas -5m means '5 meters below sea level'.

The reason Gaia uses relative spawn height instead of the absolute height is to allow you to design your spawn criteria so that they will work regardless of what the height of the sea is. This allows you to change the height of the sea from scene to scene and know that the resource will always spawn at the correct height.

If you change your Sea Level then any Spawner that references these resources will need to be Reset and re-Spawned. Ideally you should set your Sea Level at the beginning of scene creation process and then never change it. You can visualise the Sea Level by enabling Show Sea Level in your Stamper.

The Sea Level in your stamper is sourced from the Sea Level in the session, which is in turn sourced from the Sea Level in the defaults file that was used to create the session.

## Work Flow

### *Create a New Resources File*

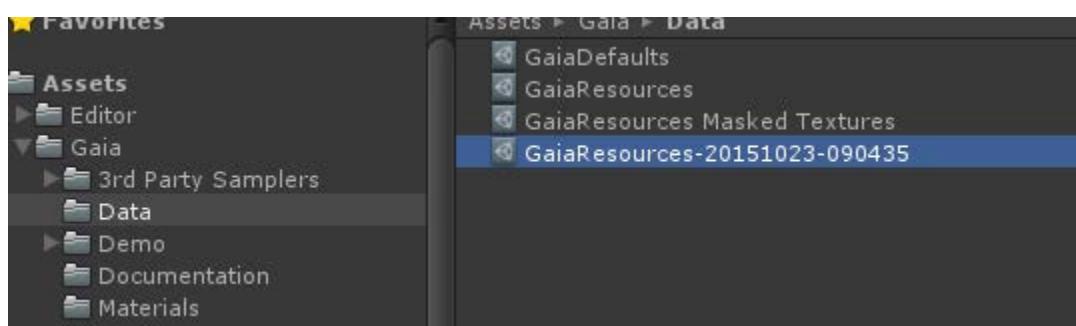
Open Gaia Manager.

Select your Defaults file (the default values for Sea Level and Beach Height flow through to the new Resources file)

Click the New button next to Resources.



This will create a new and empty Resources file in the Gaia \ Data directory and select it in Gaia Manager.



The naming standard is GR-YYYYMMDD-HHMMSS.

### *Duplicate an existing Resources File*

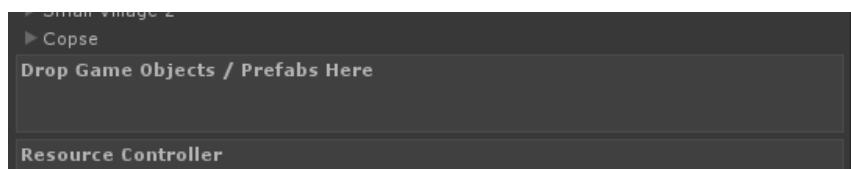
When experimenting with fitness and other settings it's a good idea to make a backup or duplicate of your existing resources file.

Select the resources file you want to duplicate.

Hit Ctrl-D to duplicate it. Or alternatively select the Edit -> Duplicate menu.

### *Drop Game Objects / Prefabs Here*

The easiest way to add Game Object prototypes is to drag the Game Objects or Prefabs onto the pad in the resources file.

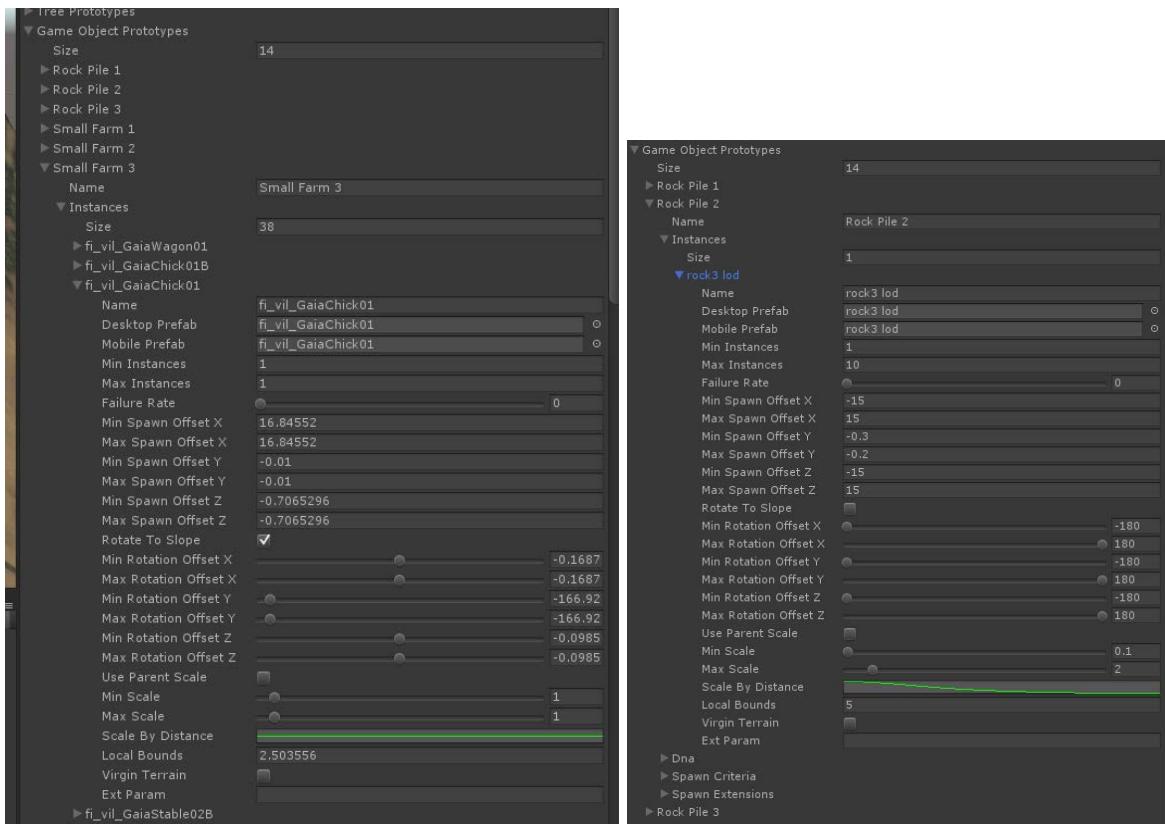


Note that only Game Objects that have been instantiated from a Prefab, or Prefabs themselves can be used as Game Object Prototypes.

This will load the Game Object or Prefab, determine its physical size for its DNA, create some default settings, and add this to the Game Object prototypes list ready to be spawned by a spawner.

You can create Points Of Interest (POI) by dragging multiple Game Objects as a group. In this case Gaia will preserve the height, rotation, and relative layout of the group as a whole.

When creating POI they should be done on a flat surface where  $y = 0$  is the equivalent of ground level. Because your scenes are never truly flat it is best to embed the Game Objects such that they are slightly below ground level so that when they are instantiated in the spawner they can handle slight variations on slope.



A Game Object instance contains the information needed to position the instance in the scene, and all offsets are provided with respect to the spawn location and rotation provided by the spawner.

In the example above just one instance of the chicken house will be placed specifically in the scene, with respect to the spawn location, whereas in the other example above one game object instance will be spawned many times in the scene around the spawn location.

### *Set Asset Association*

This will record the file names of each resource referred to inside of the resource file such that when the resource file used on another system it will attempt to use resources with the same file name regardless of where they are loaded in the project. This is done automatically, and is not something you would normally need to do.

### *Associate Assets*

This will attempt to associate any previously un-associated assets in your resource file with the matching assets in the target system. Asset to resource matching is done by file name, and happens automatically when a spawn happens. This can be problematic when there are multiple resources with the same file name in the target project, but can easily be fixed by updating the resource references with the correct

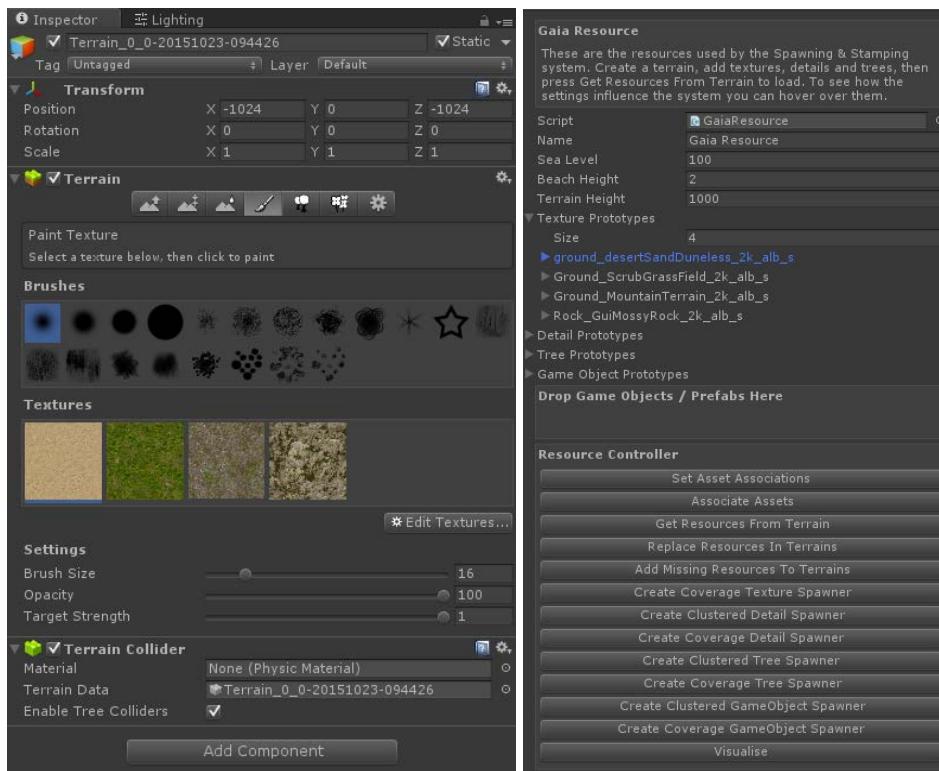
file names.

### Get Resources From Terrain

This will load the Texture, Detail and Tree resources defined in your terrain into your resources file and apply some default spawn criteria to them.

It is a handy way of quickly populating your resources, or updating them as you make changes to your terrain's resources.

If you change the content of terrain, for example, swapping one texture for another and hit this again, it will update the physical settings, but leave the existing spawn criteria in place so that you don't lose them.



In this image you can see the terrain texture resources reflected in the resources file.

Note: By convention Gaia expects texture slot 0 to be sand / base, texture slot 1 to be grass 1, texture slot 2 to be grassy rock, and texture slot 3 to be rock, however you can configure this any way you want.

### *Replace Resources In Terrains*

Will take the resources defined in your resources file and apply them to any terrains in your scene, wiping out any previously assigned resources that may have been in your terrain.

It is a handy way of quickly populating your terrains resources. This function is also called by the Gaia Manager when creating new terrains so that they are automatically pre-loaded with the correct resources.

Note: Because this overwrites other resources in the terrain please use with care.

### *Add Missing Resources To Terrains*

This adds any resources that are in your resource file, but not in the terrain into your terrain.

This is a key component in the concept of composable resources. The idea is to have multiple resource files each of which is customised to the type of resource – for example one resource file per asset pack.

You can either add all of those resources at once using this method – which could lead to performance issues in Unity, or just add the resources that will be spawned.

The spawner itself will ensure that only the missing resources are applied, so this function would be used rarely.

### *Create Coverage Texture Spawner*

Creates a spawner, configured to provide broad terrain texture coverage and populates it with the textures in this resource file.

Note: The configuration has been optimised for a four texture terrain where texture slot zero is sand / base, texture slot 1 is grass 1, texture slot 2 is grass / rock, and texture slot 3 is rock. You can add more textures and customise as these settings have been configured for convenience only.

### *Create Clustered Detail Spawner*

Creates a spawner, configured to provide clusters of grasses / details and populates it with the details in this resource file.

### *Create Coverage Detail Spawner*

Creates a spawner, configured to provide broad terrain coverage and populates it with the details in this resource file.

### *Create Clustered Tree Spawner*

Creates a spawner, configured to provide clusters of trees and populates it with the trees in this resource file.

### *Create Coverage Tree Spawner*

Creates a spawner, configured to provide broad terrain coverage and populates it with the trees in this resource file.

### *Create Clustered GameObject Spawner*

Creates a spawner, configured to provide clusters of game objects and populates it with the game objects in this resource file.

### *Create Coverage GameObject Spawner*

Creates a spawner, configured to provide broad terrain coverage and populates it with the game objects in this resource file.

### *Visualise*

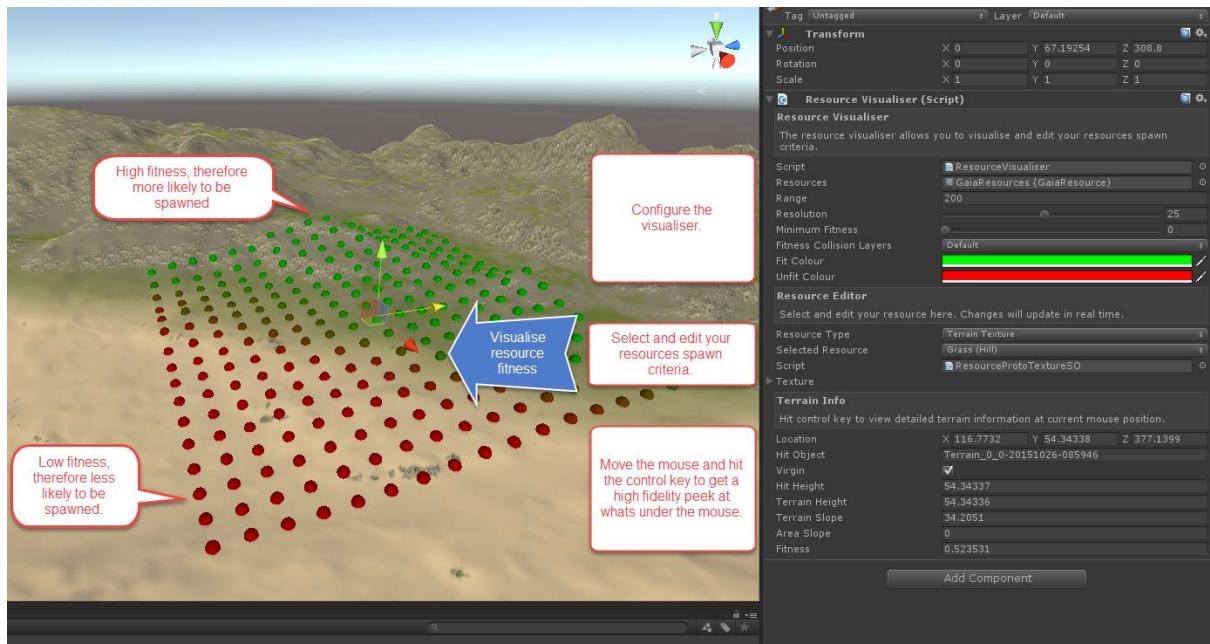
Shows the visualiser.

The visualiser allows you to fine tune the fitness criteria for a resource. As you change the resource settings in the visualiser, they are automatically reflected back in your resources file.

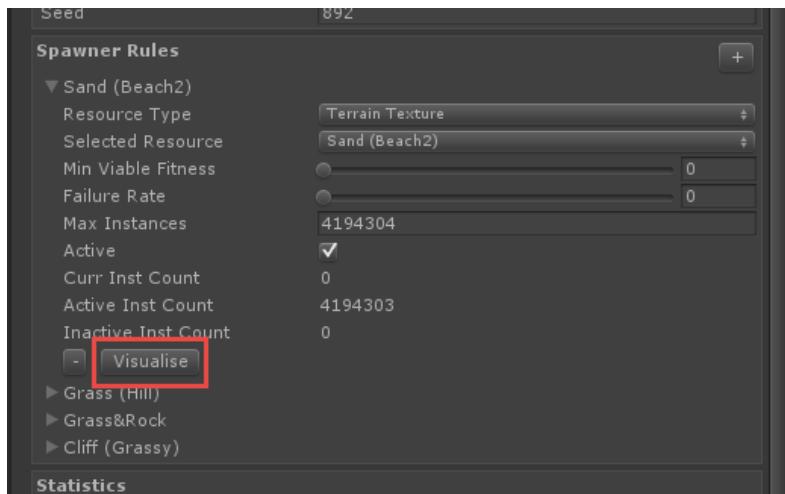
See the visualiser section below to learn more about it.

## Visualiser – Visualise and Edit Spawn Criteria

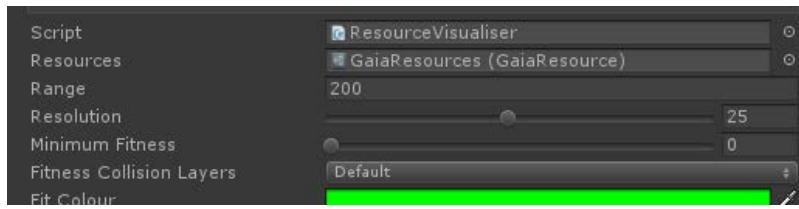
The Visualiser allows you to visualise which areas of terrain your resources view as 'fit', and then adjust this by editing your spawn criteria.



Access the Visualiser from the Utilities section in Gaia Manager, or by clicking the Visualise rule next to a resource in a Spawner.



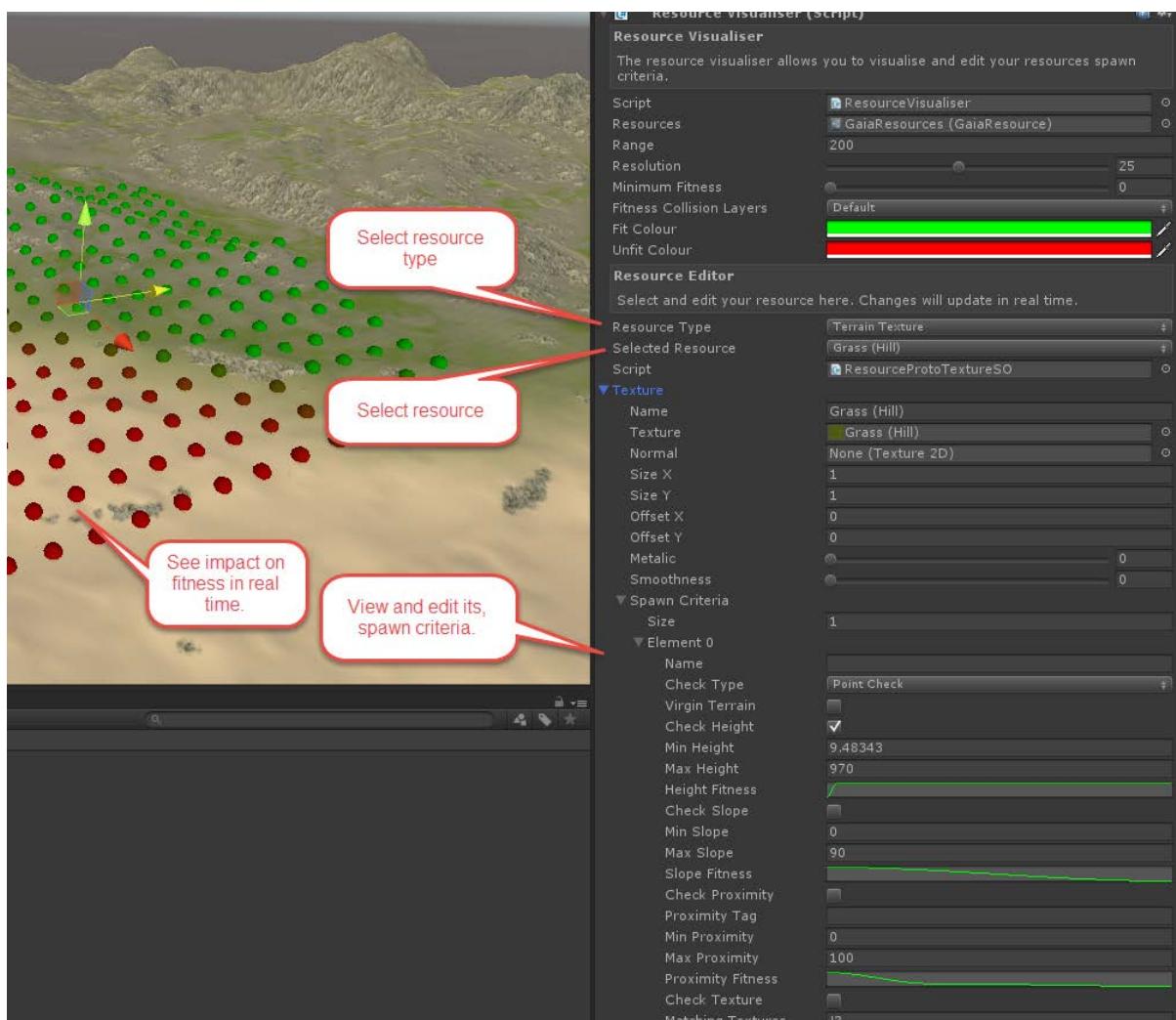
Move the visualiser by selecting and dragging it. It will automatically adapt itself to the terrain and show a visualisation fitness at that location.



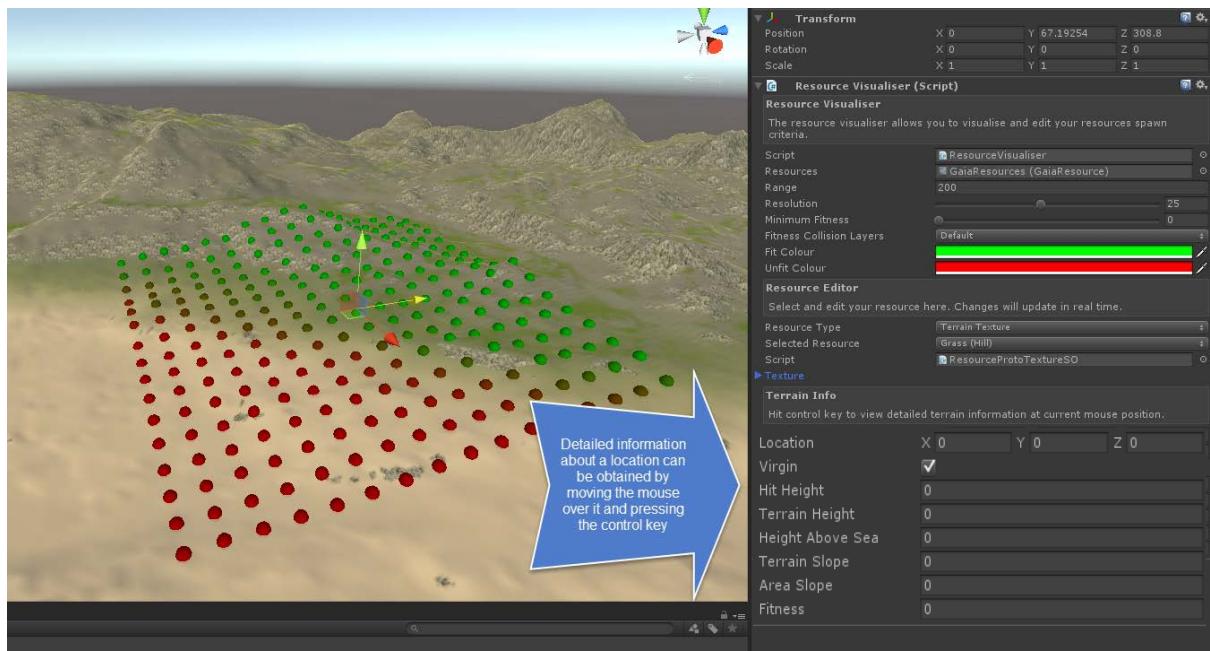
Adjust the area covered by the visualiser by adjusting the Range, and you change its resolution by adjusting the Resolution, however because the visualiser reflects updates in real time adjusting these can have a negative effect on Editor performance, so its best to keep range relatively small, and resolution relatively large.

Gaia uses raycasts to detect collisions, so the fitness collision layer needs to be selected so that it is correct for your environment. The visualiser will not display unless this has not been configured correctly.

To change what's being visualised select the resource type and resource.



All edits are automatically reflected back in the resources file, and will influence how and where the spawners spawn this resource.



For more detail about a specific location move your mouse over it while also holding the control key. Note: You will need to constrain the mouse to the area covered by the Visualiser for this to work.

### **Unity 2017.1+ NOTE**

From Unity 2017 onwards the curve editing capability in the visualiser has been broken. To work around this, select your visualiser, select the resource you want to edit, and then click on the resource file itself. Changes to your resources will be correctly reflected in your visualiser in real time.

When you are ready to select the next resource then go back to your visualiser, select it, and then go back to your resources file to edit it.

This will be rewritten from scratch in Gaia 2.

## General Settings

General resource settings.

### Name

The resource name. Purely informational.

### Sea Level

The absolute height of the sea or water table in meters. All spawn criteria heights are calculated relative to this. This can also be thought of as the water level. This value is set by the session manager, and sourced from the sea level in your defaults file.

NOTE:

\* If you change the sea level, then this will affect the way that all spawners that refer to this resources file will spawn. Ideally you should set your sea level once at the beginning of creating a scene and never change it.

## Beach Height

The beach height in meters. Beaches are spawned at sea level and are extended for this height above sea level. This is used when creating default spawn rules in order to create a beach in the zone between water and land.

## Terrain Height

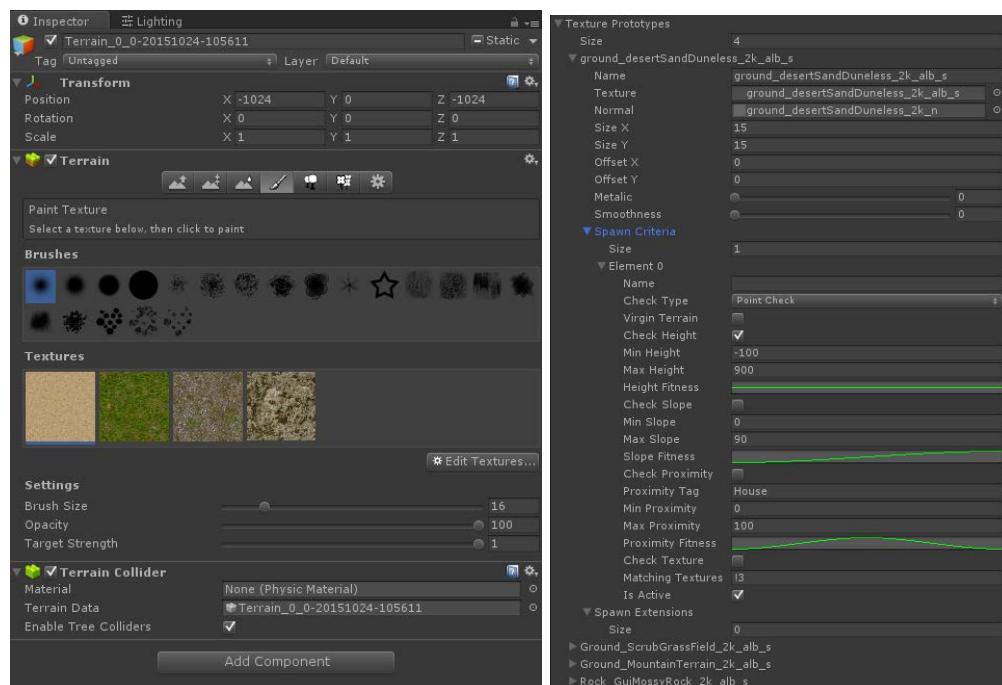
The total terrain height in meters.

## Texture Prototype Settings

Texture prototypes are used to populate the textures in your terrain, and then used by the texture spawner to texture your terrain.

When texturing terrains, the strength of texture can be a value between 0 and 1, and this controls the opacity at which the texture will be painted on the terrain. The inherent impact of this is that all of the textures must conform to the same rule, and this means that the sum of the strength all of the textures at a particular location must be 1.

With Gaia fitness is also a value between 0 and 1. When Gaia applies this texture to the terrain it will set the strength of the texture at that location to the fitness that was returned at that location, and proportionally reduce the strength of any other texture that were previously at that location in order to maintain the sum of one terrain constraint. The implication of this in spawning is that every additional texture rule will paint over the previous one.



Terrain Textures

Gaia Texture Prototypes

## Texture Prototype Physical Settings

Physical settings of the texture prototype.

Name

The texture name. Purely informational.

Texture

The texture.

Normal

The normal texture.

Size X

The width over which the image will stretch on the terrain's surface.

Size Y

The height over which the image will stretch on the terrain's surface.

Offset X

How far from the terrain's anchor point the tiling will start.

Offset Y

How far from the terrain's anchor point the tiling will start.

Metalic

Controls the overall metalness of the surface.

Smoothness

Controls the overall smoothness of the surface.

## Texture Prototype Spawn Criteria

An array of spawn criteria that are evaluated against the physical terrain and its surrounds in order to determine the fitness of a location.

There can be more than one spawn criteria defined per resource, and each criteria can select for a different type of location. When assessing a location, the fitness used is the one that returns the highest fitness. Treated as an OR function – take the highest value from this criteria OR this criteria etc.

Within a criteria the fitness is treated as an AND function, and what is returned is the lowest fitness. So if for example height AND slope was selected for evaluation, the fitness returned would be the lowest value.

## Name

The criteria name. Purely informational.

## Check Type

The type of terrain check to make. A Point Check will check a single point on the terrain, and a Bounded Area Check will check that point plus an additional radius that is determined by the Bounds Radius in the DNA. Area based checks are good for checking the terrain for larger structures but is substantially slower so use with care.

## Virgin Terrain

When selected, the criteria will only be valid if the terrain was clear of any other objects at this location. A location is determined to be ‘virgin’ when a raycast collision test hits clear terrain. To detect other objects at this location they must have colliders. You can use an invisible collider and this test to stop any resources that require virgin terrain from spawning at that location. Trees will have to have colliders enabled in order to be detected.

## Check Height

Whether or not this location will be checked for height. Heights here are relative to sea level, + is above, 0 is at, and - is below.

### Min Height

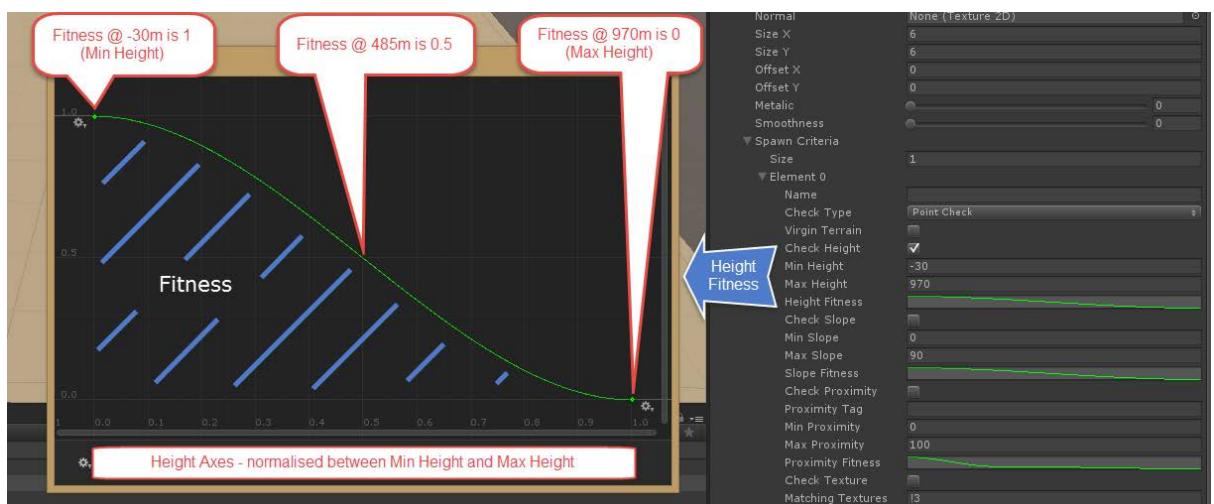
The minimum valid height relative to sea level. Only tested when Check Height is checked.

### Max Height

The maximum valid height relative to sea level. Only tested when Check Height is checked.

### Height Fitness

The fitness curve – a curve evaluated between the minimum and the maximum height when Check Height is checked. The green curve represents the fitness over that range.



## Check Slope

Whether or not this location will be checked for slope. Slopes range from 0 degrees which is horizontal, to 90 degrees which is vertical.

### Min Slope

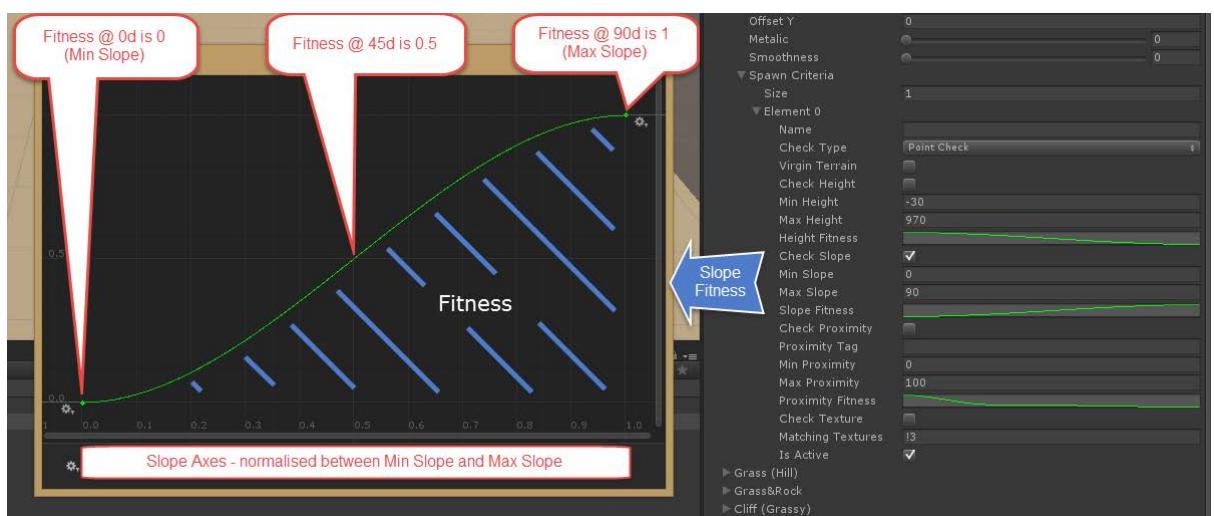
The minimum valid slope. Only tested when Check Slope is checked.

### Max Slope

The maximum valid slope. Only tested when Check Slope is checked.

### Slope Fitness

The fitness curve - a curve evaluated between the minimum and the maximum slope when Check Slope is checked. The green curve represents the fitness over that range.



## Check Proximity

Whether or not to check proximity (in meters) to game objects that have the specified tags.

### Proximity Tag

The Tag that will be checked for. Only tested when Check Proximity is checked.

### Min Proximity

The minimum proximity to the tagged object in meters. Only tested when Check Proximity is checked.

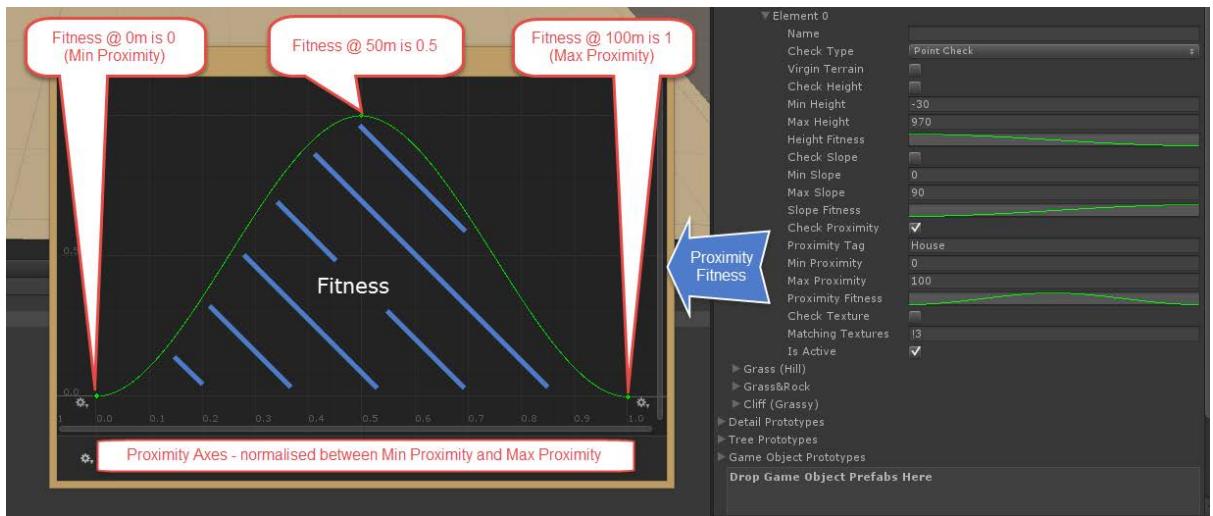
### Max Proximity

The maximum proximity to the tagged object in meters. Only tested when Check Proximity is checked.

### Proximity Fitness

The fitness curve - a curve evaluated between the minimum and the maximum proximity when Check Proximity is checked. The green curve represents the

fitness over that range.



### Check Texture

Whether or not to check the texture at this location.

### Matching Textures

Texture slots from your terrain (first valid slot is 0). Will select for presence that texture. Use exclamation mark in front of slot to select for absence of that texture. For example 3 selects for presence of texture 3, !3 checks for absence of texture 3. The fitness returned is the strength of the texture at that location in range 0..1. Only checked when Check Texture is checked.

### Is Active

Whether or not this spawn criteria is active.

### Texture Prototype Spawn Extensions

An array of spawn rule extension prefabs. Spawn Rule Extensions can be used to extend and modify the way the spawner spawns. Read more about these at the beginning of the resources section of the manual.

## Detail Prototype Settings

Detail prototypes are used to populate the details in your terrain, and then used by the detail spawner to populate the details in your terrain.

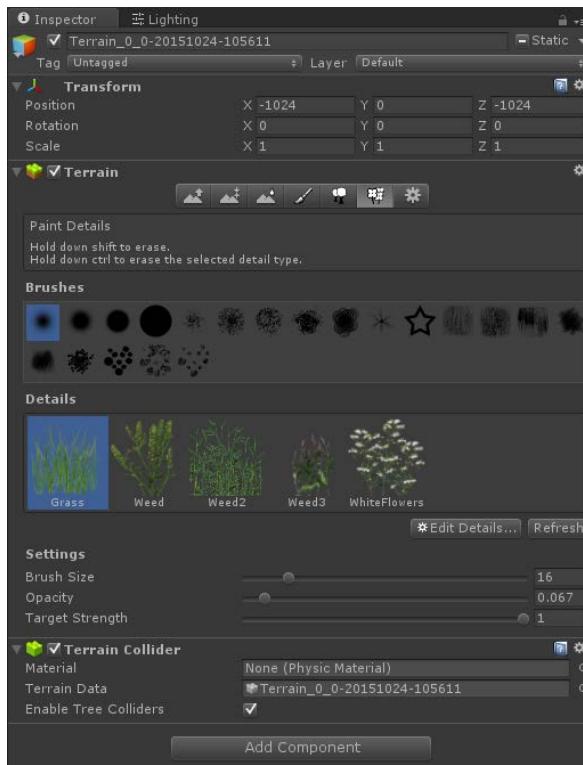
A typical use for details is grass, but also often used for small rocks and other low poly details.

When detailing terrains, the strength of detail at a location can be a value between zero and fifteen, and this controls the density at which the detail will be painted on the terrain. Multiple details can all be painted at the same location, and each can have its own strength.

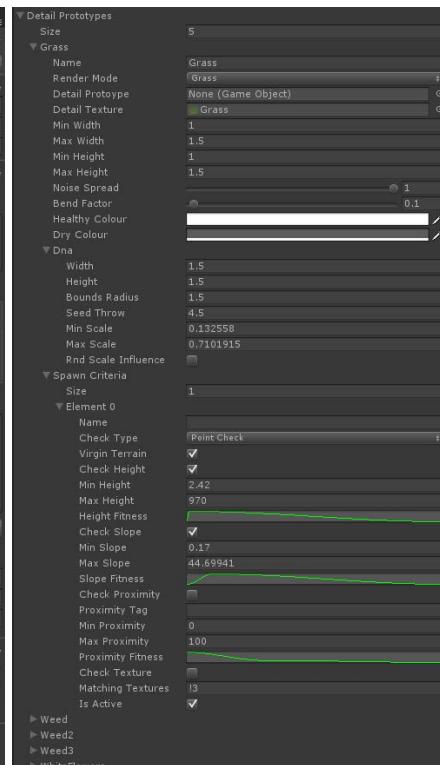
When Gaia applies this detail to the terrain it will set the strength of the detail at that location to a value somewhere between the minimum and maximum scales in its DNA multiplied by its maximum strength. For example a scale of 0 would be assigned a strength of zero, and a scale of 1 would be assigned a strength of 15.

If *Rnd Scale Influence* is set in the DNA, then this will be a random number somewhere between the min and max scale, otherwise it will be the fitness of that location normalised between the min and max scale.

The benefit of using a fitness based strength is that your grass will grow more densely where conditions are better for it i.e. where the land is best suited for it to grow.



Terrain Details



Gaia Detail Prototypes

## Detail Prototype Physical Settings

Physical settings of the detail prototype.

Name

The detail name. Purely informational.

Render Mode

The detail render mode.

Detail Prototype

The detail prototype. Used when vertex lit mode is selected.

Detail Texture

The texture that represents the grass. Used by grass and billboard grass render mode.

Min Width

Minimum width. Lower limit of the width of the clumps of grass that are generated.

Max Width

Maximum width. Upper limit of the width of the clumps of grass that are generated.

Min Height

Minimum height. Lower limit of the height of the clumps of grass that are generated.

Max Height

Maximum height. Upper limit of the height of the clumps of grass that are generated.

Noise Spread

Controls the approximate size of the alternating patches, with higher values indicating more variation within a given area.

Bend Factor

Controls the degree to which the grass will bend based on terrain settings.

Healthy Colour

Healthy grass clump colour.

Dry Colour

Dry grass clump colour.

## Detail Prototype DNA

Contains sizing, clear area bounds radius, spread and scale information that is used by the spawner to control how and where, and how strongly the details will be spawned.

### Width

The width in world units.

### Height

The height in world units.

### Bounds Radius

Radius from centre of object in world units for bounded area checks. Make this larger if you want more free space around your object when it is spawned.

### Seed Throw

The maximum distance a seed can be thrown when a new instance is spawned. Used to control spread area random clustered spawning.

### Min Scale

The minimum strength that detail will render at.

### Max Scale

The maximum strength that detail will render at. Should not be greater than 1.

### Rnd Scale Influence

Randomises the scaled strength somewhere between minimum and maximum scale. If not selected then the scaled strength will be proportionally influenced by the locations fitness.

### Ext Param

The extension parameter. Can be used to optionally provide custom configuration information through to any spawn rule extensions.

## **Detail Prototype Spawn Criteria**

An array of spawn criteria that are evaluated against the physical terrain and it's surrounds in order to determine the fitness of a location.

There can be more than one spawn criteria defined per resource, and each criteria can select for a different type of location. When assessing a location, the fitness used is the one that returns the highest fitness. Treated as an OR function – take the highest value from this criteria OR this criteria etc.

Within a criteria the fitness is treated as an AND function, and what is returned is the lowest fitness. So if for example height AND slope was selected for evaluation, the fitness returned would be the lowest value.

### **Name**

The criteria name. Purely informational.

### **Check Type**

The type of terrain check to make. A Point Check will check a single point on the terrain, and a Bounded Area Check will check that point plus an additional radius that is determined by the Bounds Radius in the DNA. Area based checks are good for checking the terrain for larger structures but is substantially slower so use with care.

### **Virgin Terrain**

When selected, the criteria will only be valid if the terrain was clear of any other objects at this location. A location is determined to be 'virgin' when a raycast collision test hits clear terrain. To detect other objects at this location they must have colliders. You can use an invisible collider and this test to stop any resources that require virgin terrain from spawning at that location. Trees will have to have colliders enabled in order to be detected.

### **Check Height**

Whether or not this location will be checked for height. Heights here are relative to sea level, + is above, 0 is at, and - is below.

#### **Min Height**

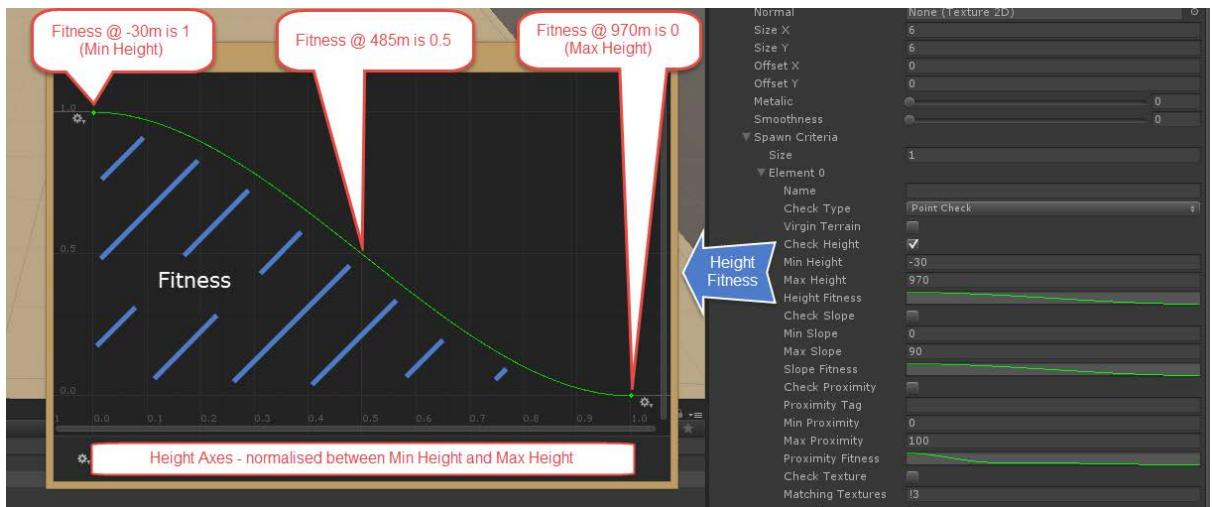
The minimum valid height relative to sea level. Only tested when Check Height is checked.

#### **Max Height**

The maximum valid height relative to sea level. Only tested when Check Height is checked.

#### **Height Fitness**

The fitness curve – a curve evaluated between the minimum and the maximum height when Check Height is checked. The green curve represents the fitness over that range.



## Check Slope

Whether or not this location will be checked for slope. Slopes range from 0 degrees which is horizontal, to 90 degrees which is vertical.

### Min Slope

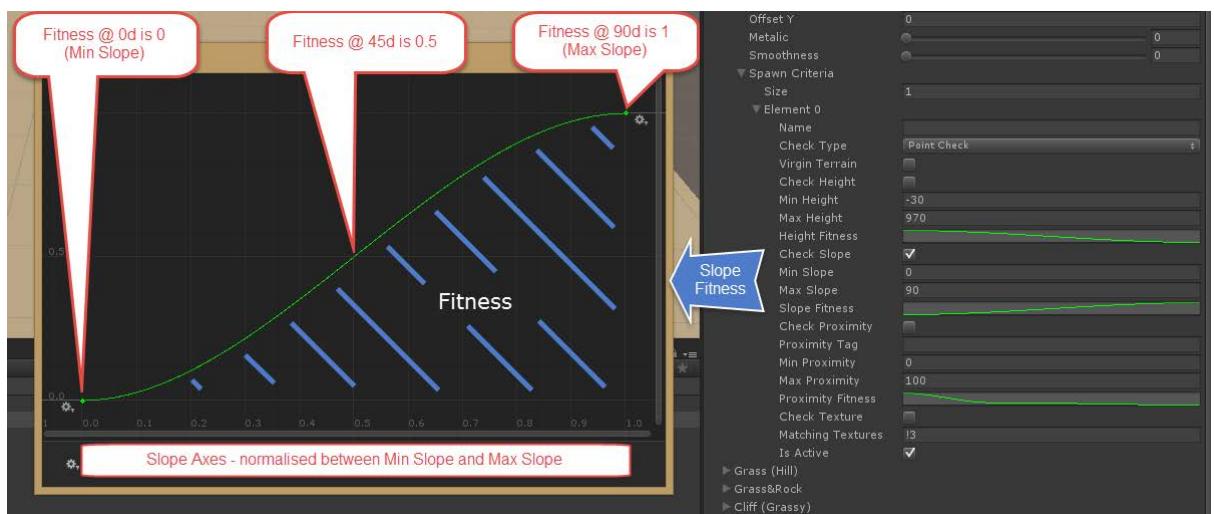
The minimum valid slope. Only tested when Check Slope is checked.

### Max Slope

The maximum valid slope. Only tested when Check Slope is checked.

### Slope Fitness

The fitness curve - a curve evaluated between the minimum and the maximum slope when Check Slope is checked. The green curve represents the fitness over that range.



## Check Proximity

Whether or not to check proximity (in meters) to game objects that have the specified tags.

### Proximity Tag

The Tag that will be checked for. Only tested when Check Proximity is checked.

### Min Proximity

The minimum proximity to the tagged object in meters. Only tested when Check Proximity is checked.

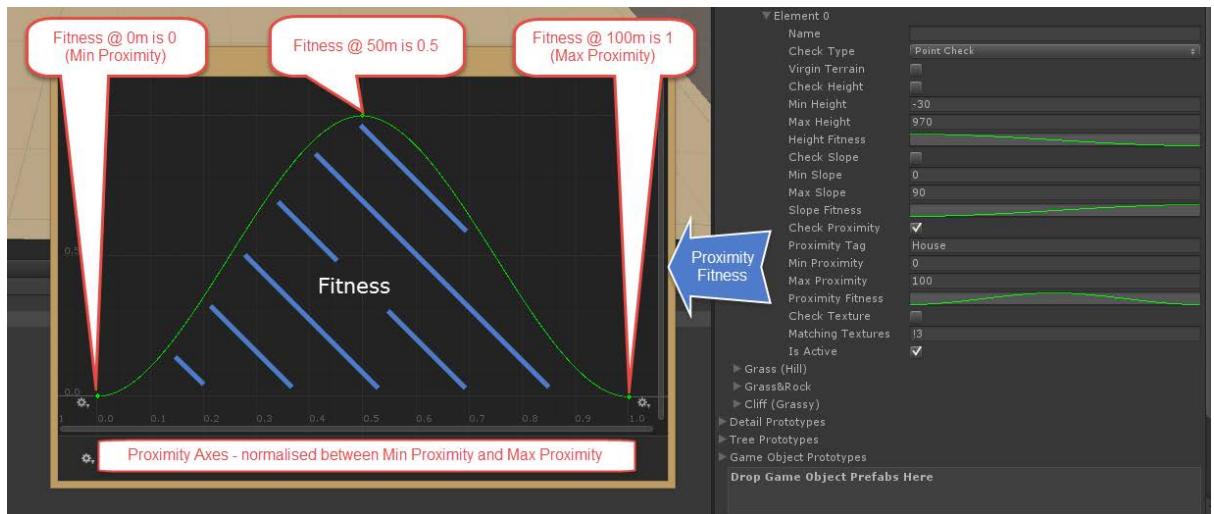
### Max Proximity

The maximum proximity to the tagged object in meters. Only tested when Check Proximity is checked.

### Proximity Fitness

The fitness curve - a curve evaluated between the minimum and the maximum proximity when Check Proximity is checked. The green curve represents the

fitness over that range.



### Check Texture

Whether or not to check the texture at this location.

### Matching Textures

Texture slots from your terrain (first valid slot is 0). Will select for presence that texture. Use exclamation mark in front of slot to select for absence of that texture. For example 3 selects for presence of texture 3, !3 checks for absence of texture 3. The fitness returned is the strength of the texture at that location in range 0..1. Only checked when Check Texture is checked.

### Is Active

Whether or not this spawn criteria is active.

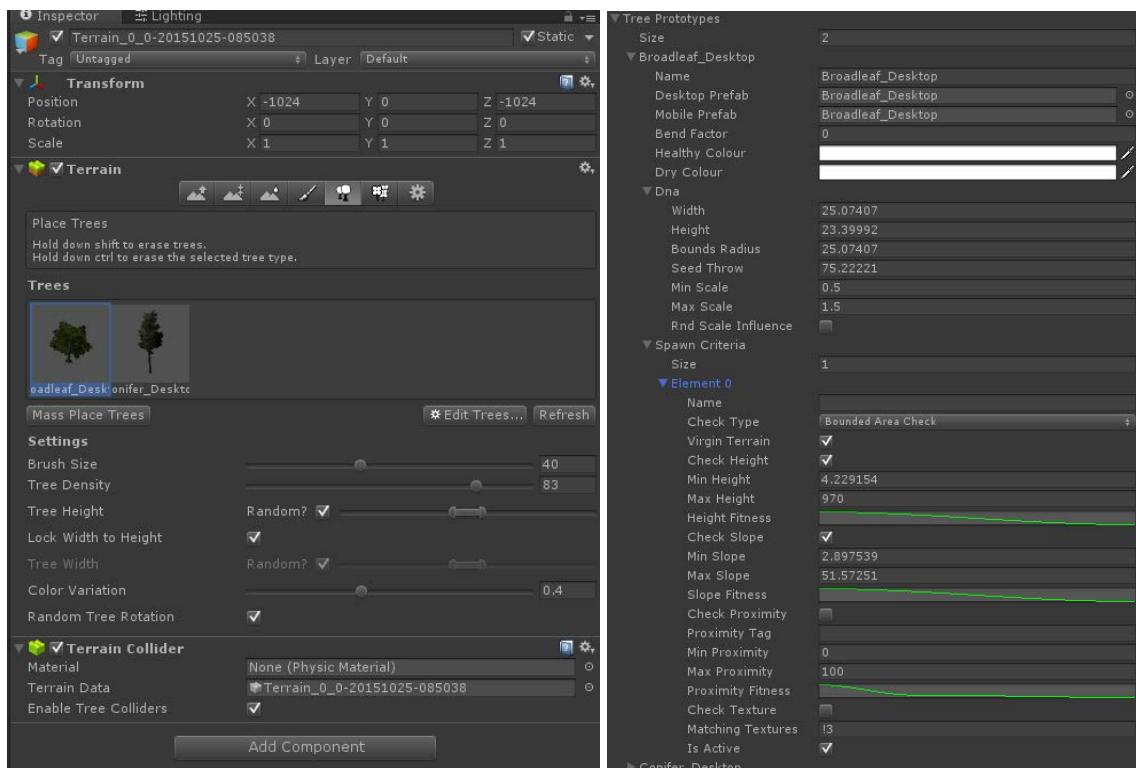
### *Detail Prototype Spawn Extensions*

An array of spawn rule extension prefabs. Spawn Rule Extensions can be used to extend and modify the way the spawner spawns. Read more about these at the beginning of the resources section of the manual.

## Tree Prototype Settings

Tree prototypes are used to populate the trees in your terrain, and then used by a tree spawner to place terrain trees into your terrain.

*Tip:* Because Tree prototypes leverage tree prefabs, you can also place trees with a game object spawner, and using a game object spawner enables your trees to gain additional placement and rotation options and behaviours (although generally with the impact of greater performance cost).



Terrain Trees

Gaia Tree Prototypes

## Tree Prototype Physical Settings

Physical settings of the tree prototype.

### Name

The tree name. Purely informational.

### Desktop Prefab

The prefab for the desktop version of the tree.

### Mobile Prefab

The prefab for the mobile version of the tree. This for future proofing and is not currently used.

## Bend Factor

How much the tree bends in the wind - only used by unity tree creator trees, ignored by SpeedTree trees.

## Healthy Colour

The colour of healthy trees - only used by unity tree creator trees, ignored by SpeedTree trees.

## Dry Colour

The colour of dry trees - only used by unity tree creator trees, ignored by SpeedTree trees.

## Tree Prototype DNA

Contains sizing, clear area bounds radius, spread and scale information that is used by the spawner to control how and where, and how large the trees will be spawned. Width, height are calculated automatically from the tree prefab, and bounds radius and seed throw are given generic defaults that you can override.

### Width

The width in world units.

### Height

The height in world units.

### Bounds Radius

Radius from centre of object in world units for bounded area checks. Make this larger if you want more free space around your object when it is spawned.

### Seed Throw

The maximum distance a seed can be thrown when a new instance is spawned. Used to control spread area random clustered spawning.

### Min Scale

The minimum scale that this tree can be rendered into the world.

### Max Scale

The maximum scale that this tree can be rendered into the world.

### Rnd Scale Influence

Randomises the scale somewhere between minimum and maximum scale. If not selected then the scale will be proportionally influenced by the locations fitness.

### Ext Param

The extension parameter. Can be used to optionally provide custom configuration information through to any spawn rule extensions. Note: adding 'nograss' here

will stop grass from spawning under the tree in the radius defined by its dna.

### **Tree Prototype Spawn Criteria**

An array of spawn criteria that are evaluated against the physical terrain and it's surrounds in order to determine the fitness of a location.

There can be more than one spawn criteria defined per resource, and each criteria can select for a different type of location. When assessing a location, the fitness used is the one that returns the highest fitness. Treated as an OR function – take the highest value from this criteria OR this criteria etc.

Within a criteria the fitness is treated as an AND function, and what is returned is the lowest fitness. So if for example height AND slope was selected for evaluation, the fitness returned would be the lowest value.

#### **Name**

The criteria name. Purely informational.

#### **Check Type**

The type of terrain check to make. A Point Check will check a single point on the terrain, and a Bounded Area Check will check that point plus an additional radius that is determined by the Bounds Radius in the DNA. Area based checks are good for checking the terrain for larger structures but is substantially slower so use with care.

#### **Virgin Terrain**

When selected, the criteria will only be valid if the terrain was clear of any other objects at this location. A location is determined to be 'virgin' when a raycast collision test hits clear terrain. To detect other objects at this location they must have colliders. You can use an invisible collider and this test to stop any resources that require virgin terrain from spawning at that location. Trees will have to have colliders enabled in order to be detected.

#### **Check Height**

Whether or not this location will be checked for height. Heights here are relative to sea level, + is above, 0 is at, and - is below.

##### **Min Height**

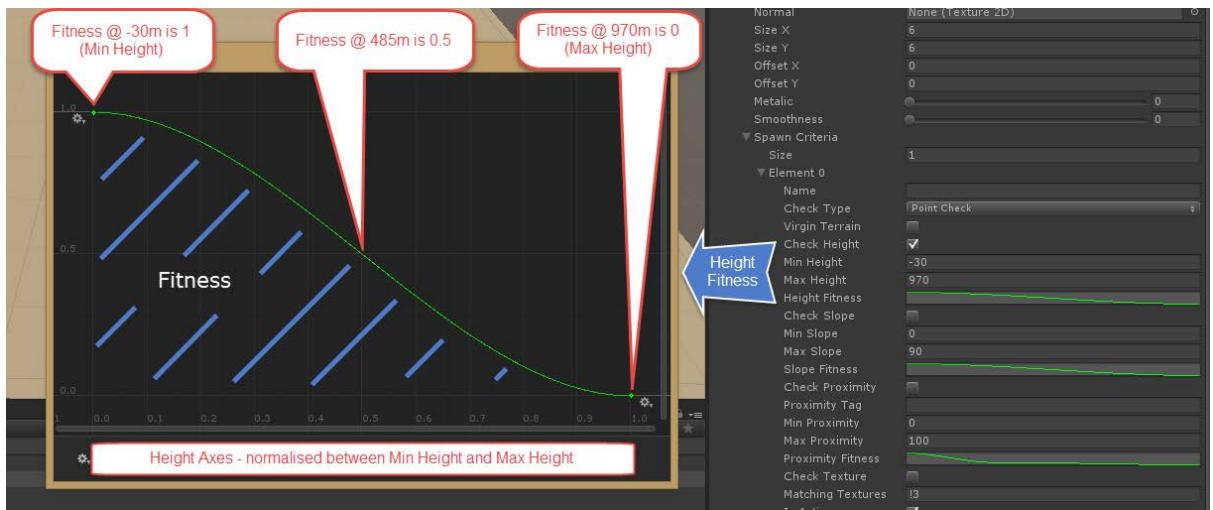
The minimum valid height relative to sea level. Only tested when Check Height is checked.

##### **Max Height**

The maximum valid height relative to sea level. Only tested when Check Height is checked.

##### **Height Fitness**

The fitness curve – a curve evaluated between the minimum and the maximum height when Check Height is checked. The green curve represents the fitness over that range.



## Check Slope

Whether or not this location will be checked for slope. Slopes range from 0 degrees which is horizontal, to 90 degrees which is vertical.

### Min Slope

The minimum valid slope. Only tested when Check Slope is checked.

### Max Slope

The maximum valid slope. Only tested when Check Slope is checked.

### Slope Fitness

The fitness curve – a curve evaluated between the minimum and the maximum slope when Check Slope is checked. The green curve represents the fitness over that range.



## Check Proximity

Whether or not to check proximity (in meters) to game objects that have the specified tags.

### Proximity Tag

The Tag that will be checked for. Only tested when Check Proximity is checked.

### Min Proximity

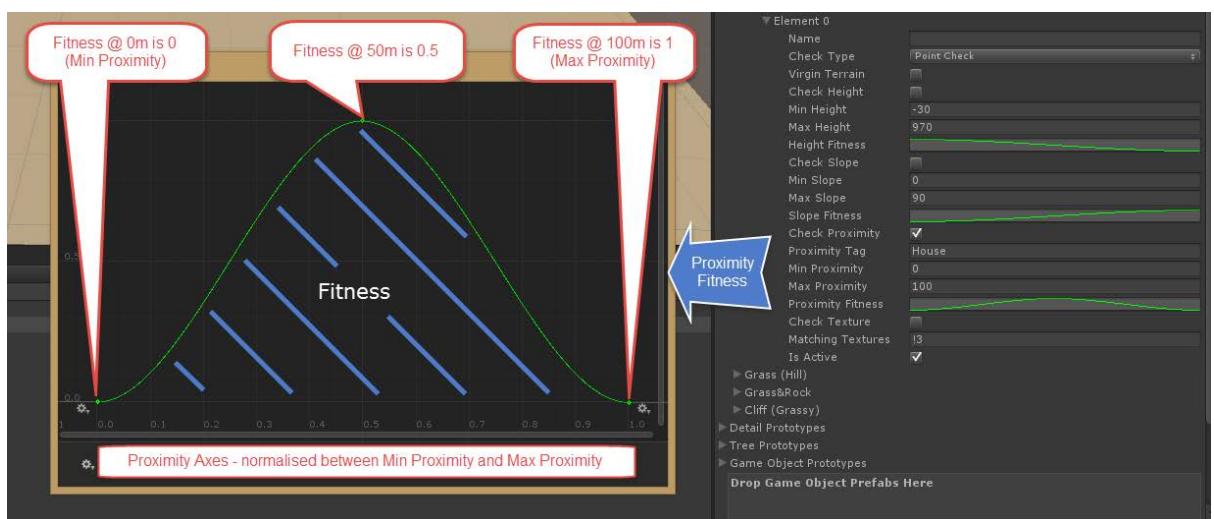
The minimum proximity to the tagged object in meters. Only tested when Check Proximity is checked.

### Max Proximity

The maximum proximity to the tagged object in meters. Only tested when Check Proximity is checked.

### Proximity Fitness

The fitness curve - a curve evaluated between the minimum and the maximum proximity when Check Proximity is checked. The green curve represents the fitness over that range.



## Check Texture

Whether or not to check the texture at this location.

### Matching Textures

Texture slots from your terrain (first valid slot is 0). Will select for presence that texture. Use exclamation mark in front of slot to select for absence of that texture. For example 3 selects for presence of texture 3, !3 checks for absence of texture 3. The fitness returned is the strength of the texture at that location in range 0..1. Only checked when Check Texture is checked.

## Is Active

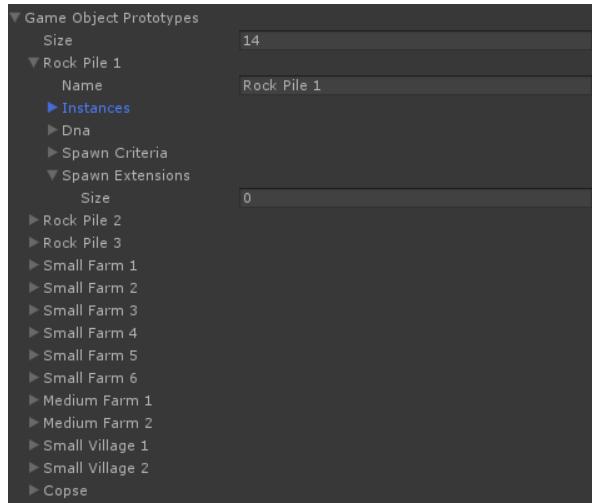
Whether or not this spawn criteria is active.

### ***Tree Prototype Spawn Extensions***

An array of spawn rule extension prefabs. Spawn Rule Extensions can be used to extend and modify the way the spawner spawns. Read more about these at the beginning of the resources section of the manual.

## Game Object Prototype Settings

Game Object prototypes are used by Game Object spawner to place instances of Game Objects into your scene.



Gaia Game Object Prototypes

### Game Object Prototype Physical Settings

Physical settings of the game object prototype.

#### Name

The game object name. Purely informational.

#### Game Object Prototype Instances

Contains an array of Game Objects Prototypes that will be spawned and information on how to rotate and place them.

#### Name

Instance name. Is picked up automatically when a Game Object Prefab is dropped onto the Drop Game Object Prefabs here slot.

#### Desktop Prefab

The desktop prefab that will be spawned.

#### Mobile Prefab

The mobile prefab that will be spawned. Not currently used.

## Min Instances

The minimum instances that will be attempted to spawn.

## Max Instances

The maximum instances that will be attempted to spawn.

## Failure Rate

The chance that an instance spawn attempt will fail. Makes it possible to spawn rare instances in points of interest less frequently.

## Min Spawn Offset X

The minimum offset in X direction from the spawn location that the instance will be spawned.

## Max Spawn Offset X

The maximum offset in X direction from the spawn location that the instance will be spawned.

## Min Spawn Offset Y

The minimum offset in Y direction from the terrain height at the spawn location that the instance will be spawned.

## Max Spawn Offset Y

The maximum offset in Y direction from the terrain height at the spawn location that the instance will be spawned.

## Min Spawn Offset Z

The minimum offset in Z direction from the spawn location that the instance will be spawned.

## Max Spawn Offset Z

The maximum offset in Z direction from the spawn location that the instance will be spawned.

## Rotate to Slope

Rotate the object to conform it to the terrain normal. Allows natural slope following. Great for things like trees to give them a bit more variation in your environment, or fences to have them follow the natural terrain slope.

## Min Rotation Offset X

The minimum rotation offset in X direction from the spawned rotation that the instance will be spawned. Can be used to rotate objects relative to the spawn rotation.

## Max Rotation Offset X

The minimum rotation offset in X direction from the spawned rotation that the instance will be spawned. Can be used to rotate objects relative to the spawn rotation.

### Min Rotation Offset Y

The minimum rotation offset in Y direction from the spawned rotation that the instance will be spawned. Can be used to rotate objects relative to the spawn rotation.

### Max Rotation Offset Y

The minimum rotation offset in Y direction from the spawned rotation that the instance will be spawned. Can be used to rotate objects relative to the spawn rotation.

### Min Rotation Offset Z

The minimum rotation offset in Z direction from the spawned rotation that the instance will be spawned. Can be used to rotate objects relative to the spawn rotation.

### Max Rotation Offset Z

The minimum rotation offset in Z direction from the spawned rotation that the instance will be spawned. Can be used to rotate objects relative to the spawn rotation.

### Use Parent Scale

Scale the object based on the scale information passed from the parent if selected.

### Min Scale

Scale the object from this scale.

### Max Scale

Scale the object to this scale.

### Scale By Distance

Scale the object between minimum and maximum scale based on the distance the object is from the spawn point.

### Local Bounds

Bounds radius of this game object. Can be used to stop other game objects from spawning within this radius.

### Virgin Terrain

Will only spawn this game object on virgin terrain if selected.

### Ext Param

The extension parameter. Can be used to optionally provide custom configuration information through to any spawn rule extensions. Note: adding ‘nograss’ here will cause grass to be blocked within the bounds radius of this object.

## Game Object Prototype DNA

Contains sizing, clear area bounds radius, spread and scale information that is used by the spawner to control how and where, and how large the Game Object instances will be spawned. Width, height are calculated automatically from the prefab, and bounds radius and seed throw are given generic defaults that you can override.

### Width

The width in world units.

### Height

The height in world units.

### Bounds Radius

Radius from centre of object in world units for bounded area checks. Make this larger if you want more free space around your object when it is spawned.

### Seed Throw

The maximum distance a seed can be thrown when a new instance is spawned. Used to control spread area random clustered spawning.

### Min Scale

The minimum scale that this tree can be rendered into the world.

### Max Scale

The maximum scale that this tree can be rendered into the world.

### Rnd Scale Influence

Randomises the scale somewhere between minimum and maximum scale. If not selected then the scale will be proportionally influenced by the locations fitness.

### Ext Param

The extension parameter. Can be used to optionally provide custom configuration information through to any spawn rule extensions. Note: adding ‘nograss’ here will cause grass to be blocked within the bounds radius of this object as a whole.

## Game Object Prototype Spawn Criteria

An array of spawn criteria that are evaluated against the physical terrain and its surrounds in order to determine the fitness of a location.

There can be more than one spawn criteria defined per resource, and each criteria can select for a different type of location. When assessing a location, the fitness used is the one that returns the highest fitness. Treated as an OR function – take the highest value from this criteria OR this criteria etc.

Within a criteria the fitness is treated as an AND function, and what is returned is the lowest fitness. So if for example height AND slope was selected for evaluation, the fitness returned would be the lowest value.

### Name

The criteria name. Purely informational.

### Check Type

The type of terrain check to make. A Point Check will check a single point on the terrain, and a Bounded Area Check will check that point plus an additional radius that is determined by the Bounds Radius in the DNA. Area based checks are good for checking the terrain for larger structures but is substantially slower so use with care.

### Virgin Terrain

When selected, the criteria will only be valid if the terrain was clear of any other objects at this location. A location is determined to be ‘virgin’ when a raycast collision test hits clear terrain. To detect other objects at this location they must have colliders. You can use an invisible collider and this test to stop any resources that require virgin terrain from spawning at that location. Trees will have to have colliders enabled in order to be detected.

### Check Height

Whether or not this location will be checked for height. Heights here are relative to sea level, + is above, 0 is at, and - is below.

#### Min Height

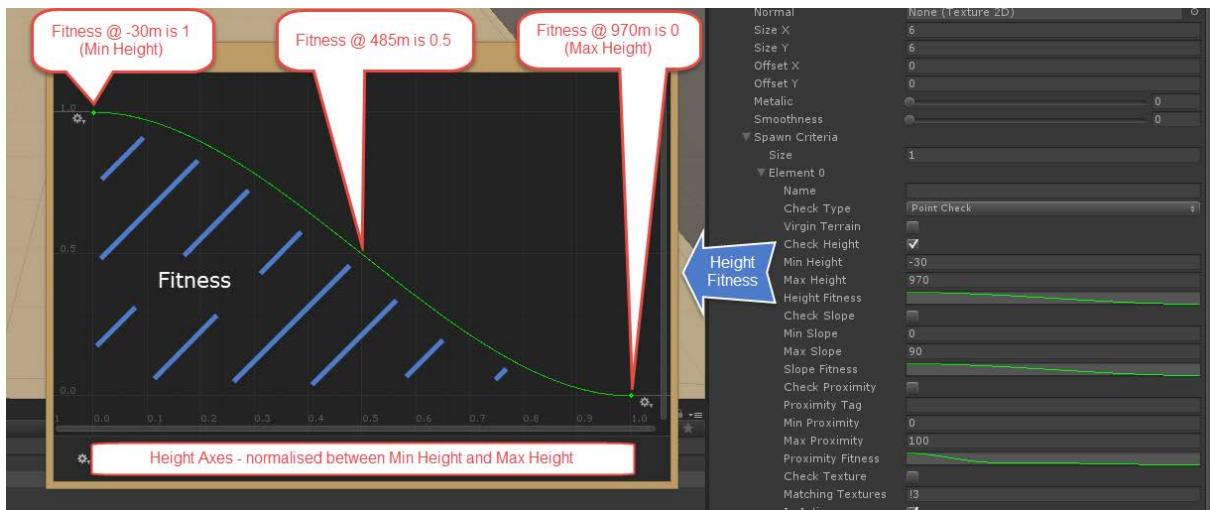
The minimum valid height relative to sea level. Only tested when Check Height is checked.

#### Max Height

The maximum valid height relative to sea level. Only tested when Check Height is checked.

#### Height Fitness

The fitness curve – a curve evaluated between the minimum and the maximum height when Check Height is checked. The green curve represents the fitness over that range.



## Check Slope

Whether or not this location will be checked for slope. Slopes range from 0 degrees which is horizontal, to 90 degrees which is vertical.

### Min Slope

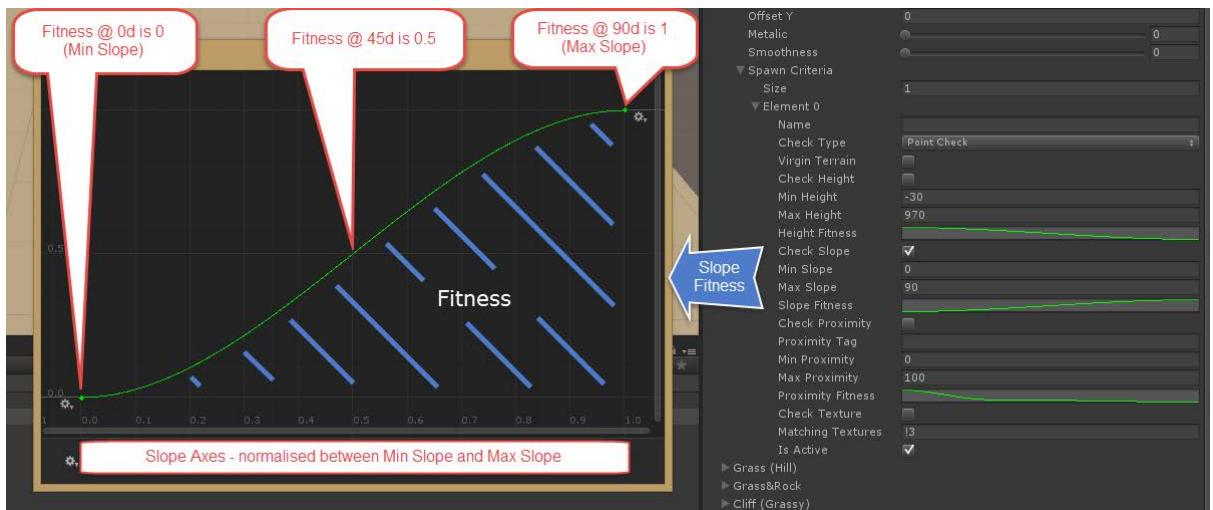
The minimum valid slope. Only tested when Check Slope is checked.

### Max Slope

The maximum valid slope. Only tested when Check Slope is checked.

### Slope Fitness

The fitness curve – a curve evaluated between the minimum and the maximum slope when Check Slope is checked. The green curve represents the fitness over that range.



## Check Proximity

Whether or not to check proximity (in meters) to game objects that have the specified tags.

## Proximity Tag

The Tag that will be checked for. Only tested when Check Proximity is checked.

## Min Proximity

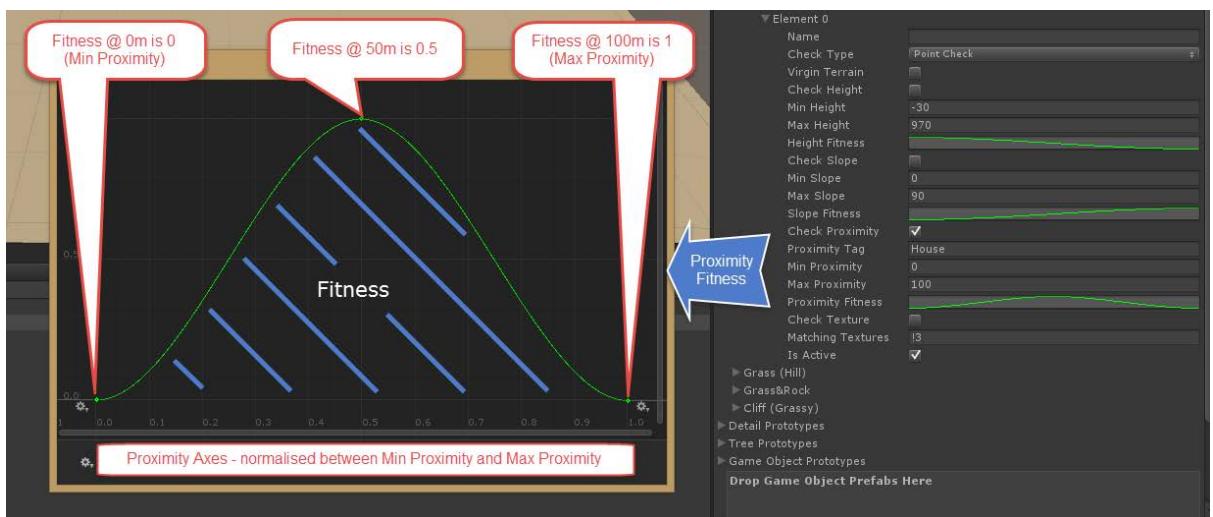
The minimum proximity to the tagged object in meters. Only tested when Check Proximity is checked.

## Max Proximity

The maximum proximity to the tagged object in meters. Only tested when Check Proximity is checked.

## Proximity Fitness

The fitness curve - a curve evaluated between the minimum and the maximum proximity when Check Proximity is checked. The green curve represents the fitness over that range.



## Check Texture

Whether or not to check the texture at this location.

## Matching Textures

Texture slots from your terrain (first valid slot is 0). Will select for presence that texture. Use exclamation mark in front of slot to select for absence of that texture. For example 3 selects for presence of texture 3, !3 checks for absence of texture 3. The fitness returned is the strength of the texture at that location in range 0..1. Only checked when Check Texture is checked.

## Is Active

Whether or not this spawn criteria is active.

## Game Object Prototype Spawn Extensions

An array of spawn rule extension prefabs. Spawn Rule Extensions can be used to extend and modify the way the spawner spawns. Read more about these at the beginning of the resources section of the manual.

## Stamping System

The Stamping system is responsible for terraforming your terrain.

The stamper allows you to take predefined height map based shapes or 'Stamps', and allows you to stamp them into your terrain.

These stamps are generally mountain or hill shaped, but could be any shape you like, for example rocks, house bases or lakes.

Stamps are extensible via the scanning system. You can source new stamps from 3d meshes, Windows 16 bit RAW files (and World Machine R16 files), any image texture such as a PNG & JPG based height maps, and terrains.

The stamper also allows you to visualise your sea level. This is the first and last time you should modify your sea level as it affects all subsequent spawns.

Note: Stampers are transitory objects, so delete and recreate them as needed.



This terrain will be created in the work flow section.

## Work Flow

### 1. Create a Terrain & a Stamper

Open up the Gaia Manager by pressing Control G.

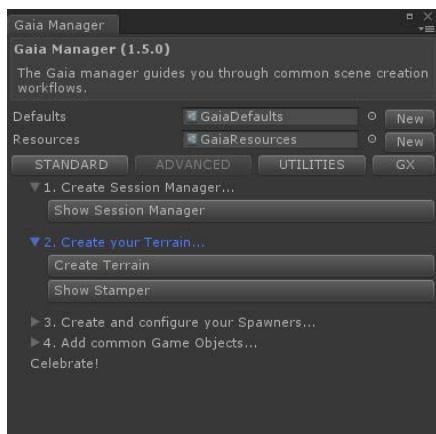
Select your Defaults and Resources files, create a terrain if you don't already have one, and then create a Stamper.

In STANDARD mode:



Create a Terrain and Stamper by clicking the 1. Create Terrain & Show Stamper button.

Or, in ADVANCED mode:



Open the 2. Create your Terrain... foldout

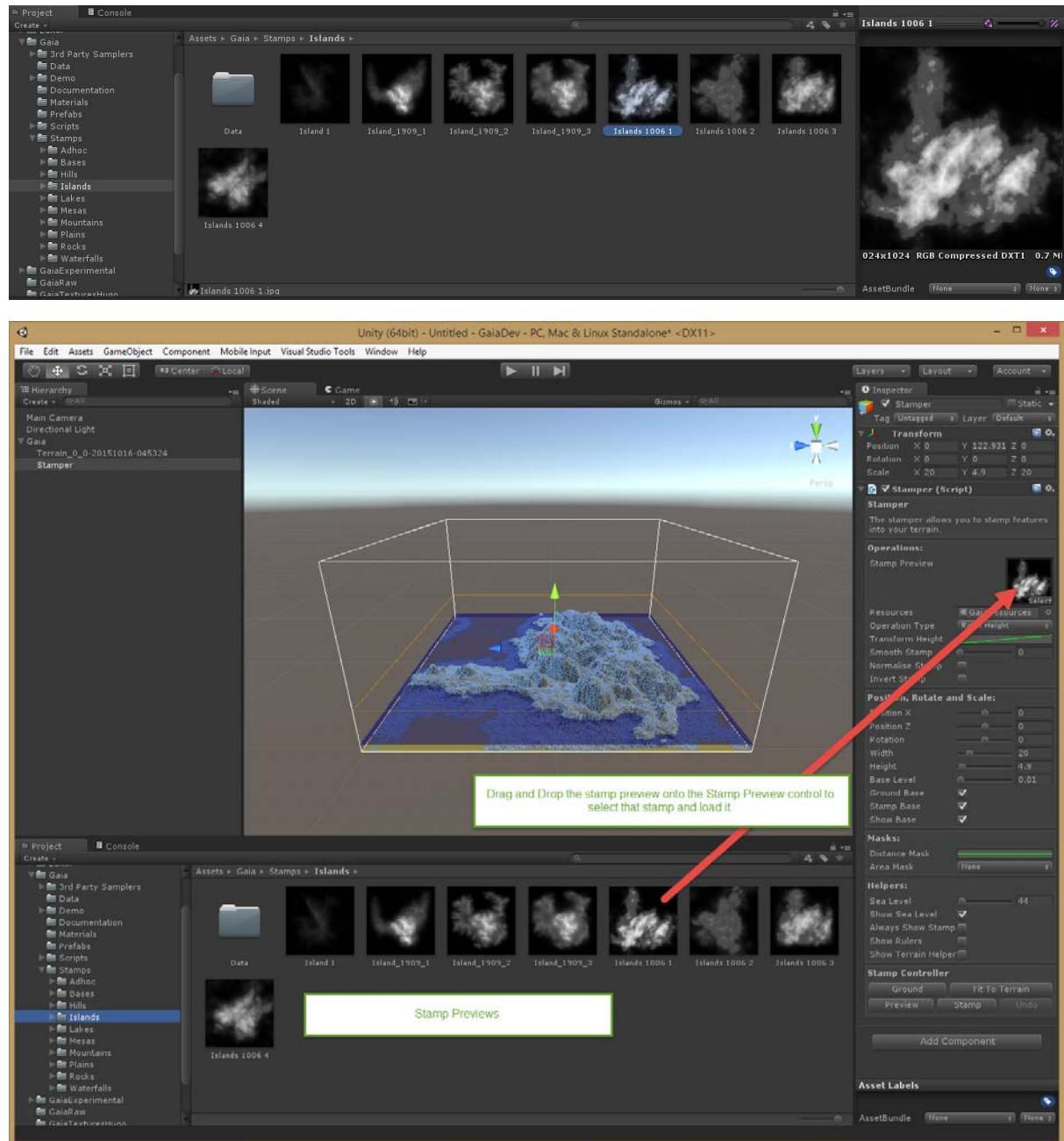
If you have not already created a terrain, click the Create Terrain button.

Click the Show Stamper button.

Gaia will then configure and instantiate a new Stamper, or if there was already one in your scene it will select it.

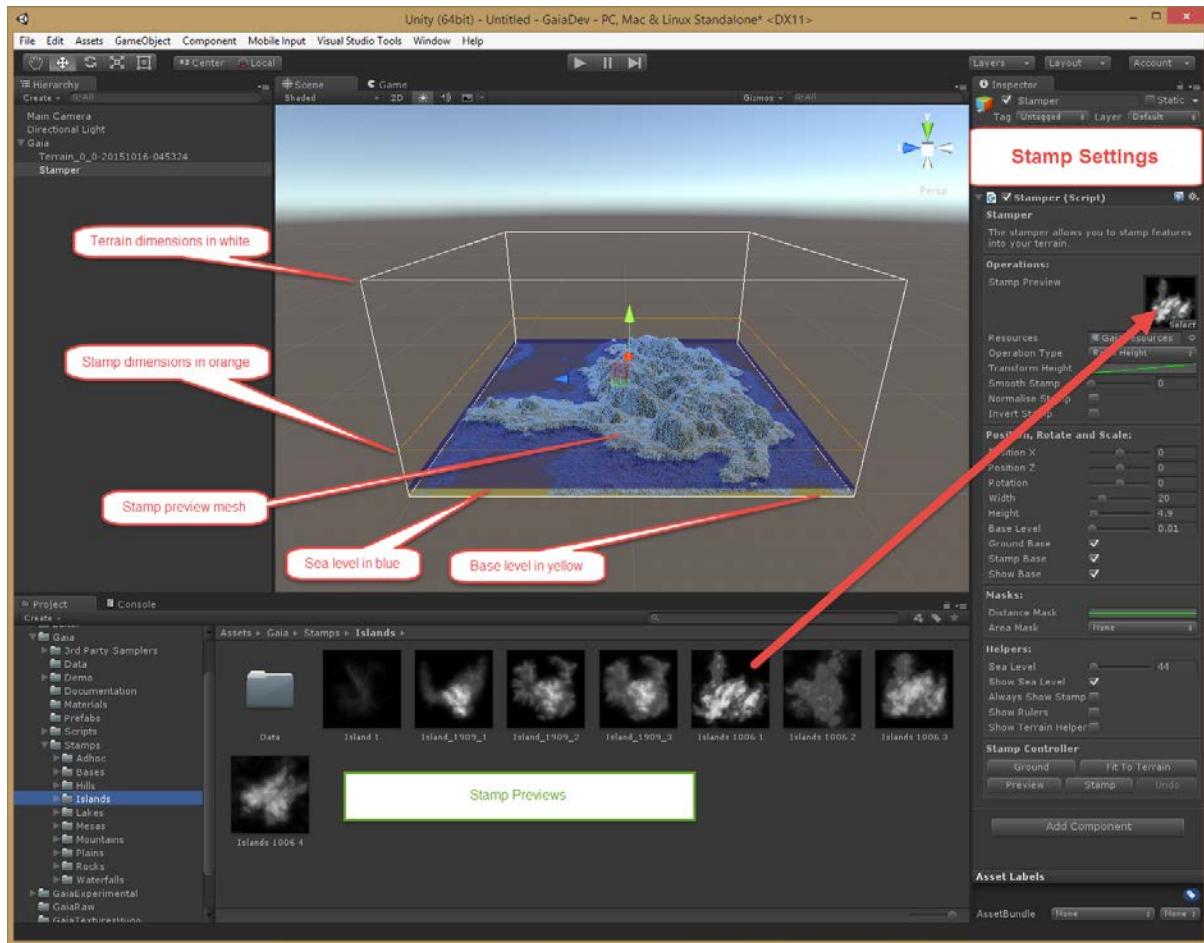
## 2. Select a Stamp

Navigate to the Stamps subdirectory in the Gaia folder, and then browse through the different sub directories to find a stamp. Drag and drop it onto the Stamp Preview control.



Note: The Stamp Preview is a snapshot of the stamp as a height map. White depicts 'high' areas of the stamp, and Black represents low areas of the stamp. These images will give you a sense of what your stamp will look like. The Data directory under the preview directory contains the actual stamp.

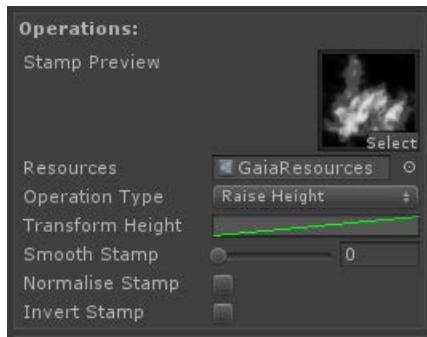
The following image shows the stamper interface after a stamp has been loaded.



The Stamper shows a low resolution preview mesh of the original stamp, as well as the Base level of the Stamp, the Sea Level from the Resources file, the Stamp dimensions in Orange, and the Terrain dimensions in White.

This information is all designed to help you position, rotate, scale and then stamp your feature exactly where you want it on your terrain.

### 3. Choose the Operations you would like to apply to the stamp



The Resources file is the same resources file that is used in your spawners, and was selected by the manager when you created the stamper. Your sea level has been taken from this, and any changes you make to sea level will be stored back here so that your spawners will spawn at the right height.

The *Operation Type* is the type of operation the stamper will use to apply the stamp to the terrain.

**Raise Height** – Will raise the height of the terrain wherever the stamp is higher than the terrain (use this to add mountains, hills and rocks etc).

**Lower Height** – Will lower the height of the terrain wherever the stamp is lower than the terrain (use an inverted mountain and this operation to carve a lake out of the terrain).

**Blend Height** – Will blend the height of the stamp with the height of the terrain. The Slider will control which one is more dominant with left being the terrain, and right being the stamp. For best effect normalise the stamp first.

**Stencil Height** – Will take the high points from the stamp and stencil them into the terrain. The stencil can be either added or subtracted from the terrain. The Stencil height is in meters, and if you normalise the stamp first, will ensure that the highest point on the stamp will be stencilled at that height into the terrain.

**Difference Height** – Will calculate the difference in height between the stamp and the terrain. Can be useful for adding detail to the terrain.

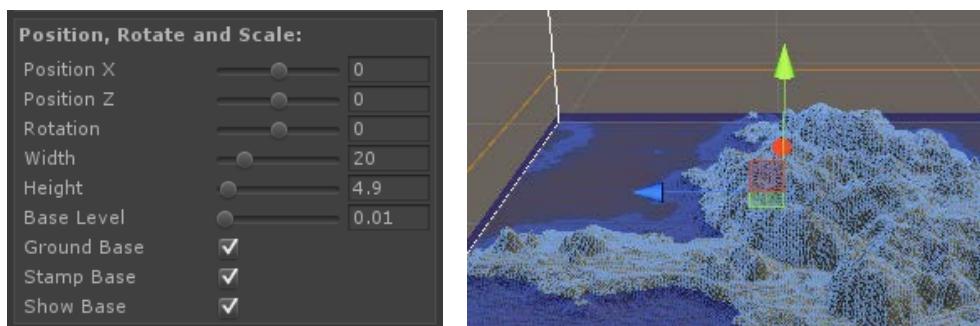
Sometime you will want to modify the height of the stamp prior to stamping it and this is done with the *Transform Height* curve. Heights are represented in the range of zero to one, and this allows you to map the original height values to new height values. By default you do not need to change this.

Depending on where you sourced your stamp from, the stamp could be quite noisy. With the *Smooth Stamp* setting you can pre-smooth the stamp before applying it to the terrain. The smooth will only affect the stamp, not the terrain around the stamp.

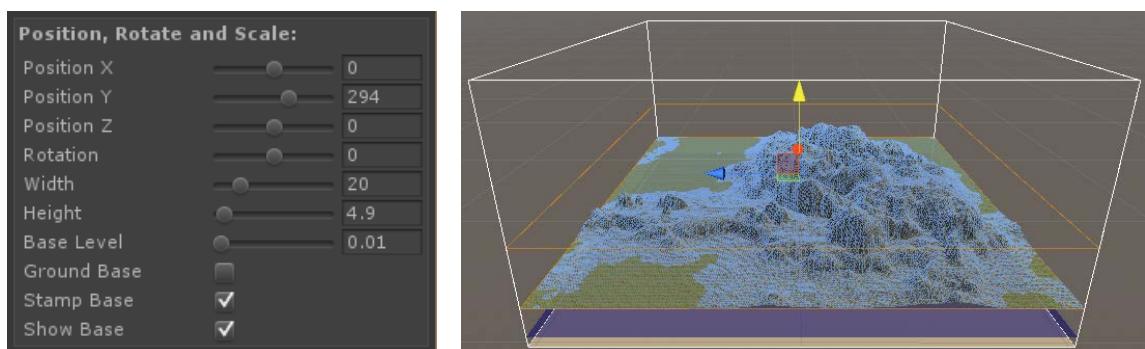
Often a stamp does not use the full range of heights possible between zero and one. Selecting *Normalise Stamp* will scale the stamp heights so that they utilise the full range.

To get a different effect, you can also *Invert Stamp*. This is quite useful when combined with the *Lower Height* operation to allow you to cut features like lakes into the terrain.

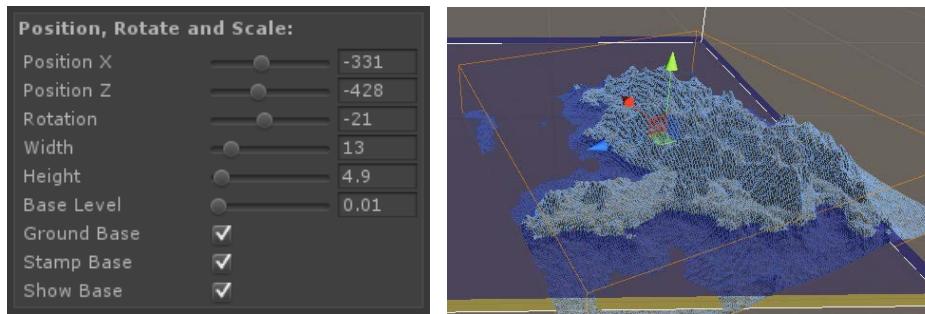
#### 4. Choose the Position, Rotation and Scale of the stamp



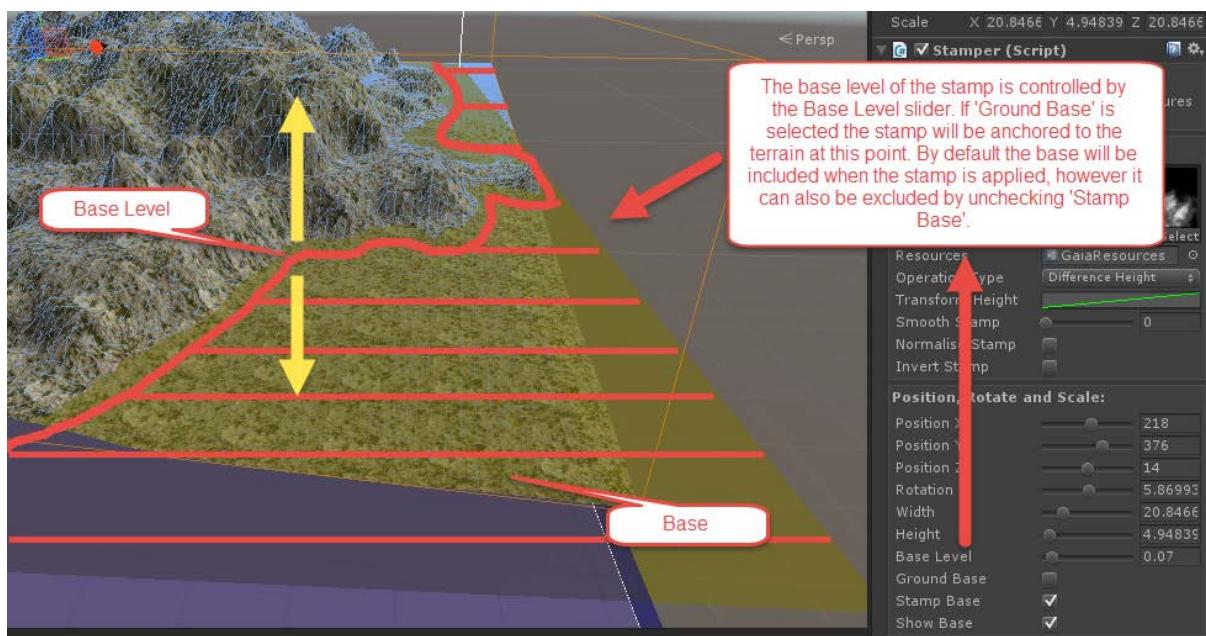
The position can be changed either by dragging the stamp, or by using the sliders. By default the stamp is ‘Grounded to Base’ which means that the base level of the stamp is attached to the terrain base. To enable vertical positioning just un-set the ‘Ground Base’.



In the image above, the stamp has had ‘Ground Base’ unselected, and the stamp has been raised. Notice that the Position Y control has now been exposed as well.



To rotate the stamp, and change its width and height use the sliders or use the standard Unity scale and rotation gizmos. In the example above I re-selected 'Ground Base' for its convenience, and positioned, rotated and scaled the stamp. Notice that the stamp goes off the edge of the terrain. Out of bounds areas will be ignored.

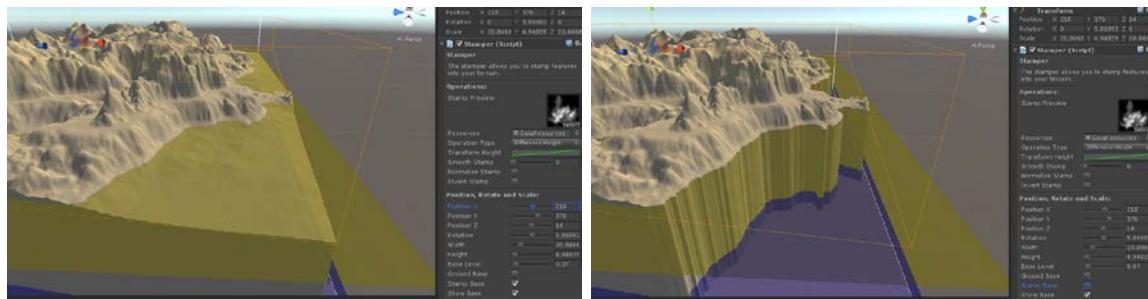


The Base Level setting allows you to raise or lower the Base level of the stamp and is visualised here as the yellow plane. The Base is the portion of the stamp underneath the Base Level.

Selecting Ground Base will stick the base level of the stamp to the ground level of the terrain. You can also ground the base level of the stamp to the ground level of the terrain by clicking the Ground button.

Selecting Stamp Base will ensure that the portion of the stamp under the base level will be stamped, or conversely, unselecting Stamp Base will make sure that it is not stamped. This can be quite useful for removing bases of stamps placed on hills.

The Show Base toggle will show or hide the yellow base guide.



The image above is an example of the effect of stamp base in the final result.

## 5. Select your mask

By default there are no masks assigned to a stamp, which means that the stamp will be stamped at its full resolution, however if you want to you can reduce the strength of the stamp by Masking it.

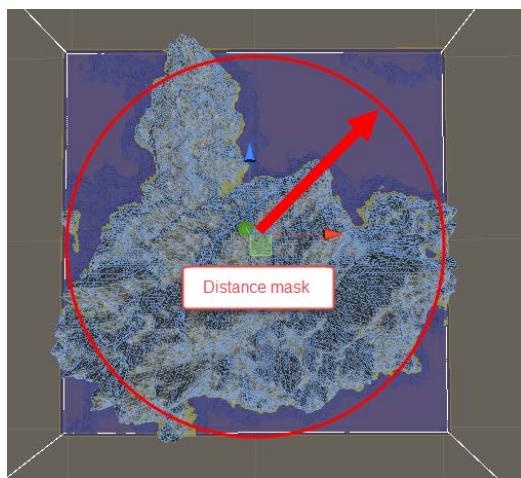
To understand how mask's work lets dig a little into how terrains and stamps work. A Unity terrain is a mesh representation of a height map, and a height map is a normalised array of values that represent heights between zero (low / ground) and one (full terrain height).

The process of stamping works by taking the height of the terrain and the height of the stamp, applying the selected operation to them to calculate a new height, and then applies that new height to the underlying terrain.

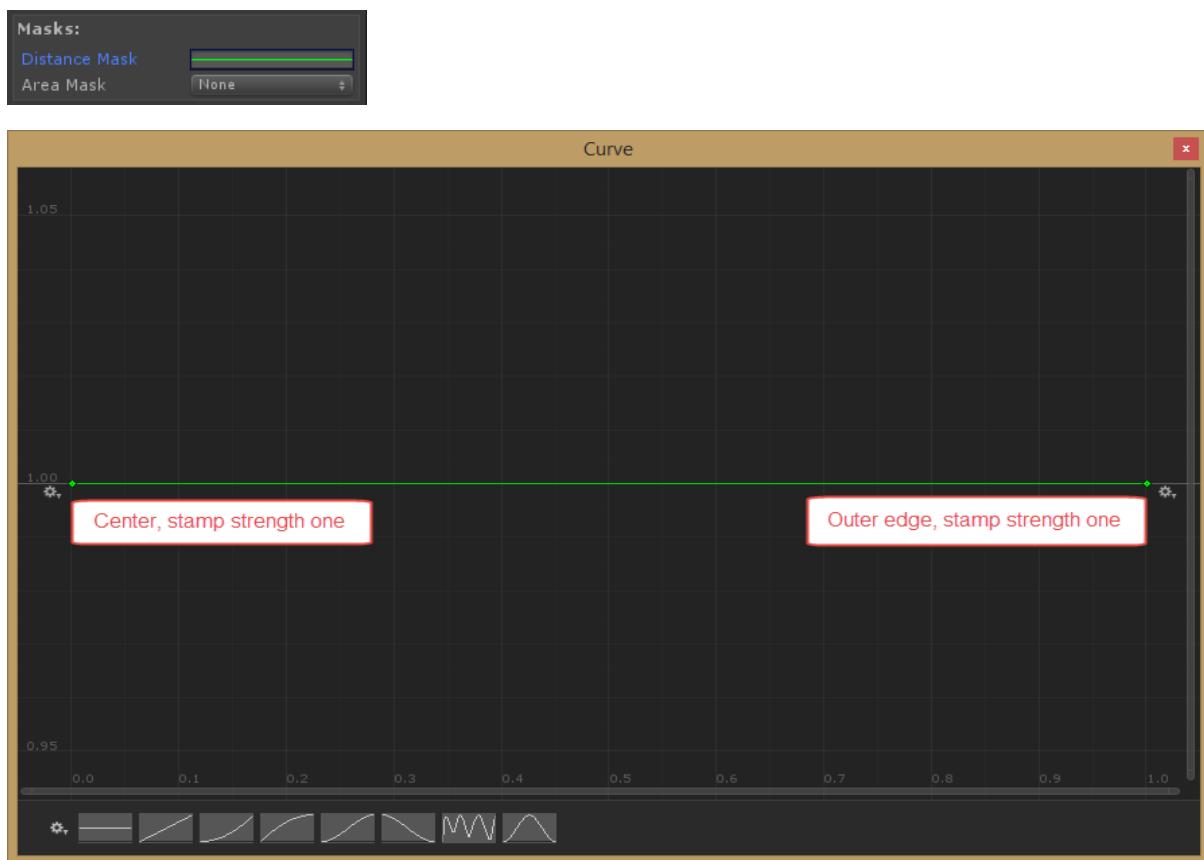
The process of masking takes the difference in terrain height from before and after the stamp operation, and multiplies that by the mask. The new terrain height will now be the old terrain height, plus the newly masked difference.

So if the mask has a value of zero, no change will be made to the terrain, and a mask with a value of one will show the full effect of the stamp.

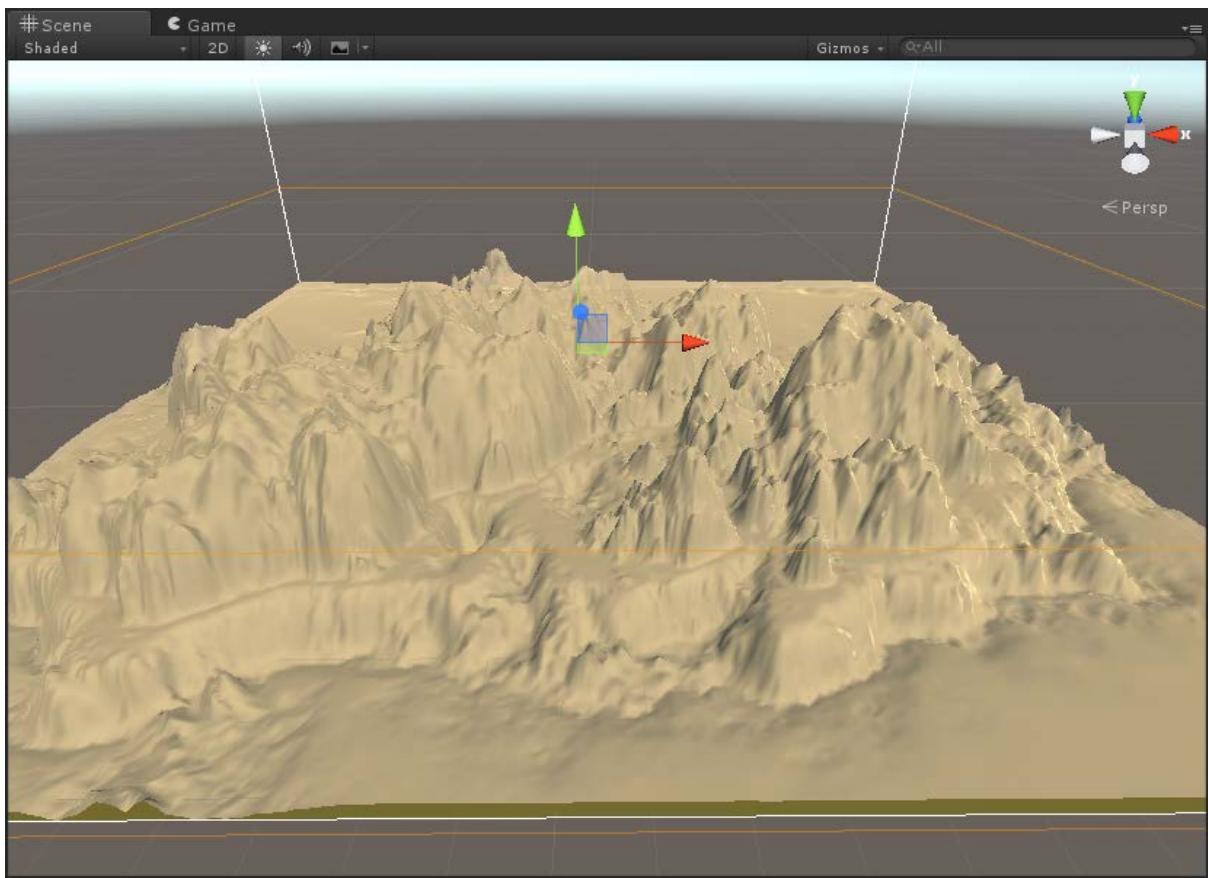
So let's explore the application of a *Distance Mask*.



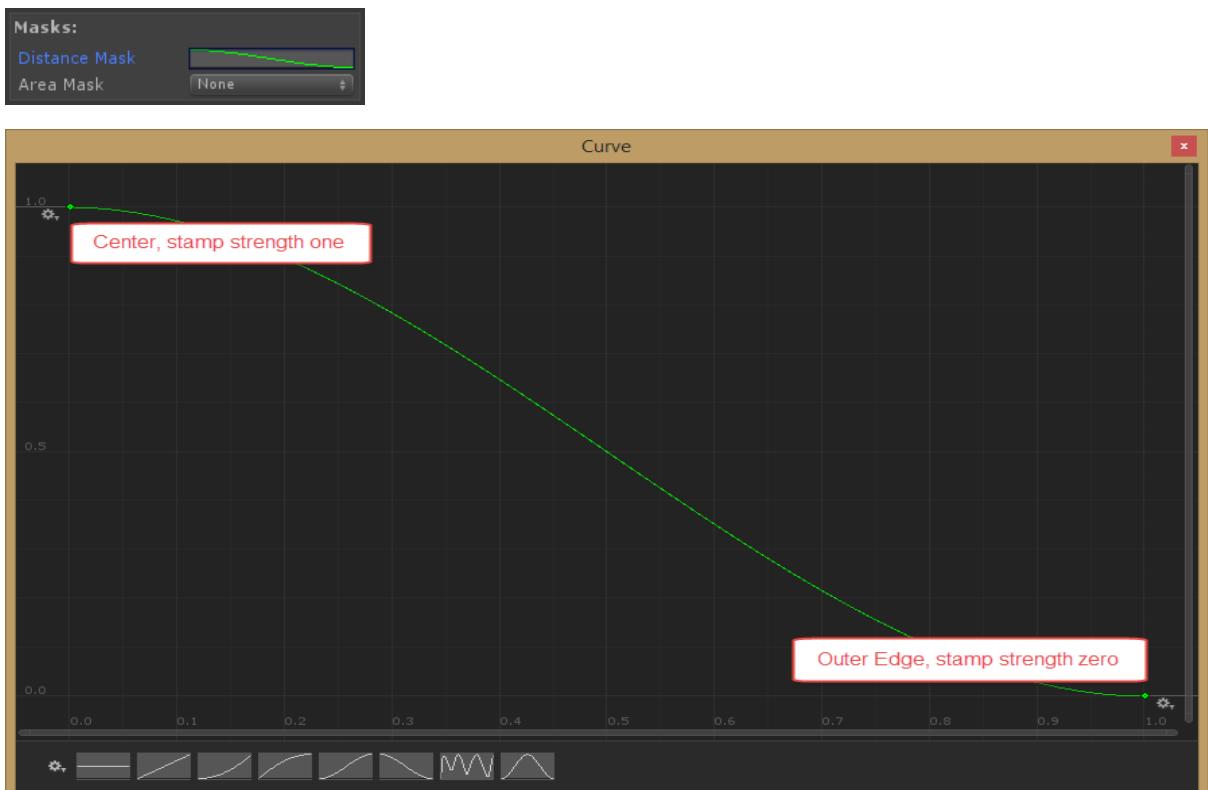
A distance mask is applied from the centre of the stamp, to its furthest edge, in a circular motion.

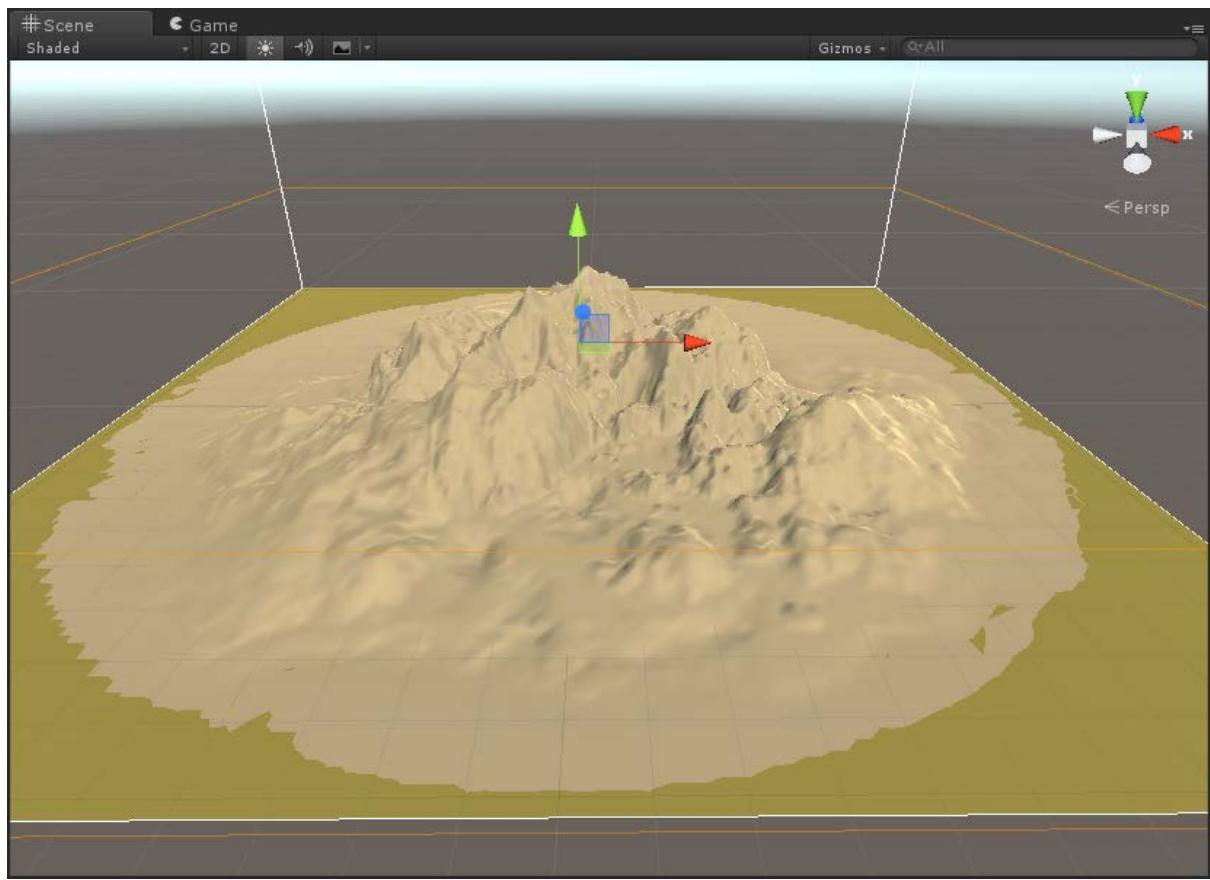


In this example, the mask is applied at a value of one in the center of the stamp, and one at the outer edge of the stamp. In essence this mask will have no impact.

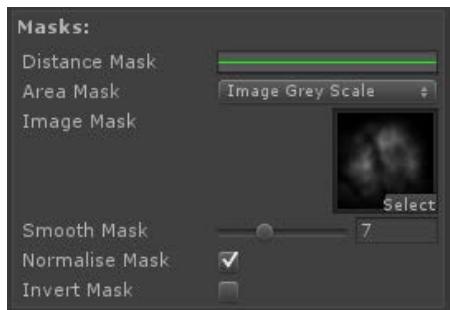


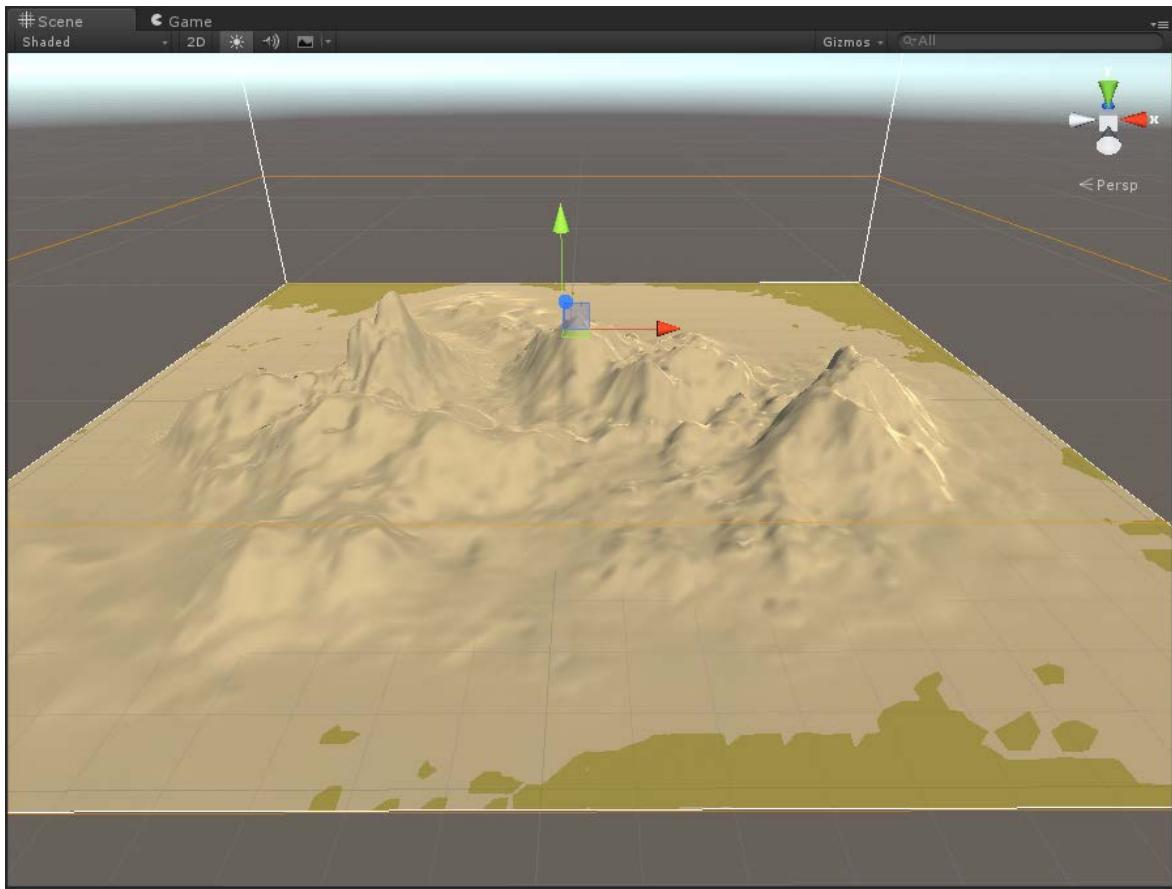
However, let's assume that we do not want the stamp to go all the way to the edge of the terrain, so let's apply a distance mask that tapers the strength of the stamp off towards the outer edge.





When used carefully this is a great technique, the circular result is not particularly organic, so to get a more organic result let's use an *Image Mask* instead.





This result was obtained by using the preview image of one of the mountains as the image mask.

When using images as masks, loss of precision, and compression artefacts can often degrade the quality of the mask and introduce visible noise into the end result, so it is often a good idea to smooth it. In the example above the source image was smoothed seven times.

Because the impact of the mask is based on its relative strength, it is also a good idea to normalise the mask first so that it does not artificially limit the height of the stamp.

You can also invert the mask to reverse its area of influence, and Flip the mask to reverse its X, Y coordinates.

Another type of mask, which is similar to an Image Mask is a *Terrain Texture Mask*. You can paint a texture on the terrain and use that to influence where the stamper will stamp in much the same way as an image mask.

The final type of mask is a Noise Mask in which you can configure Noise parameters and use the generated noise to influence where the stamper will stamp. This also works in much the same way as the image mask.

## *6. Stamp your Stamp!*

To apply the stamp press the Stamp Button.

Often the preview will be in the way so you can toggle its visibility with the Preview button.

If you like what you see then keep it, otherwise hit the Undo button. You can re-apply it again by hitting the Redo button.

The stamper also has some helper functions that can be shown by checking Show Terrain Helper. The terrain helper provides some handy terrain wide helpers such as clearing trees and details, as well as flattening and smoothing the terrain.

Applying one or two smooths to a terrain can have a positive effect on the game performance of the terrain as well.

## *7. Celebrate!*

The following image shows the same terrain fully textured and planted. By changing the position, rotation, scaling of a stamp it is possible to get a multitude of different results from one stamp.



## Stamper Settings

For an overview on how to use the Stamper check out the Work Flow in the previous section.

### Operations

The operations section controls the high level stamper settings and operations.

#### Stamp Preview

The stamp preview. Drag a stamp preview from a stamp sub directory onto this to load and preview a stamp.

#### Resources

The Resources file is the same resources file that is used in your spawners, and was selected by the manager when you created the stamper. Your sea level has been taken from this, and any changes you make to sea level will be stored back here so that your spawners will spawn at the right height.

#### Operation Type

This is the type of operation the stamper will use to apply the stamp to the terrain.

*Raise Height* - Will raise the height of the terrain wherever the stamp is higher than the terrain (use this to add mountains, hills and rocks etc).

*Lower Height* - Will lower the height of the terrain wherever the stamp is lower than the terrain (use an inverted mountain and this operation to carve a lake out of the terrain).

*Blend Height* - Will blend the height of the stamp with the height of the terrain. The Slider will control which one is more dominant with left being the terrain, and right being the stamp. For best effect normalise the stamp first.

*Stencil Height* - Will take the high points from the stamp and stencil them into the terrain. The stencil can be either added or subtracted from the terrain. The Stencil height is in meters, and if you normalise the stamp first, will ensure that the highest point on the stamp will be stencilled at that height into the terrain.

*Difference Height* - Will calculate the difference in height between the stamp and the terrain. Can be useful for adding detail to the terrain.

#### Transform Height

Allows you to modify the heights in the stamp prior to stamping it. Heights are represented in the range of zero to one, and this allows you to map the original height values to new height values. Use only when needed.

### *Smooth Stamp*

Depending on where you sourced your stamp from, the stamp could be quite noisy. This setting allows you to set the number of smooth iterations to apply to the stamp before applying it to the terrain. Only the stamp will be smoothed, not the surrounding terrain.

### *Normalise Stamp*

Often a stamp does not use the full range of heights possible between zero and one. Normalise Stamp will scale the stamp heights so that they utilise the full range.

### *Invert Stamp*

Invert stamp makes high areas low, and low areas high. This is quite useful when combined with the Lower Height operation to allow you to cut features like lakes into the terrain.

## **Position, Rotate and Scale**

Allows you to position, rotate and scale the stamp. Also controls whether or not any area below the base level of the stamp base will be stamped. The standard unity positioning gizmos can also be used to do this.

### *Position X*

X location of stamp centre.

### *Position Y*

Y location of stamp centre. Only shown when Ground Base is not selected.

### *Position Z*

Z location of stamp centre.

### *Width*

The horizontal scale of the stamp.

### *Height*

The vertical scale of the stamp.

### *Base Level*

The Base Level setting allows you to raise or lower the Base level of the stamp and is visualised with a yellow plane. The Base of the stamp is the portion of the stamp underneath the Base Level.

### *Ground Base*

Selecting Ground Base will stick the base level of the stamp to the ground level of the terrain.

## *Stamp Base*

Selecting Stamp Base will ensure that the portion of the stamp under the base level will be stamped, or conversely, unselecting Stamp Base will make sure that it is not stamped.

## *Show Base*

Show or hide the yellow base level visualisation plane.

## **Masks**

Masks allow you to override or mask the effect of a stamp. See the workflow above for a practical demonstration of them. A mask value of 1 applies the full effect of the stamp, and the effect decreases down to a value of 0, which applies none of the stamp.

### *Distance Mask*

A distance mask is applied from the centre of the stamp, to its furthest edge, in a circular motion.

### *Area Mask*

An area mask is applied across the entire stamp.

None - No area mask will be applied.

Image Grey Scale - the mask is sourced from the grey scale conversion of the image mask.

Image Red Channel - the mask is sourced from the red channel of the image mask.

Image Green Channel - the mask is sourced from the green channel of the image mask.

Image Blue Channel - the mask is sourced from the blue channel of the image mask.

Image Alpha Channel - the mask is sourced from the alpha channel of the image mask.

Terrain Texture 0 - the mask is sourced from texture slot 0 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 1 - the mask is sourced from texture slot 1 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 2 - the mask is sourced from texture slot 2 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 3 - the mask is sourced from texture slot 3 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 4 - the mask is sourced from texture slot 4 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 5 - the mask is sourced from texture slot 5 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 6 - the mask is sourced from texture slot 6 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 7 - the mask is sourced from texture slot 7 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Perlin Noise - the mask is sourced from the output of a Perlin Noise function.

Billow Noise - the mask is sourced from the output of a Billow Noise function.

Ridged Noise - the mask is sourced from the output of a Ridged Noise function.

### *Image Mask*

The image to be used as the source of area mask.

### *Smooth Mask*

The number of smooth iterations to be applied to the mask before using it. Can be used to soften noisy masks, and to soften mask edges.

### *Normalise Mask*

The mask will be normalised to use the full dynamic range of the mask.

### *Invert Mask*

The mask will be inverted to reverse its area of effect.

### *Flip Mask*

The mask will be flipped on its x & y axes before applying it. Useful sometimes to match it with the unity terrain which internally is also flipped.

### *Noise Seed*

The seed passed through to the noise function.

### *Octaves*

The amount of detail in the noise. More octaves mean more detail and longer calculation times.

### *Persistence*

The roughness of the noise. Controls how quickly amplitudes diminish for successive octaves. 0..1.

### *Frequency*

The frequency of the first octave.

### *Lacunarity*

The frequency multiplier between successive octaves. Experiment between 1.5 - 3.5.

### *Zoom*

The zoom level of the noise. Larger zooms display the noise over larger areas.

## Helpers

Handy helpers to make working with the stamper and terrain easier.

### *Sea Level*

The sea level used in the scene. This affects all spawning, so it should only be changed when necessary, and with the awareness that it will impact where everything else gets placed within the scene. Changes to this will be reflected in the Resources file, and all spawners will need to be re-spawned.

**Sea Level** is session specific and influenced by the resources file that was used to create the session.

### *Show Sea Level*

Show the current sea level – useful to visualise what will be above and below water.

### *Always Show Stamper*

When checked the stamper will always show in the scene, even when it is not selected. When not checked the stamper will only show in the scene when it has been selected.

### *Show Rulers*

Show some handy rulers to help visualise how big the stamp is in world units.

## Show Terrain Helper

Shows the Terrain Helper buttons.

### *Flatten*

Flattens all terrains in the scene - not reversible so treat with care.

### *Smooth*

Smooths all terrains in the scene - not reversible so treat with care. It is generally good practice to run a single smooth iteration over your scene as the last step - it can have a remarkably positive impact on frame rate.

### *Clear Trees*

Clears terrain trees in all terrains in the scene - not reversible to treat with care.

### *Clear Details*

Clears terrain details (grass) in all terrains in the scene - not reversible so treat with care.

## Stamp Controller

Controls the stamper.

### *Ground*

Aligns the stamp base to the base of the terrain.

### *Fit To Terrain*

Makes the stamp the same size as the terrain. This is particularly important for masking.

### *Preview*

Toggles the preview mesh on and off.

### *Stamp*

Stamps the selected stamp into the terrain.

### *Undo*

Undo's the last stamp. Undo only works on the previous stamp in order to conserve memory.

### *Redo*

Re-applies the last stamp after it has been undone.

## Spawning System

The spawning system allows you to procedurally texture, plant and populate your scene.

The spawning system separates the resources that can be applied to the scene from the spawner that applies them, and the process of instantiating a resource into the scene is called spawning.

The resources understand which types of terrain are ‘fit’ for them to spawn in, and the spawner controls where, how, and when those resources can be spawned.

Spawners can be grouped together and applied to different areas of your terrain. This is very powerful as it allows you to control where and when you want to leverage your resources. For example you might have desert biome and a tropical wetlands biome. A biome is an area that has common characteristics.

Because the spawning system is separated from the underlying terrain, and is rules based, you can change the terrain, and then have run the spawners against the new terrain, and they will adapt their output to the new terrain.

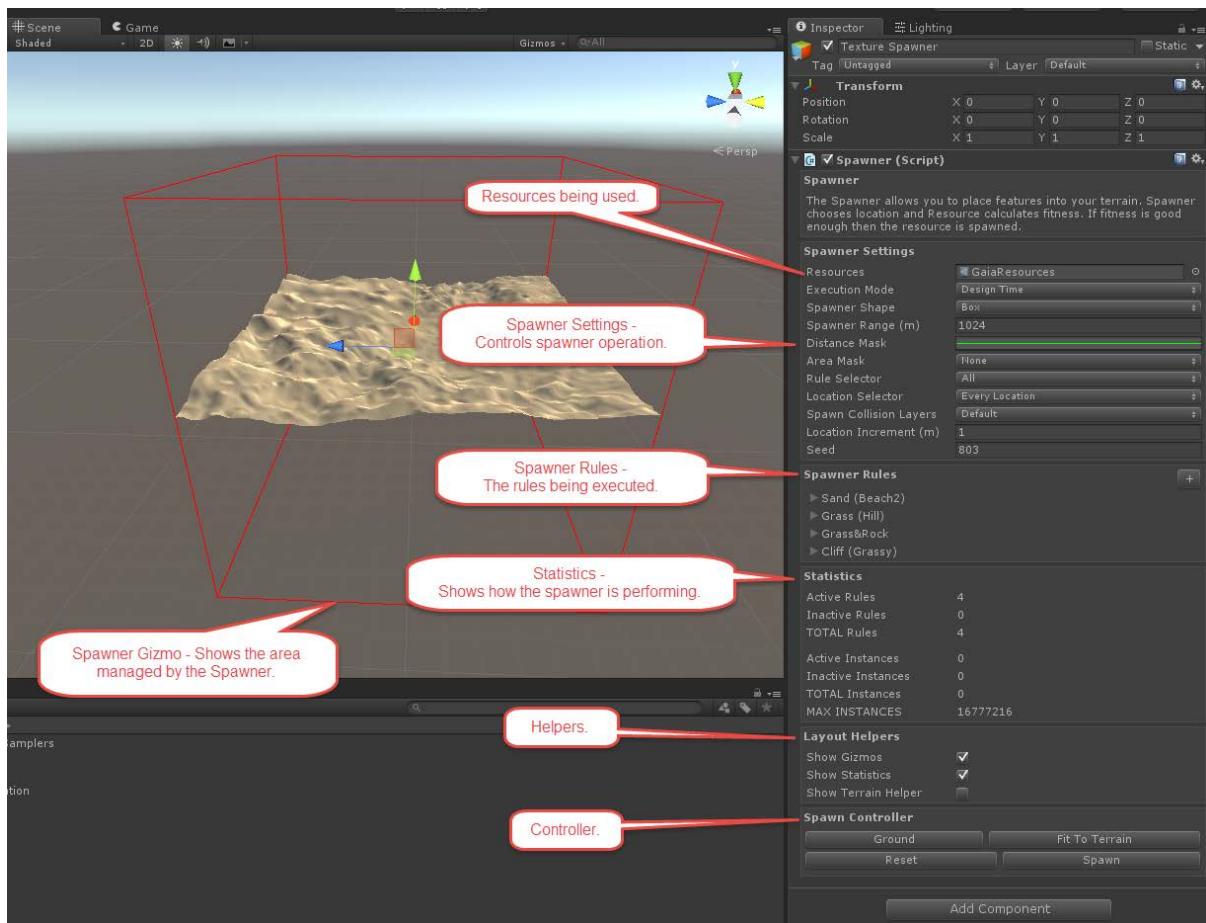
Spawners and spawner groups can be prefabbbed, and because of this you can quickly and easily adapt to new scenes and new environments. In fact, you could prefab an entire game and spawn it into each new scene with the click of a button on a group spawner.

The key pieces of the spawning system are:

- The Resources file – this contains the resources and the criteria through which they can be expressed;
- The Visualiser component – a component that allows individual resource settings to be visualised and modified;
- The Spawner component – a component which chooses where and how resources can be spawned into a scene, and then spawns them;
- The Group Spawner component – a component which allows Spawners to be aggregated and executed as one entity.

The four types of resources that can be spawned are:

- Texture resources to texture the terrain. They are applied directly to the underlying terrain splat maps;
- Detail resources for grass, small bushes and stones. They are applied directly to the terrain detail maps;
- Tree resources to populate the terrain with trees. They are applied directly to the terrain as terrain trees;
- Game Object resources to populate the terrain by instantiating prefabs. They are instantiated into the scene as game objects.



All spawners operate in the same way, and while it is possible to mix different resource types in the same spawner it is often more practical to group similar resource types together and run similar sets of rules against them.

A spawner has a reference to the resources it wants to spawn, settings that control how it operates, a set of rules that it evaluates based on its settings, statistics to show progress, helpers and a controller.

Each execution of the spawner is called an iteration, and is executed either at design time when the spawn button is pressed, or at run time based on the execution mode selected.

Each spawner has its own random number generator with its own seed value and is deterministic i.e. the spawner will use the same set of random numbers in the same sequence to get the same output every time an iteration sequence runs.

Every iteration, the spawner will run its rules against its resources and if the rule is satisfied will spawn those resources into the terrain. Underpinning this the random number generator will provide a series of random numbers that drive the process in a repeatable way.

Hitting the reset button or calling the reset method will reset the spawner's random number generator back to its seed value and future iterations on that spawner will execute the same sequence again.

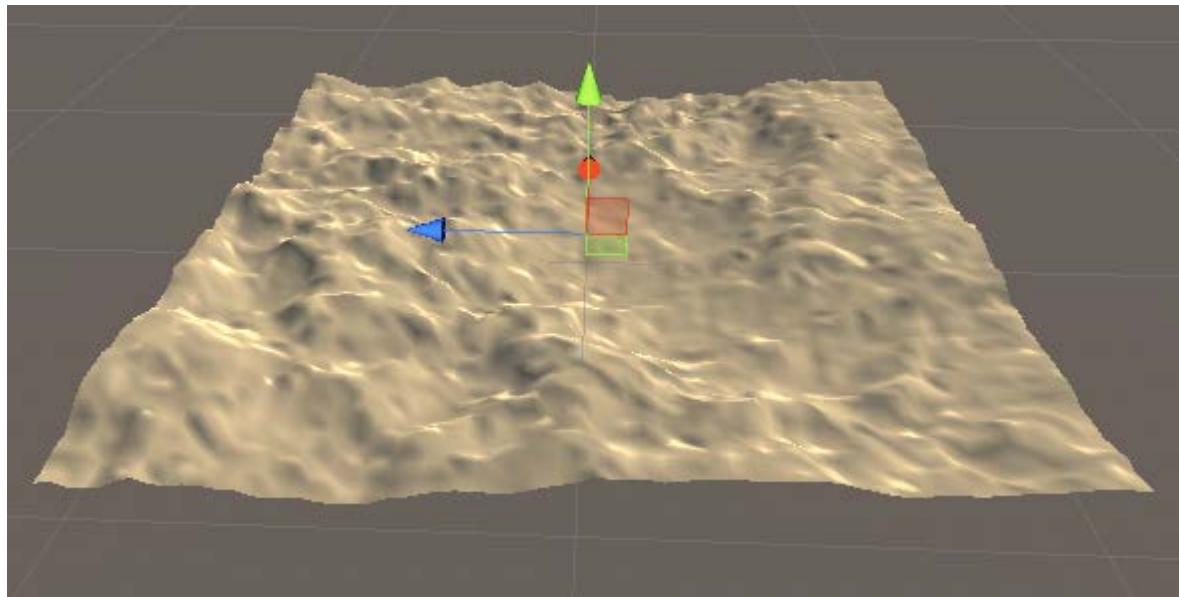
The effect of this is that a spawner will generate the same outcome every time, and will always do so, except for when the underlying environment has changed. If the underlying environment changes, even though the same set of random numbers is driving the process, the resulting output will most likely not be the same.

While spawners are typically designed to be used in the editor, but they can also be used at runtime.

## Work Flow

### 1. Create a new Scene & Stamp a Terrain

See the Stamper Workflow for guidance on how to do this.



### 2. Create your Spawners

Open up the Gaia Manager by pressing Control G.

Select your Defaults and Resources files.

In STANDARD mode:

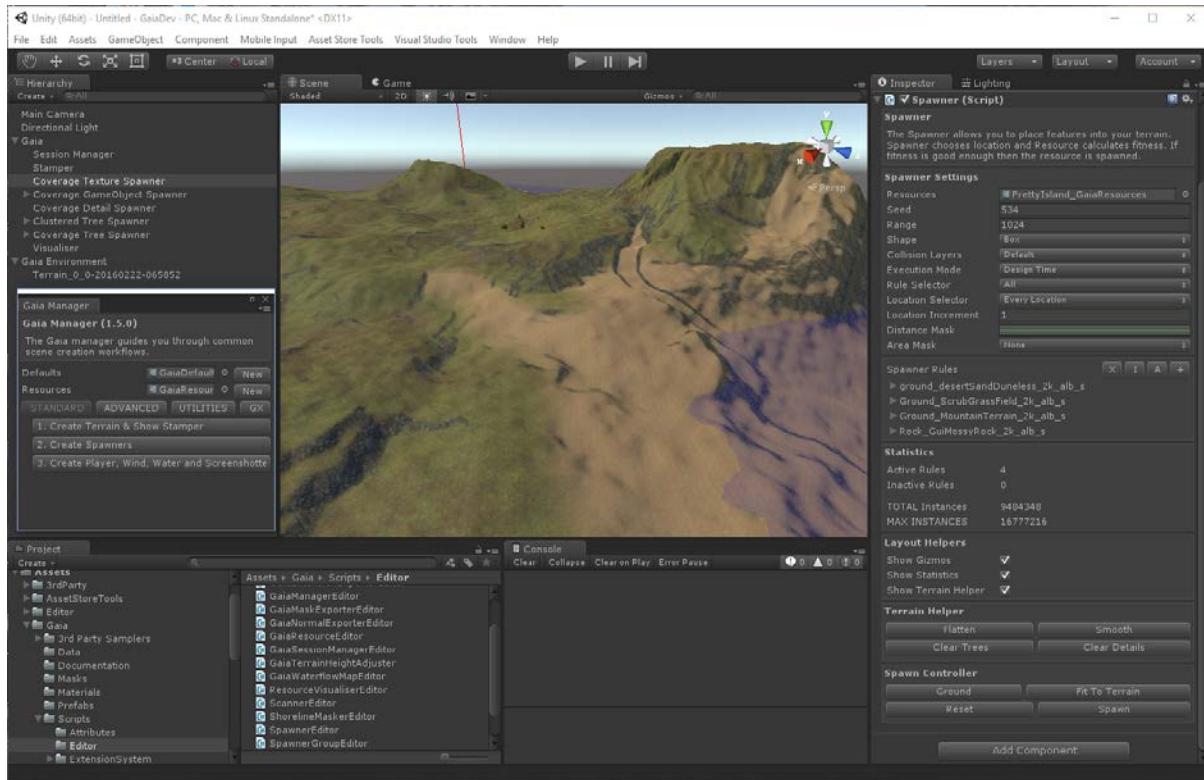


Create your spawners by clicking the 2. Create Spawners button.

Gaia will create a default set of all Spawners and a Group Spawner.

### 3. Texture your terrain with the Coverage Texture Spawner

Select your Coverage Texture Spawner and click Spawn.

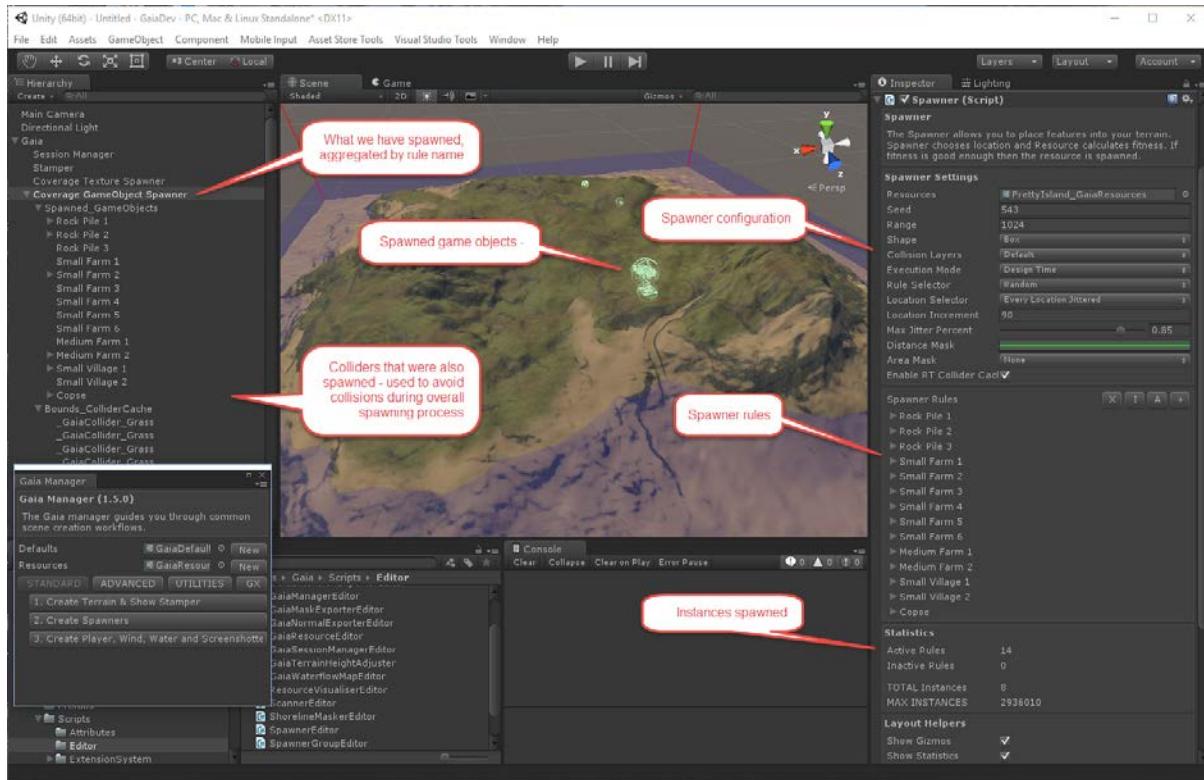


Gaia will now texture the terrain. At the completion of this process it will also show you some statistics on how many instances of the textures were applied to the terrain.

For more details on Texture Spawning see the Texture Spawner section.

#### 4. Place your Game Objects with the Coverage Game Object Spawner

Select your Coverage Game Object Spawner and click Spawn.

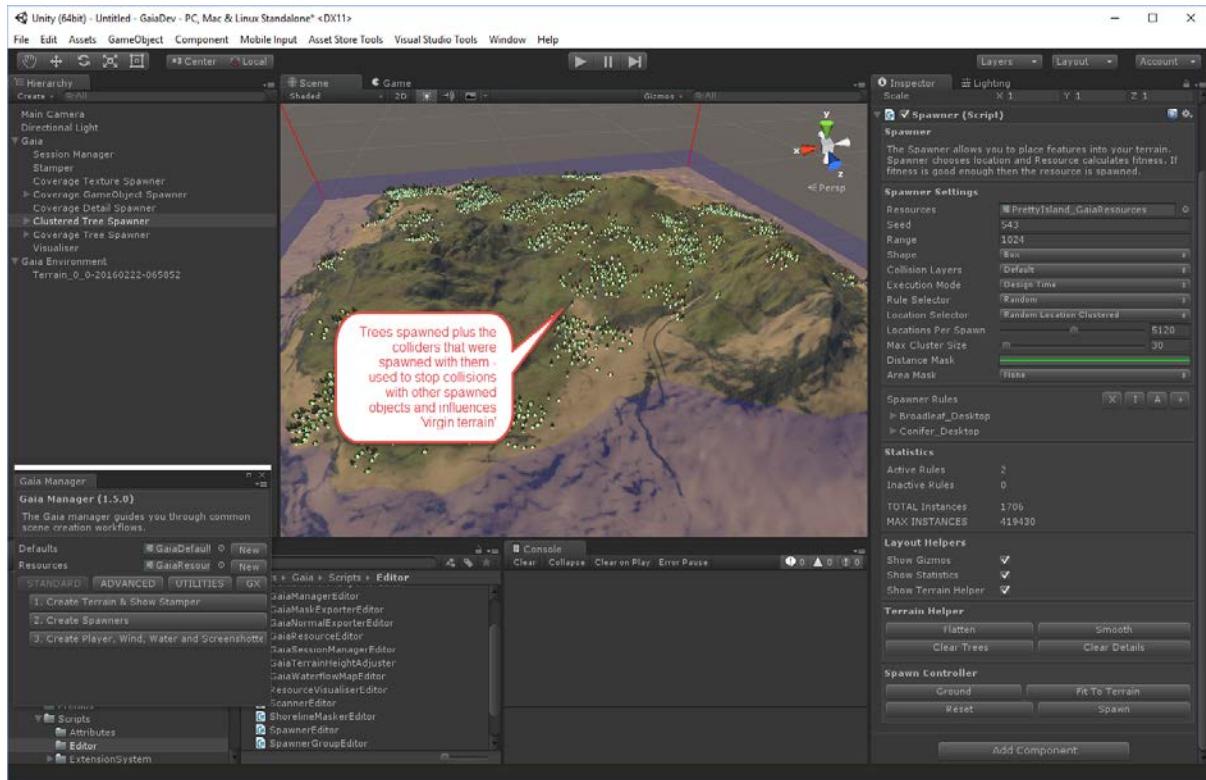


Gaia will now instantiate and place your game prefabs into the terrain. At the completion of this process it will also show you some statistics on how many instances Game Object were applied to the terrain.

For more details on Game Object Spawning see the Game Object Spawner section.

## 5. Place your Trees with the Tree Spawner

Select your Tree Spawner and click Spawn.

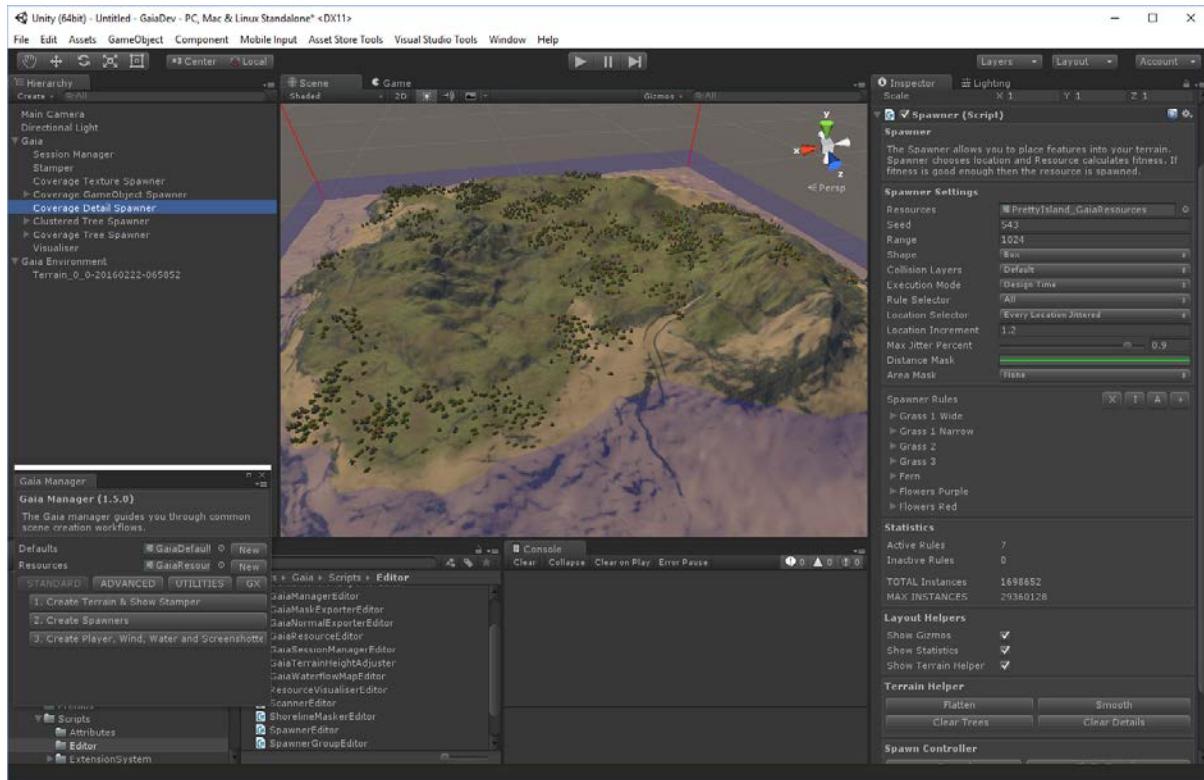


Gaia will now instantiate and place your trees into the terrain. At the completion of this process it will also show you some statistics on how many tree instances were applied to the terrain.

For more details on Tree Spawning see the Tree Spawner section.

## 6. Place your Details (grass) with the Detail Spawner

Select your Detail Spawner and click Spawn.

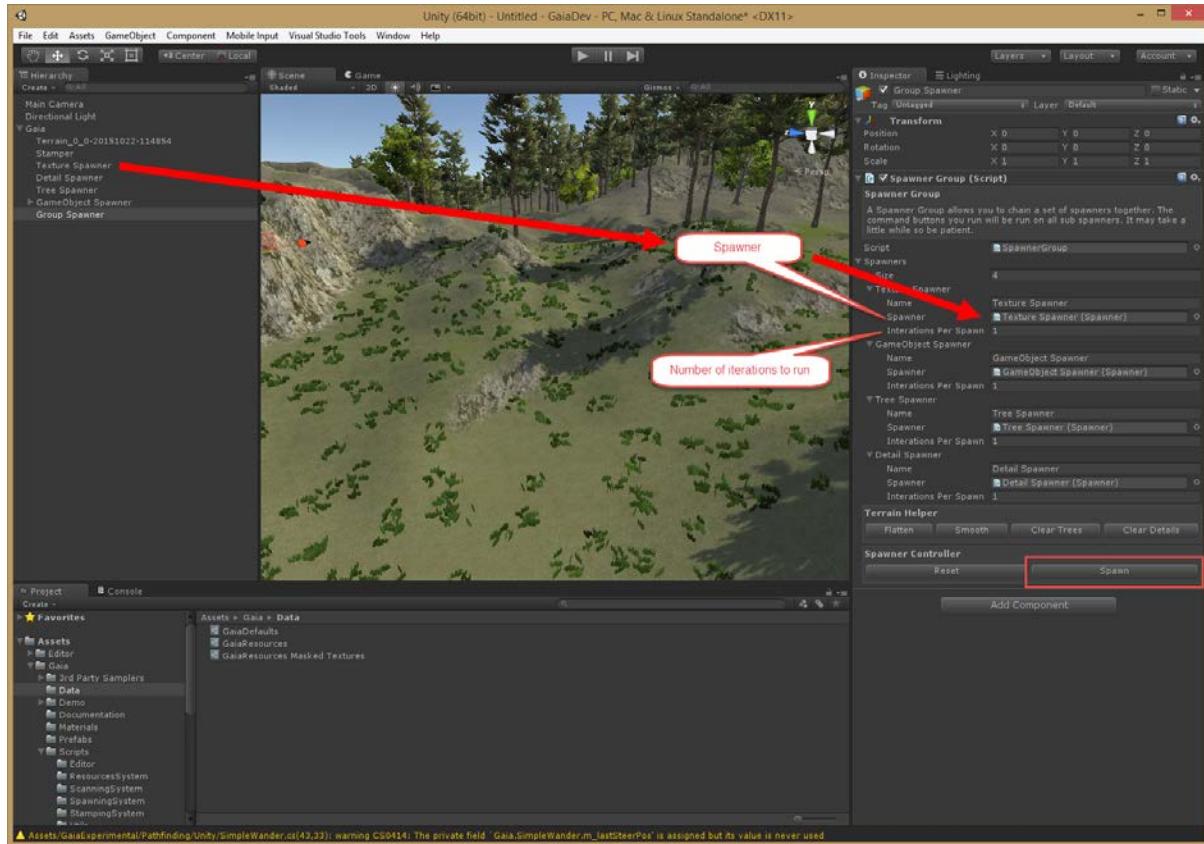


Gaia will now instantiate and place these details into the terrain. At the completion of this process it will also show you some statistics on how many detail instances were applied to the terrain.

For more details on Detail Spawning see the Detail Spawner section.

## 7. As an Alternative to Steps 3, 4, 5 and 6.

Select your Group Spawner and click Spawn.



When you hit Spawn, Gaia will run through each spawner in order and execute the number of iterations defined. Each spawn iteration is the equivalent pressing the Spawn button once.

Conversely, hitting the Reset button will have the same effect of hitting the Reset button once on every spawner defined.

The spawner execution order is important however it is only a suggestion and you can easily change it.

Here is why it is done this way:

- You would typically texture the terrain first so that any subsequent Spawners that have a resource criteria that depends on checking for the right texture is supported.
- Next we lay down your Game Objects – this ensures that there is adequate free space in your environment so that trees do not act as obstacles when checking for clear or virgin terrain.

- Next we lay down the Tree's. By having the game objects already in place, the trees will avoid growing where they have been placed.
- Finally the details or grass is laid down. The reason this is done last is because grass is by default configured only to spawn on virgin or clear terrain. This stops grass from growing up through buildings, trees and rocks.

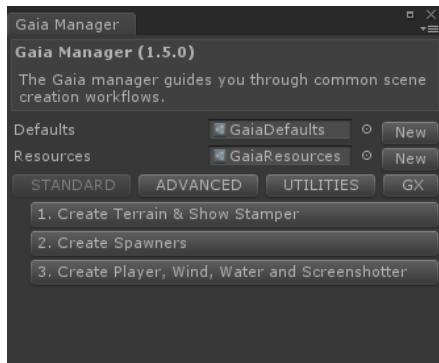
## 8. Add your Player, Wind, Water and Screenshotter

Optionally you can have Gaia configure the rest of the scene for you. You must have installed the relevant Standard Assets for this to work properly.

Open up the Gaia Manager by pressing Control G.

Select your Defaults and Resources files.

In STANDARD mode:



Click 3. Create Player, Wind, Water and Screen Shotter

Gaia will:

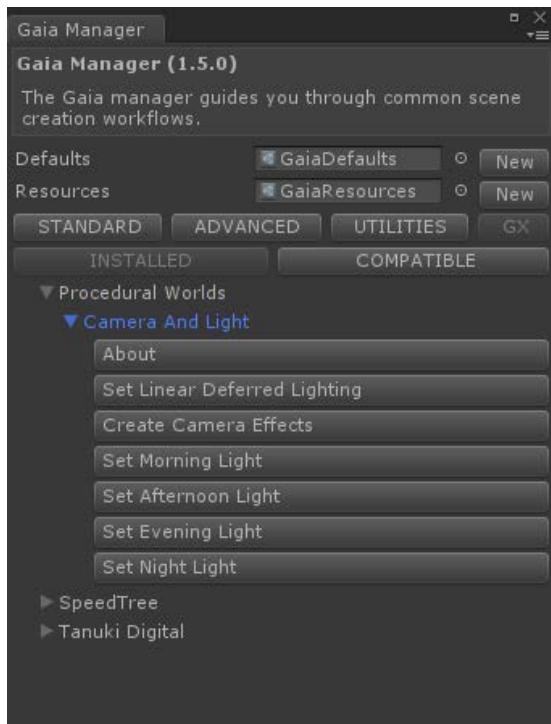
- Create a FPS Controller or the Player Prefab defined in your settings file and place it at ground level in the centre of your scene.
- Create a WindZone and place it in your scene.
- Create a Wayer4Advanced object or the prefab defined in your settings file and place it in your scene.
- Create a Screen Shotter object and place it into your scene.

## 9. Add Lighting and Camera FX

Optionally you can have Gaia configure the rest of the scene for you. You must have installed the relevant Standard Assets for this to work properly.

Open up the Gaia Manager by pressing Control G.

In GX mode:



Click the Set Linear Deferred lighting mode – will set your project into linear deferred lighting mode.

Click Create Camera Effects – will add a set of default unity effects to your camera – insta pretty.

Click Set Afternoon Light – will set up the default sky settings with some pleasing afternoon light.

## *10. Press Play and Celebrate!*

Press the play button and enjoy exploring your scene. To take a screen shot hit the F12 key and when you have finished playing the scene check the screen shot out in the Screenshots folder.



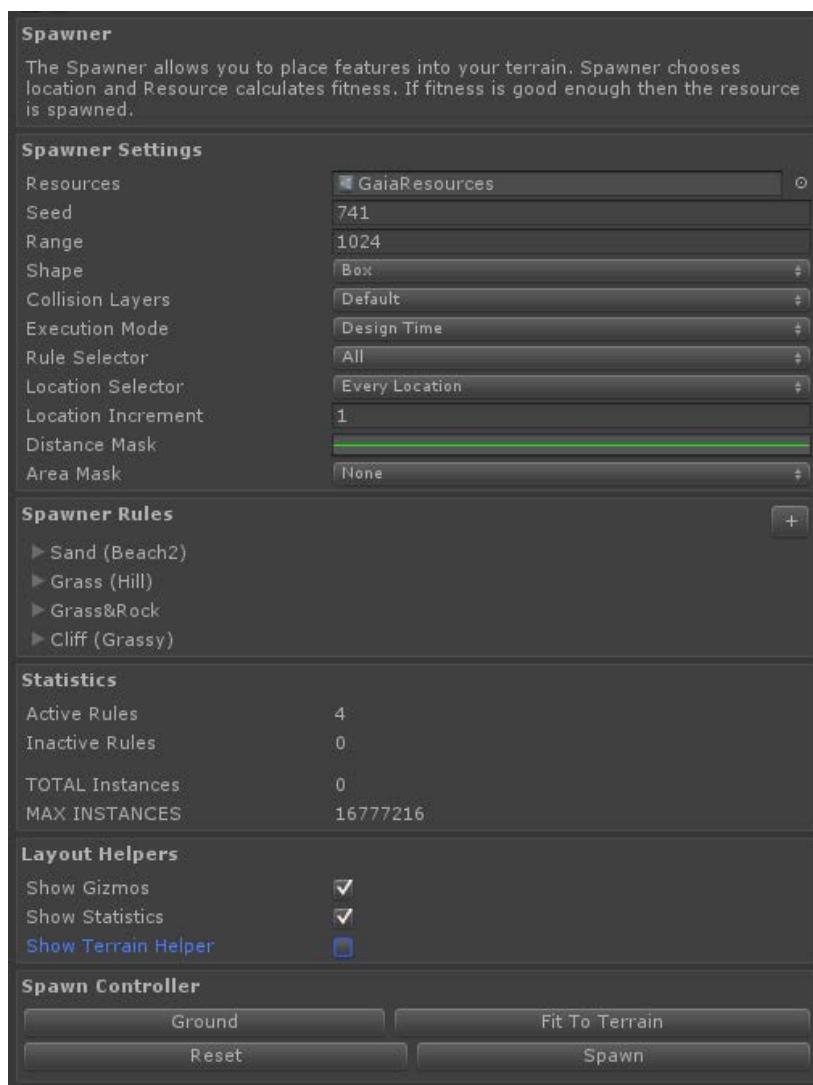
## Texture Spawners

Texture Spawners are designed to texture your terrain.

First make create and stamp a terrain. Then open Gaia Manager and make sure you have the right resources file selected.

Then use Gaia Manager to create a texture spawner. In advanced mode you can create as many texture spawners as you like.

Gaia Manager will create and configure the spawner for you, and add all the texture resources in your resources file into it. By default Gaia has been set up to do a reasonable job with just 4 textures, but there is nothing to stop you from adding more textures.



By default the texture spawner will be fitted to your terrain, however you can easily resize and move it around as you please.

The Location Selector has been set to iterate though every location at 1 meter intervals, however this can sometimes be a little smaller depending on the size of your terrain. The idea is to ensure that all terrain gets its textures applied.

Rule Selector in a texture spawner is set to execute all rules, and there is one rule for every texture that will be applied.

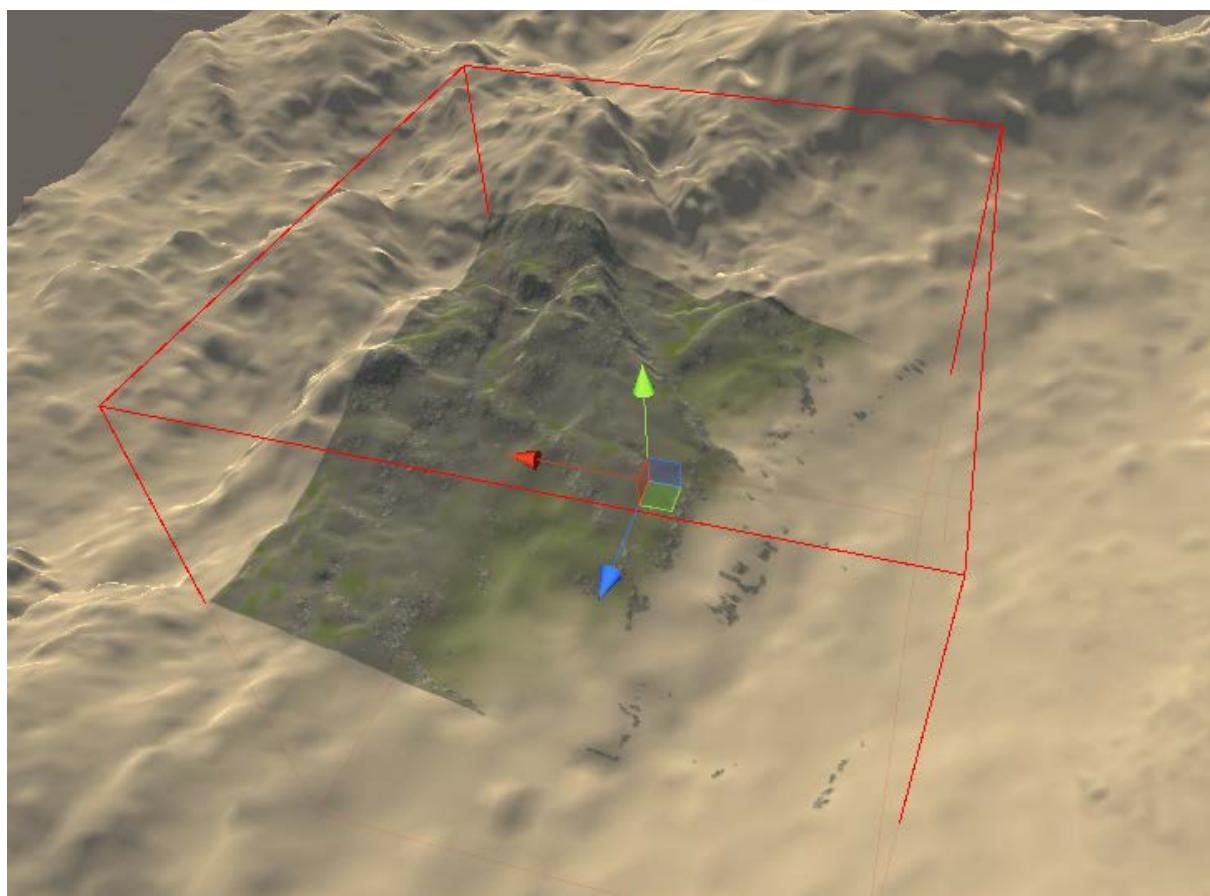
The way a texture spawner works is that it iterates over every location in your terrain, and then iterates through every rule and executes it.

When a rule is executed, the associated resources fitness criteria is evaluated to derive a fitness value for that resource at that location.

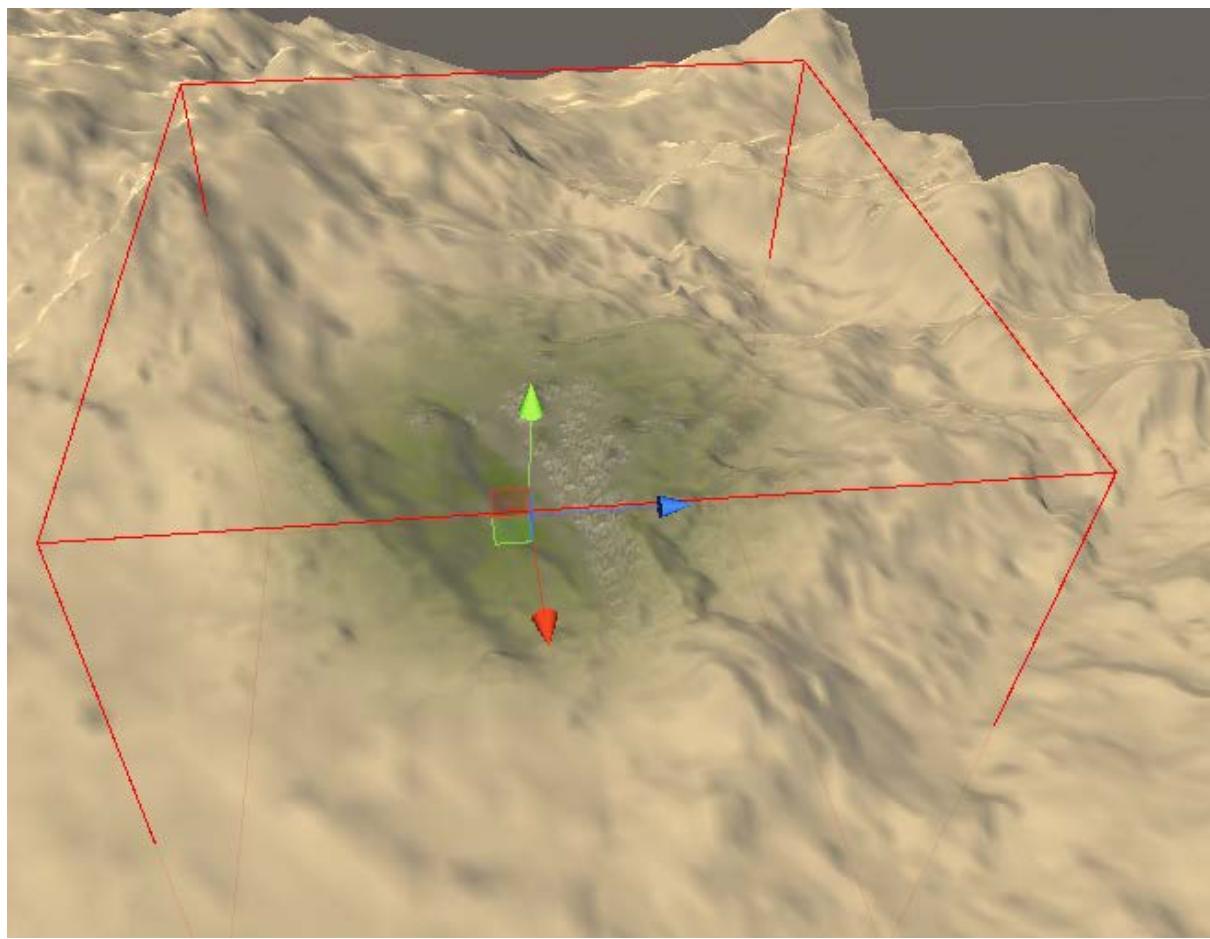
If the resources fitness is greater than its texture strength at a location then its texture strength will be increased to match the fitness, and all other texture strengths will be proportionally decreased i.e. it is painted stronger, and everything other texture is painted weaker.

The key to this is that a texture is only modified if its strength is stronger, otherwise it is ignored.

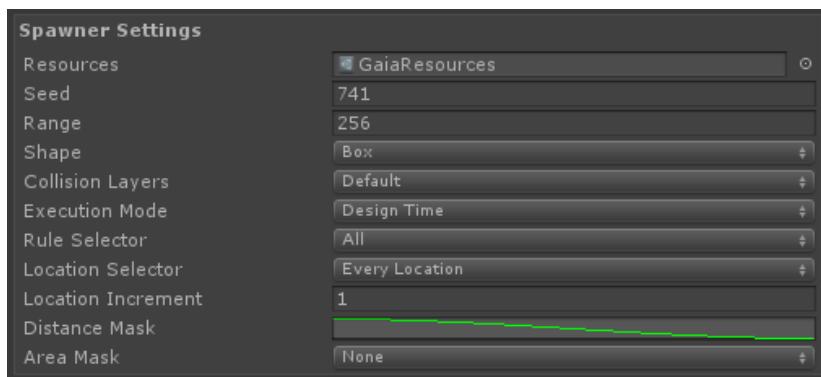
To all intents and purposes, you are essentially painting each new texture on top of the previous texture, and the last strongest texture painted is the one that wins.



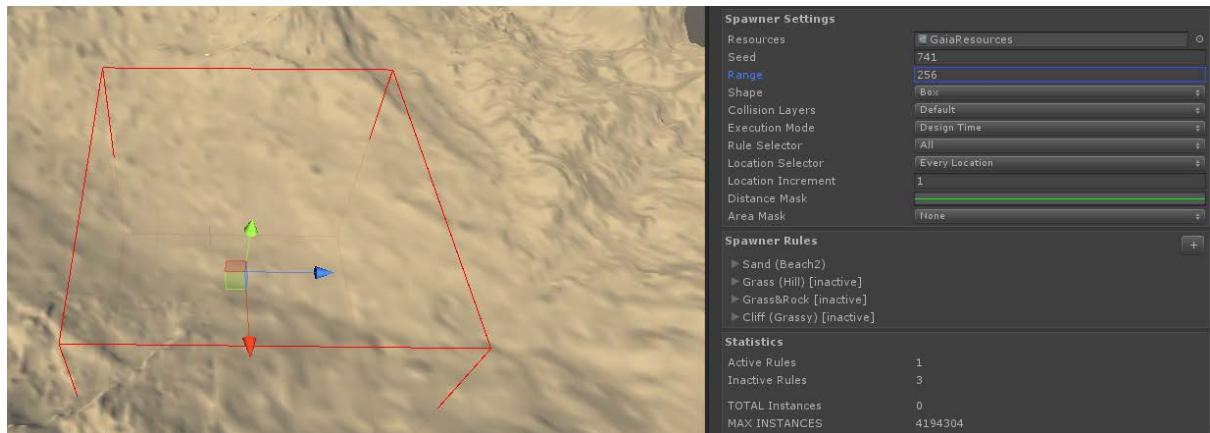
Here is sample texture spawn showing how the bounding box of the spawner can be used to constrain the area of effect.



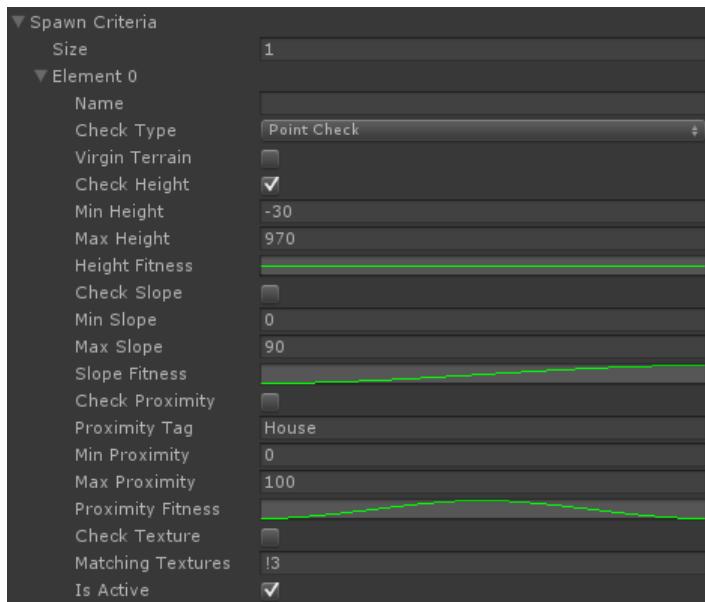
In this spawn, the distance mask was used to constrain the spawn to the centre of the spawner.



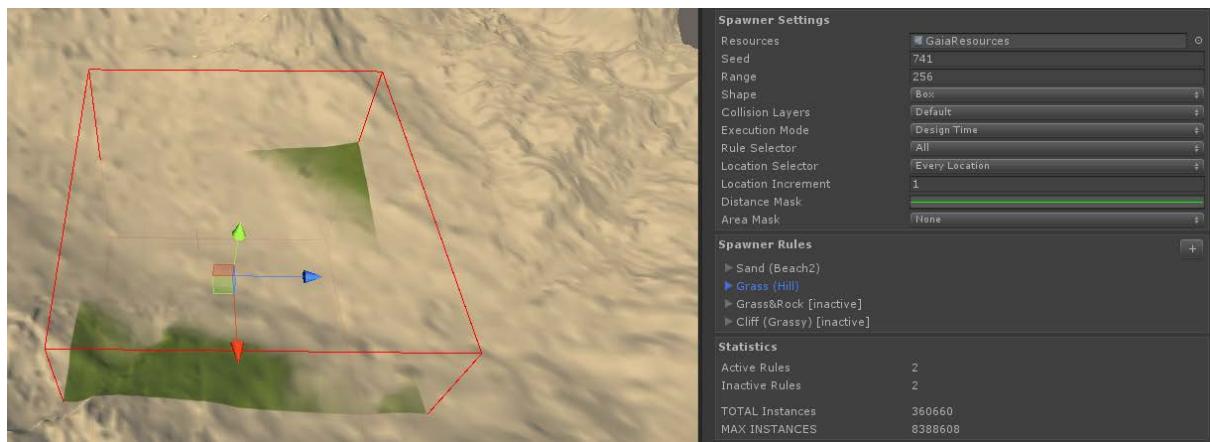
Another handy trick, and a good way to build up your texture layers is to make your spawner smaller by decreasing its range, then deactivate the rules, and then add them back one at a time.



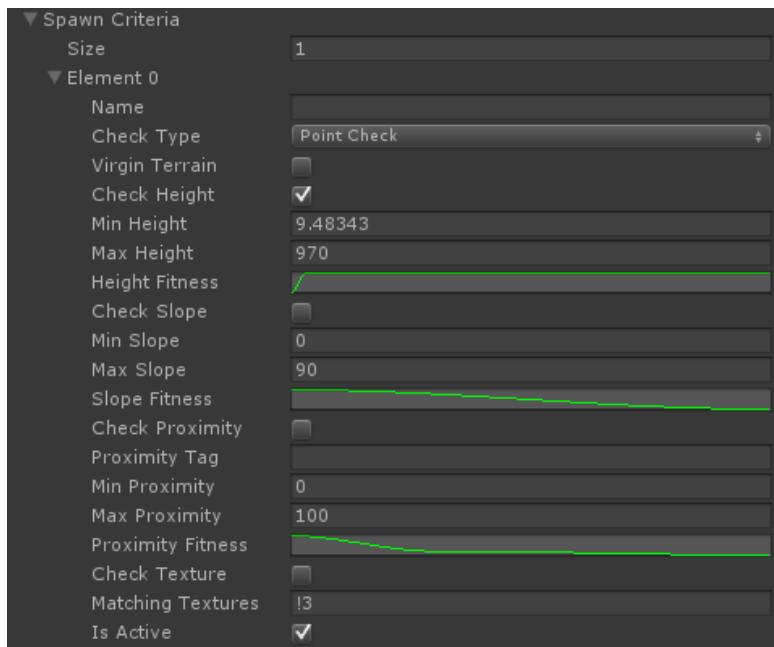
Just the sand.



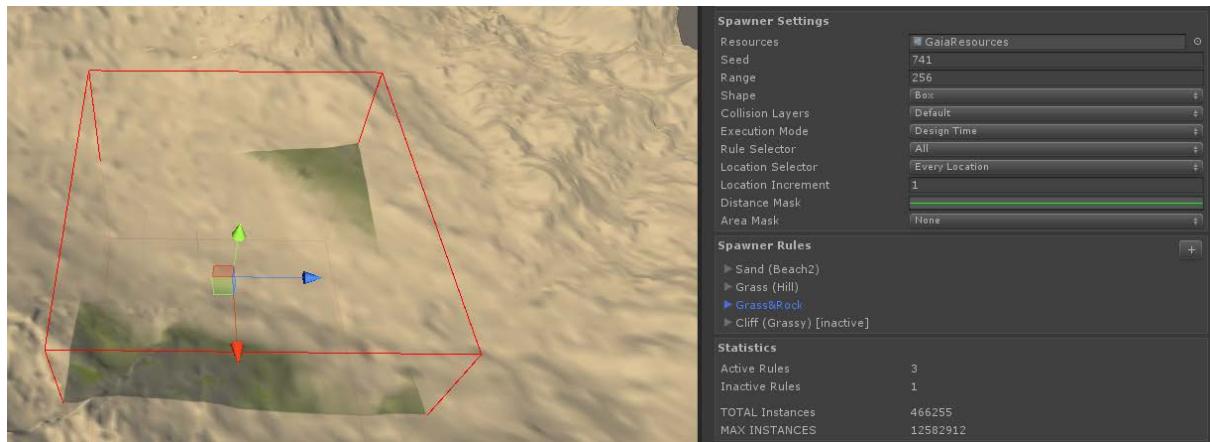
Sand spawn criteria – spawning at full strength from 30 meters below sea level to 970 meters above sea level – and essentially wiping out any pre-existing texture at that location.



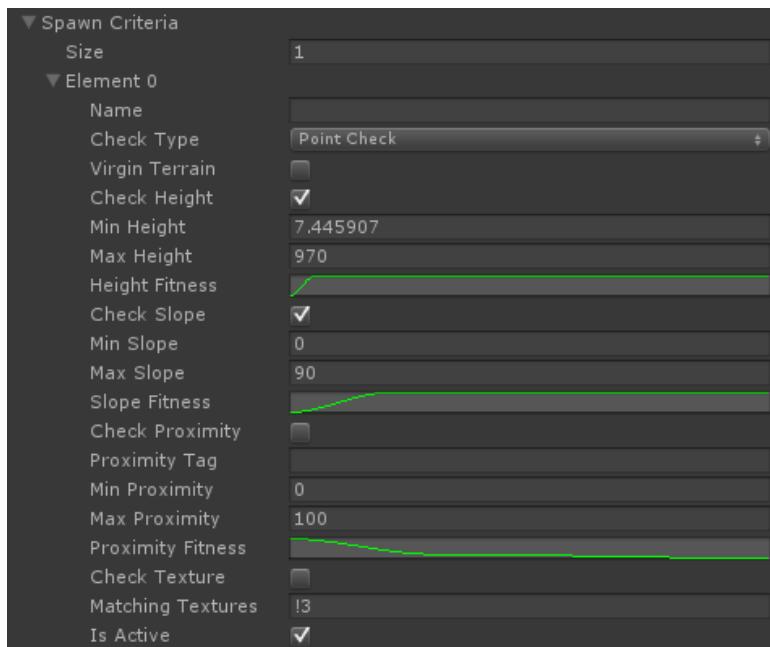
## Sand then Grass 1



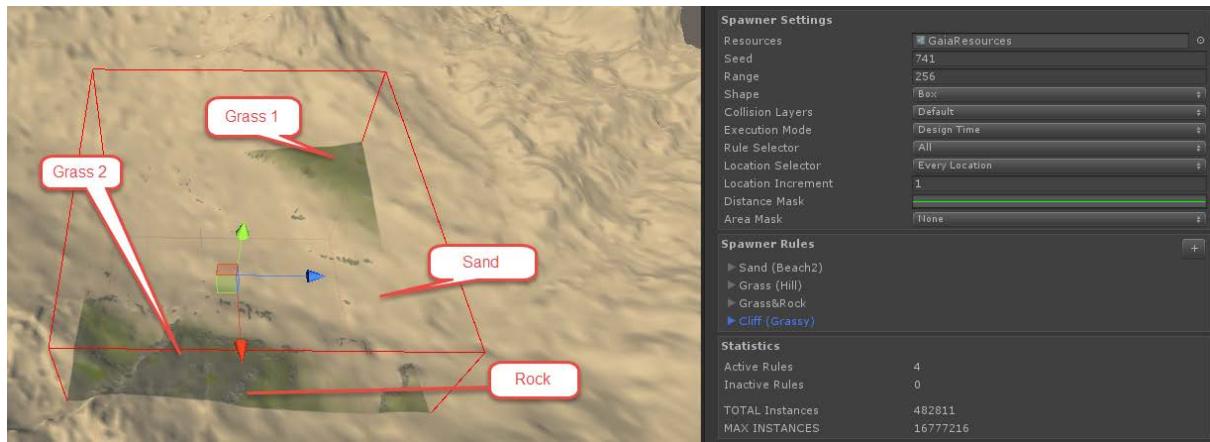
Grass 1 spawn criteria – spawning between 9.4 and 970 meters above sea level. Notice the curve – this is giving us a nice blend effect.



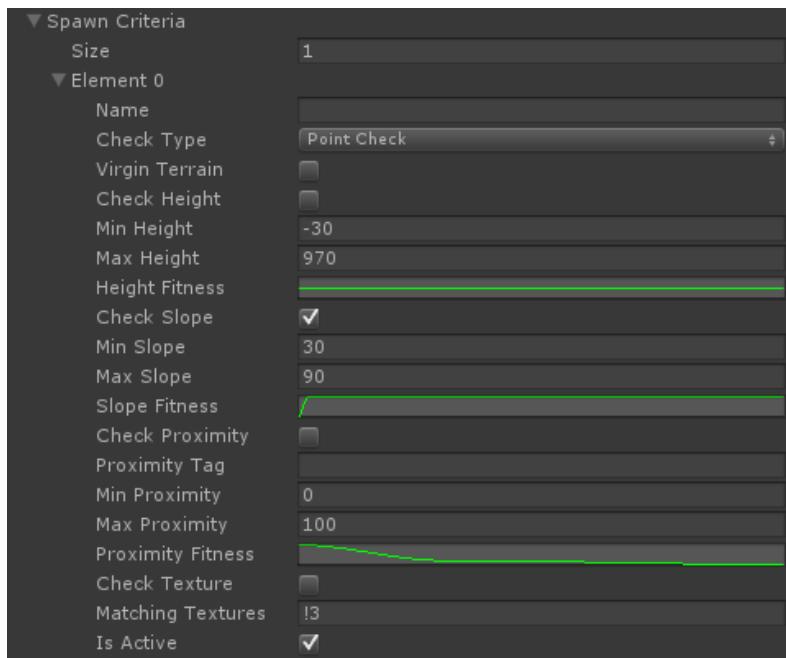
Sand then Grass 1 then Grass 2



Grass 2 spawn criteria – spawning between 7.4 and 970 meters above sea level, and also checking slope, and modifying its strength at different slope angles. This gives us some variation in the way the two grasses get mixed together.

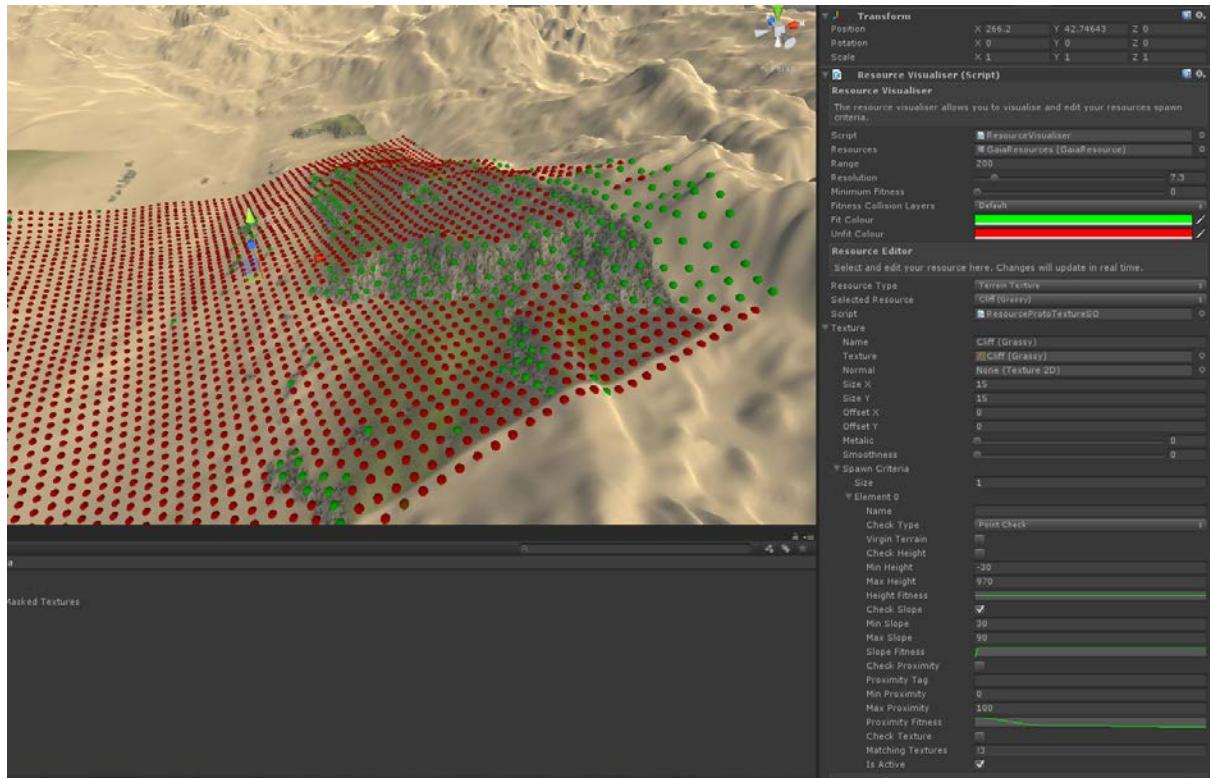


Sand then Grass 1 then Grass 2 then Rock



Rock spawn criteria – spawning wherever the slope is between 30 and 90 degrees.

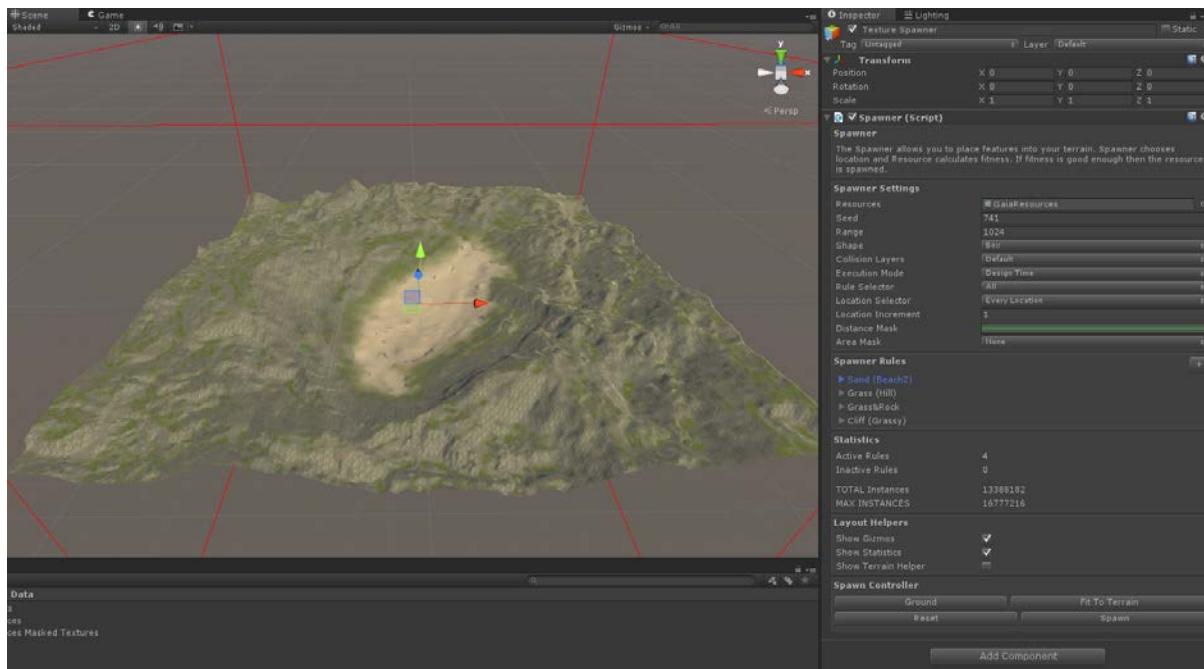
You can open up the Visualiser to fine tune your spawn criteria.



Notice how the visualiser shows where the cliff texture will go, and how strong it is.

When you have adjusted your spawn criteria rules, hit Reset on your spawner and then Spawn again to re-texture the terrain.

When you are done, you can click the 'Fit To Terrain' button, then the Reset button, and finally the Spawn button to do the entire terrain.



Note the TOTAL Instances number – this was the number of times a texture was applied to the terrain!

Incidentally this is why we also hit the Reset button. The spawner will not allow any rule to be spawned more than its maximum instances, and this has the effect of setting it back to zero. It also resets the spawners random number generator back to its seed value.

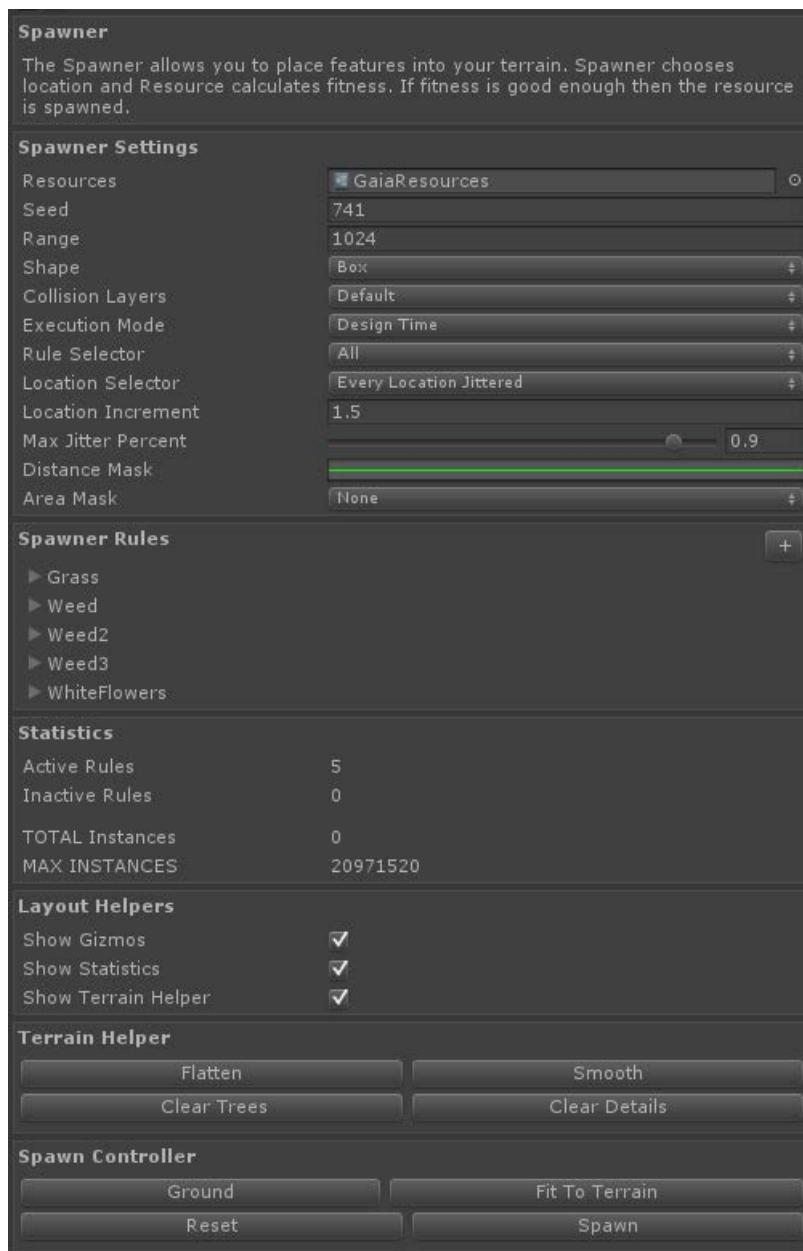
## Coverage Detail Spawners

Coverage Detail Spawners are designed to add details like grass coverage across to your terrain. They are intended to be used to provide general ground cover.

First make create and stamp a terrain. Then open Gaia Manager and make sure you have the right resources file selected.

Then use Gaia Manager to create a Coverage detail spawner. In advanced mode you can create as many detail spawners as you like.

Gaia Manager will create and configure the spawner for you, and add all the detail resources in your resources file into it.



By default the detail spawner will be fitted to your terrain, however you can easily resize and move it around as you please.

The Location Selector has been set to Every Location Jittered, and will iterate through every location on the terrain at 1.5m meter intervals, and then jitter that location by up to 90%.

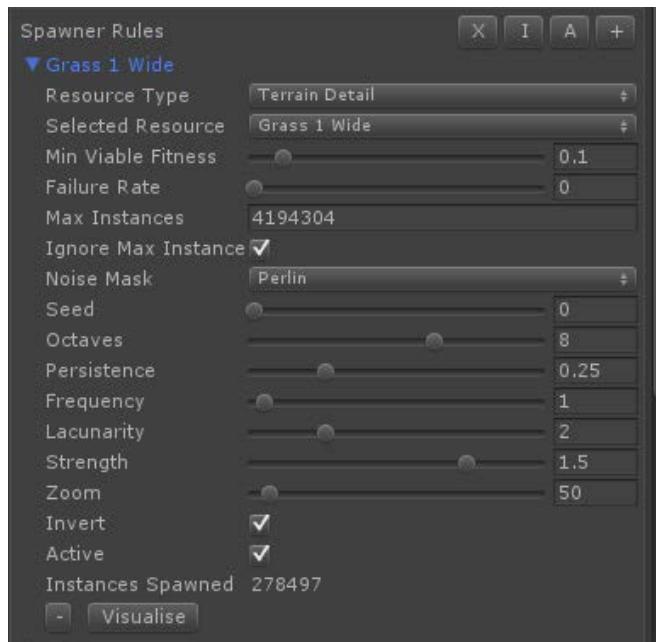
This has the effect of evaluating the majority of the terrain in order to plant grass, but also randomises or jitters the location so that straight lines are not created.

The Rule Selector in a detail spawner is set to execute all rules, and there is one rule for every detail that will be applied.

When a rule is executed, the associated resources fitness criteria is evaluated to derive a fitness value for that resource at that location.

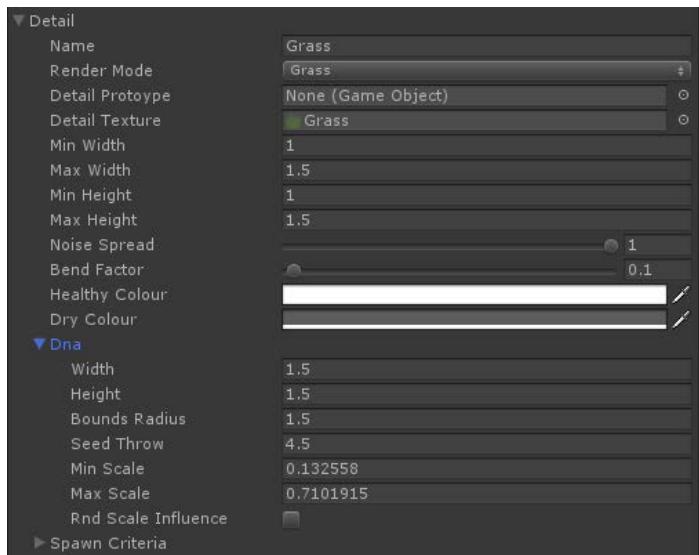
If the resources fitness is greater than the minimum viable fitness, and the rule is not cancelled due to the failure rate, then the detail will be spawned at that location.

You can further influence where the grass spawns by applying noise masks. This allows you to create interesting variation in your spawned grass patches.



Detail strength and density is driven by the minimum and maximum scale values defined in the details DNA, which should be set between 0 and 1.

If Rnd Scale Influence is selected then a random strength will be chosen, otherwise the scale value is based on the fitness at that location. Higher fitness will result in denser grass.



The result of detail spawning.

Another nice technique for detail spawning is to use a clustered approach that generates clusters of interesting grasses. This is particularly nice to generate clusters of flowers.



NOTE: By default details are set to only be placed on 'Virgin' terrain. This is so that grass doesn't grow through things, and consequently you should always spawn details last.

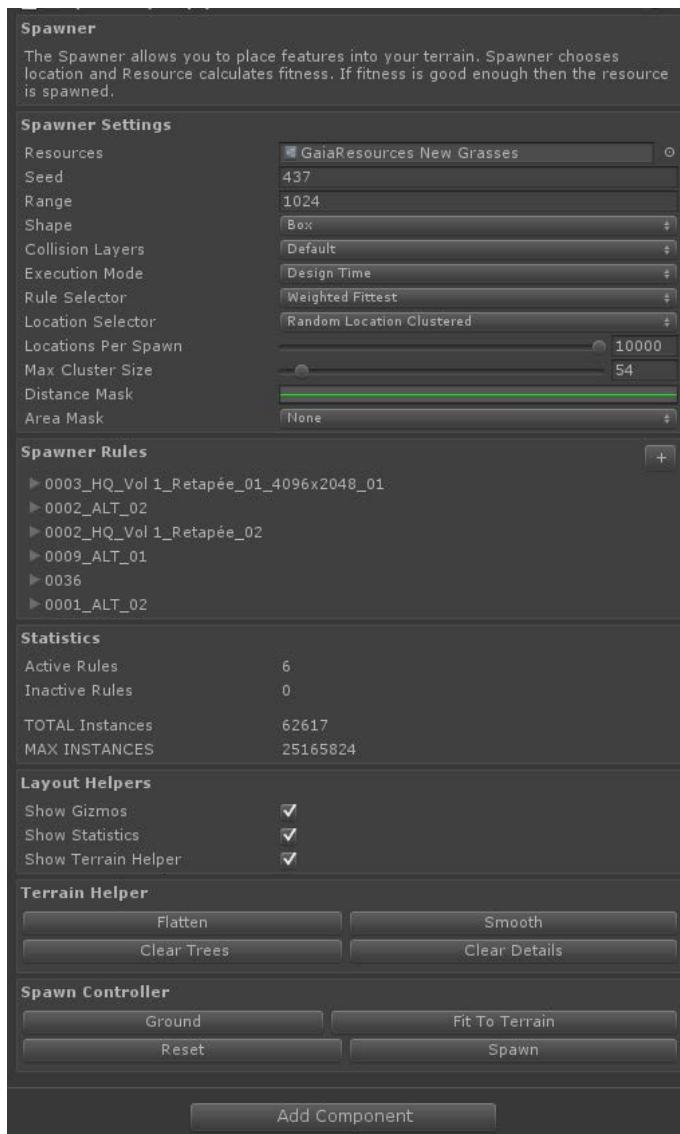
## Clustered Detail Spawners

Clustered Detail Spawners are a specialised detail spawner designed to spawn details in natural clusters. You would typically use these to add splashes of colour to your terrain with nice flowers.

First make create and stamp a terrain. Then open Gaia Manager and make sure you have the right resources file selected.

Then use Gaia Manager ADVANCED menu to create a clustered detail spawner. In advanced mode you can create as many clustered detail spawners as you like.

Gaia Manager will create and configure the spawner for you, and add all the detail resources in your resources file into it.



By default the detail spawner will be fitted to your terrain, however you can easily resize and move it around as you please.

Delete or de-activate the details that you are not interested in. It's often a nice idea to make one clustered spawner per type of flower that you are spawning – this will keep colours and flowers nicely grouped together.

The Location Selector has been set to Random Location Clustered, and will randomly choose locations on your terrain to spawn clusters of up to Max Cluster Size.

The Rule Selector in a detail spawner is set to execute Weighted Fittest, and there is one rule for every detail that will be considered for spawning.

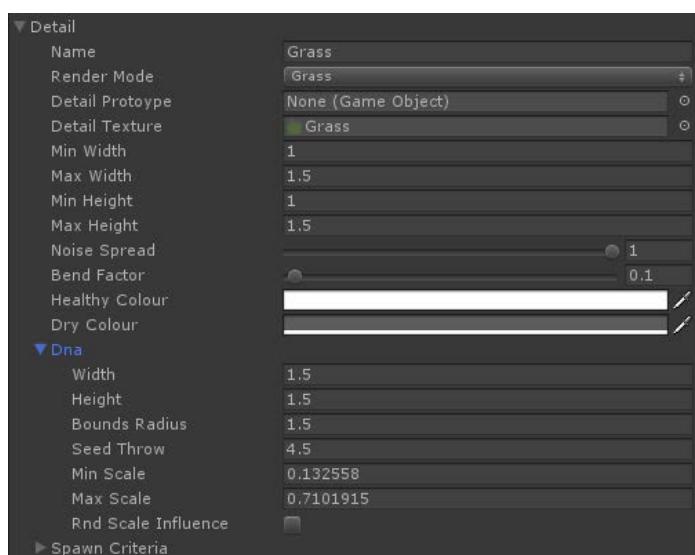
When a rule is executed, the associated resources fitness criteria is evaluated to derive a fitness value for that resource at that location.

If the resources fitness is greater than the minimum viable fitness, and the rule is not cancelled due to the failure rate, then the detail will be considered for spawning at that location.

The rule that will be selected to spawn will be in proportion to its fitness, so fitter rules will spawn more often than less fit rules.

Detail strength and density is driven by the minimum and maximum scale values defined in the details DNA, which should be set between 0 and 1.

If Rnd Scale Influence is selected then a random strength will be chosen, otherwise the scale value is based on the fitness at that location. Higher fitness will result in denser grass.



The result of clustered detail spawning.



NOTE: By default details are set to only be placed on 'Virgin' terrain. This is so that grass doesn't grow through things, and consequently you should always spawn details last.

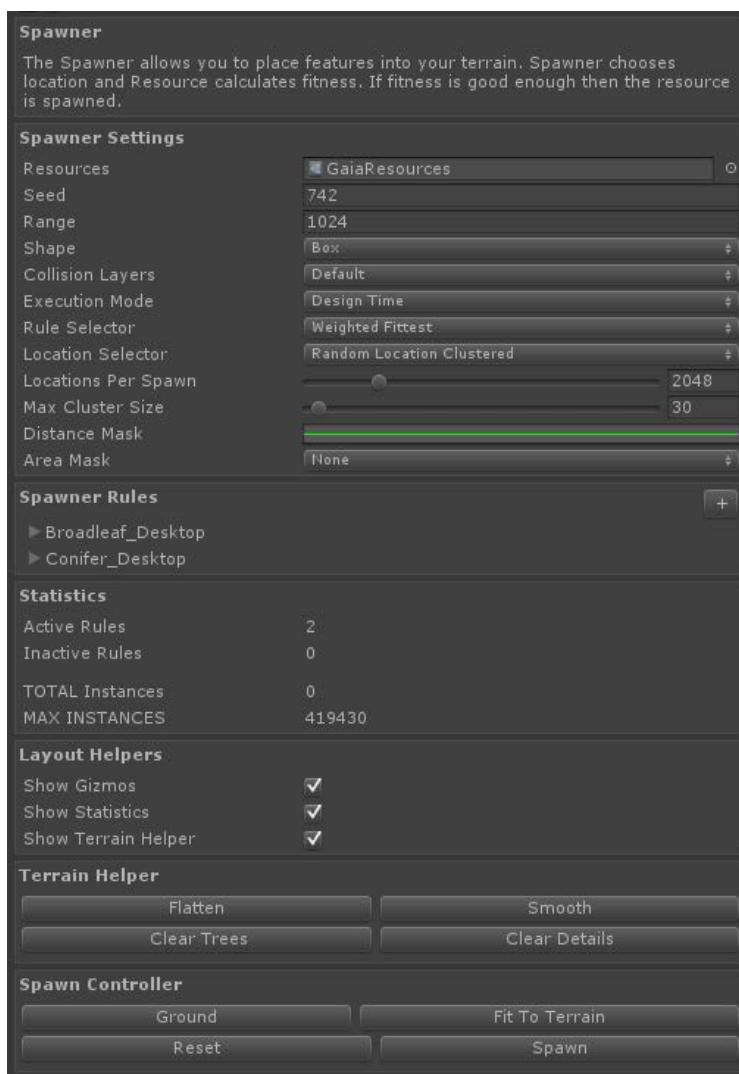
## Tree Spawners

Tree Spawners are designed to add trees to your terrain.

First make create and stamp a terrain. Then open Gaia Manager and make sure you have the right resources file selected.

Then use Gaia Manager to create a tree spawner. In advanced mode you can create as many tree spawners as you like.

Gaia Manager will create and configure the spawner for you, and add all the tree resources in your resources file into it.



By default the tree spawner will be fitted to your terrain, however you can easily resize and move it around as you please.

The Location Selector has been set to Random Location Clustered, there will be 2048 location checks per spawn iteration, and the maximum cluster size is 30.

This will cause the spawner to check up to 2048 locations per iteration, and spawn clusters of up to 30 trees.

By default trees are set to only be placed on 'Virgin' terrain, and area check mode is set on them rather than point check mode. For these to work properly you also need to have colliders enabled on your trees to detect collisions.

The density of the cluster will be dependent on the bounds radius of the tree, and the seed throw value.

The bounds radius is the radius from the centre of the tree to the edge of its bounds, and this is used to determine how far to check for collisions with other objects or trees when placing the tree.

The seed throw value is how far a spawned tree can throw a seed when in clustered mode to generate another tree nearby. Every tree spawned in clustered mode will get one or more opportunities to spawn new trees until either the cluster size is reached, or every tree in the cluster has tried and failed to spawn.

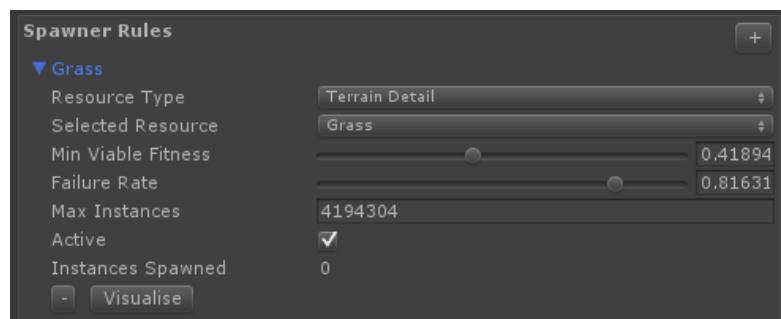
When the cluster is complete another random location is chosen and cluster generation initiated again until the locations checked per spawn has been exceeded.

The Rule Selector in a tree spawner is set to Weighted Fittest, and there is one rule for every tree that will be applied.

Using Weighted Fittest instead of fittest will create nice blended borders between the fittest tree species for every environment.

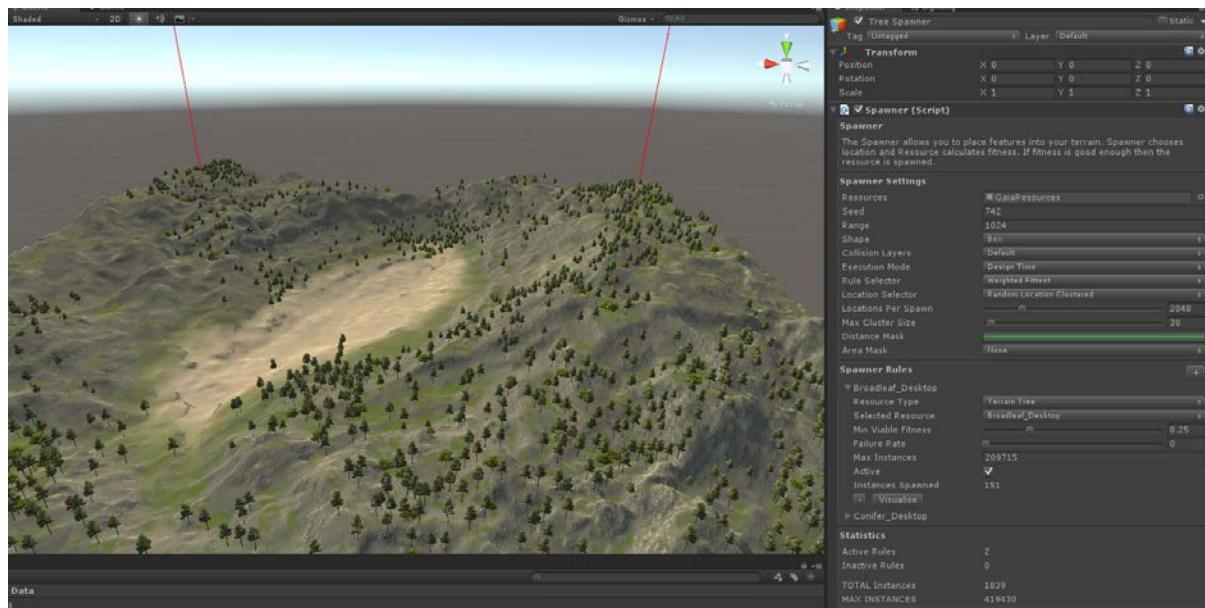
When a rule is executed, the associated resources fitness criteria is evaluated to derive a fitness value for that resource at that location.

If the resources fitness is greater than the minimum viable fitness, and the rule is not cancelled due to the failure rate, then the tree will be spawned at that location.



Tree scale is driven by the minimum and maximum scale values defined in the trees DNA.

If Rnd Scale Influence is selected then a random scale will be chosen, otherwise the scale is based on the fitness at that location. Higher fitness will result in bigger trees.



The result of tree spawning.



A sample using clustered tree spawning.

## Game Object Spawners

Game Object Spawners are designed to add game object to your scene.

First make create and stamp a terrain. Then open Gaia Manager and make sure you have the right resources file selected.

Then use Gaia Manager to create a game object spawner. In advanced mode you can create as many game object spawners as you like.

Gaia Manager will create and configure the spawner for you, and add all the game object resources in your resources file into it.



You can remove any that you don't want, or deactivate them all, activate them all or add new rules.



By default the game object spawner will be fitted to your terrain, however you can easily resize and move it around as you please.

The Location Selector has been set to Every Location Jittered, and this will cause the spawner to check every 90 meters, with a random jitter of up to 85% of 90 meters to break up lines.

By default game objects are set to only be placed on 'Virgin' terrain, and area check mode is set on them rather than point check mode.

The spawner will add sphere colliders to the spawned game object instances based on the individual instance bounds as well as the overall POI bounds.

The bounds radius is the radius from the centre of the game object to the edge of its bounds, and this is used to determine how far to check for collisions with other objects or trees when placing the game object.

When the spawner has spawned its instances the next location will be checked until the terrain has been exceeded or the instance count has been exceeded. At a spawner level, the instance count is the number of spawns that were actually spawned (not the instances within each spawn).

The Rule Selector in a game object spawner is set to Random, and there is one rule for every game object that will be applied. This will randomly choose one of the game objects and attempt to spawn it.

When a rule is executed, the associated resources fitness criteria is evaluated to derive a fitness value for that resource at that location.

If the resources fitness is greater than the minimum viable fitness, and the rule is not cancelled due to the failure rate, then the game object will be spawned at that location.

Game object scale is driven by the minimum and maximum scale values defined in the game objects DNA.

If Rnd Scale Influence is selected then a random scale will be chosen, otherwise the scale is based on the fitness at that location. Higher fitness will result in bigger game

objects.

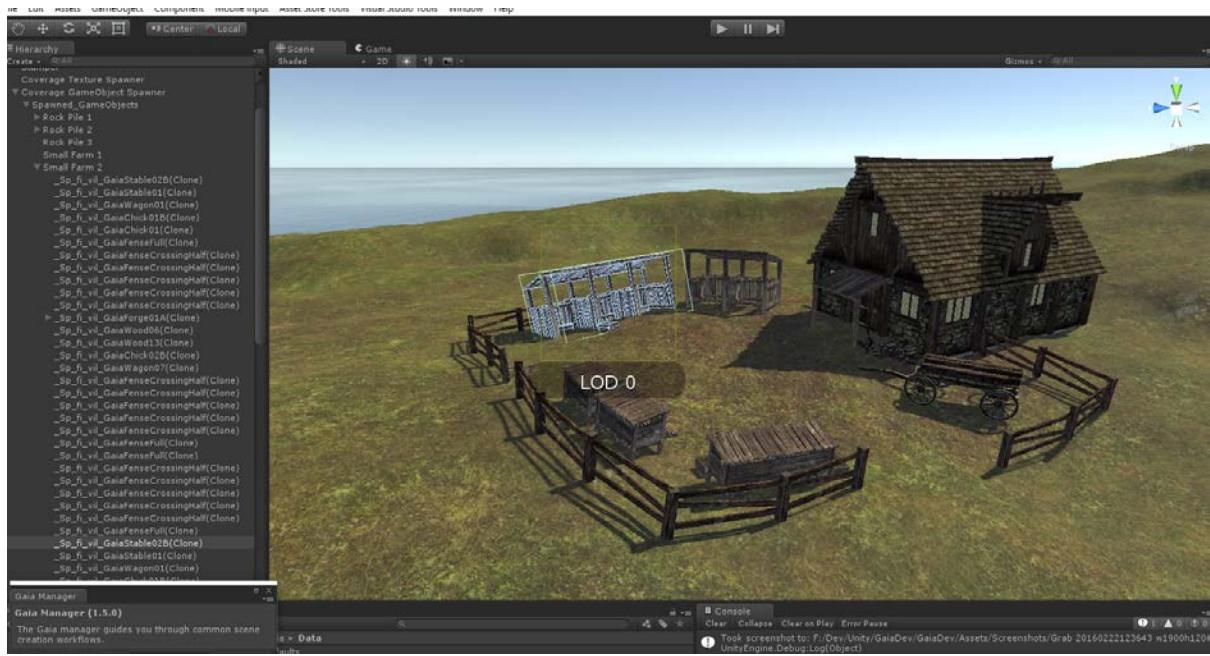


While the spawner chooses a spawn location and rotation, the instances within that spawn are spawned relative to the spawn location and rotation.

Game object instances can be embedded above or below the terrain by changing their terrain offset.

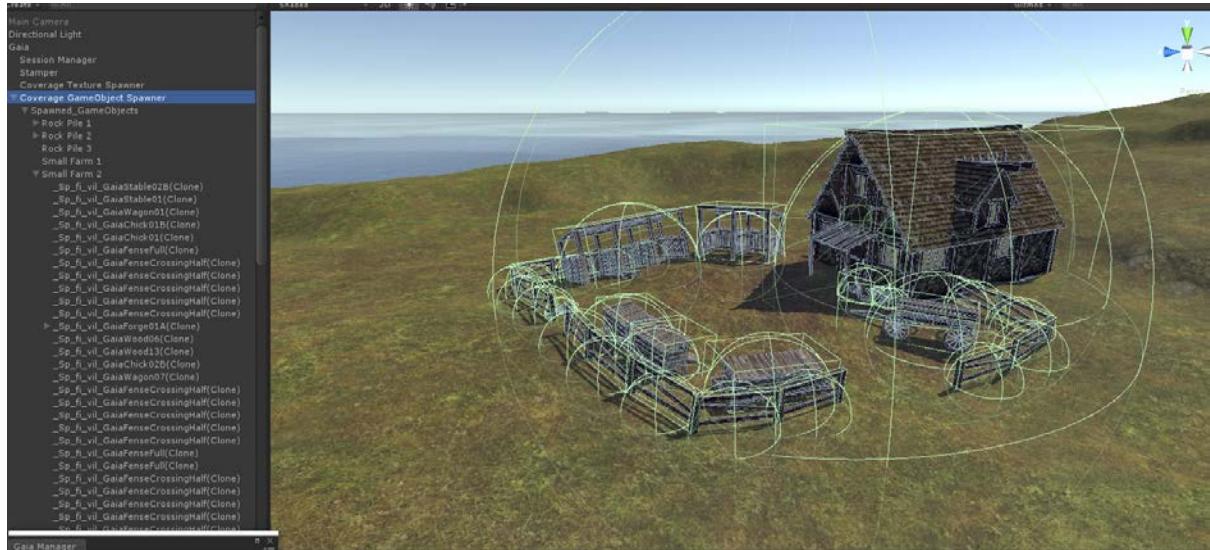
They can also rotate so that they conform to the terrain slope / normal or be randomly located and between the minimum and maximum values provided.

The game objects created are parented to the spawner, and this can be used to your advantage to find tune where the game object has been placed.



To do this double click on the instance, which is parented to the Spawner to zoom to the specific instantiation. You can then fine tune location as you please using standard techniques.

You can also re-parent the object to another object if you wasn't to. There is no link between the spawned Game Object and the Spawner other than parenting for convenience and neatness.

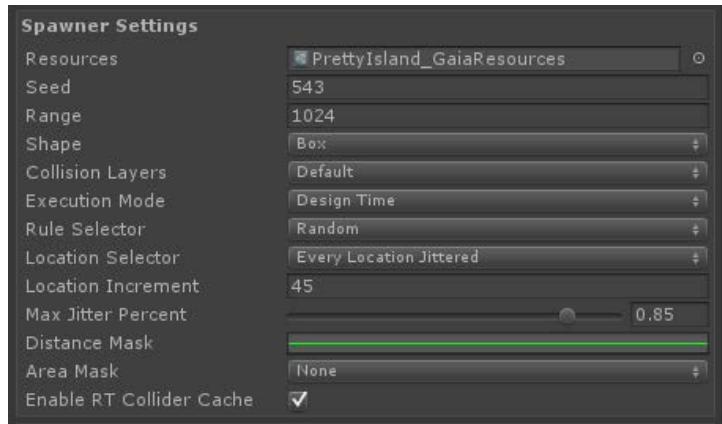


When spawning trees and game object the spawner adds sphere and capsule colliders to them as way of detecting collisions. These are purely used in the editor and are disabled at run time. Even though grass has been set to virgin terrain, it will spawn though these colliders unless 'nograss' has been selected in Ext Params on the dna of the tree or game object being spawned. This allows grass to spawn nicely around the objects.

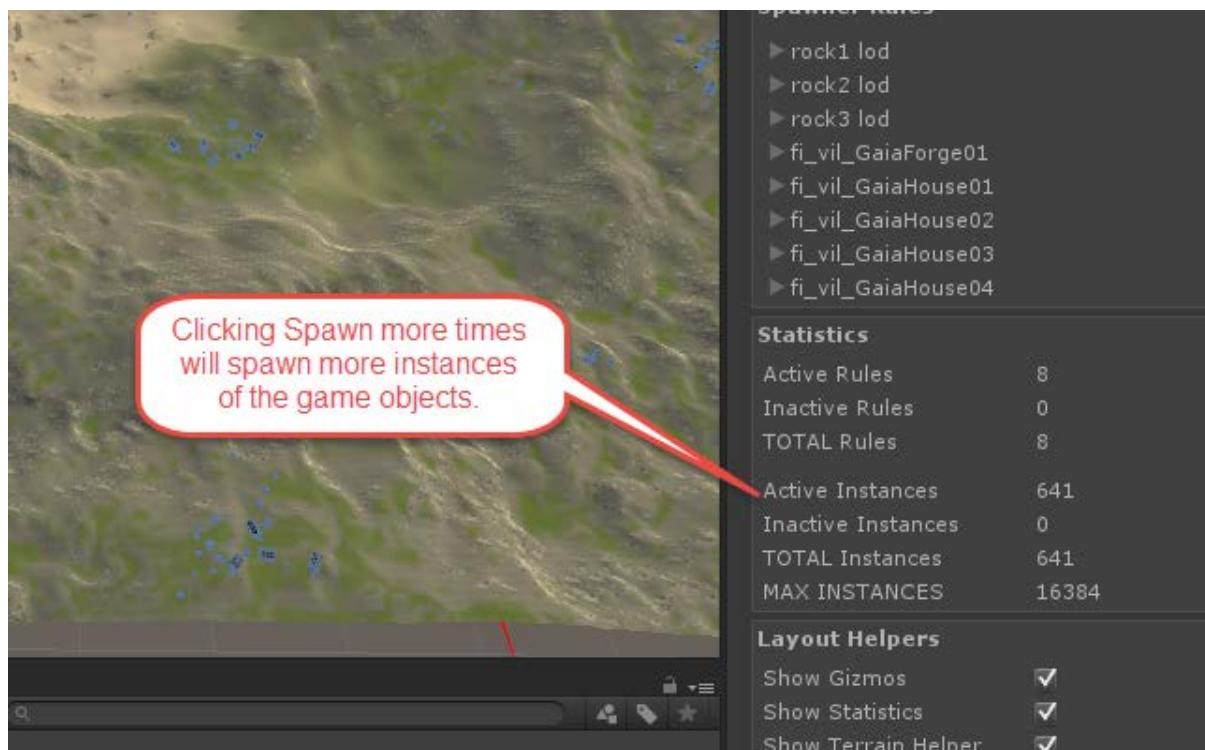
However we do not want grass to spawn through objects like houses etc, so by

default the spawner will also add mesh colliders to any mesh that does not have colliders and will not spawn when these collisions are detected.

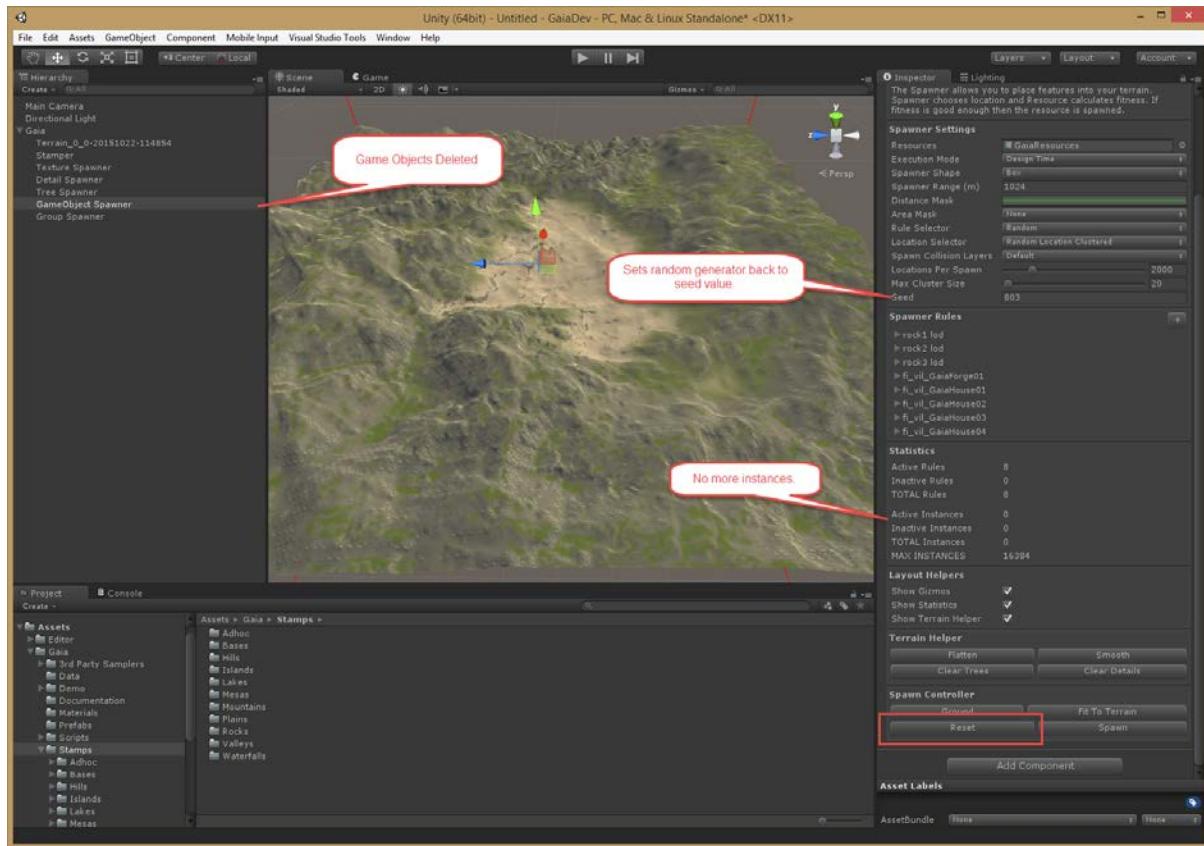
You can choose to leave the mesh colliders that were added by the active at runtime by selecting 'Enable RT Collider Cache' on the spawner.

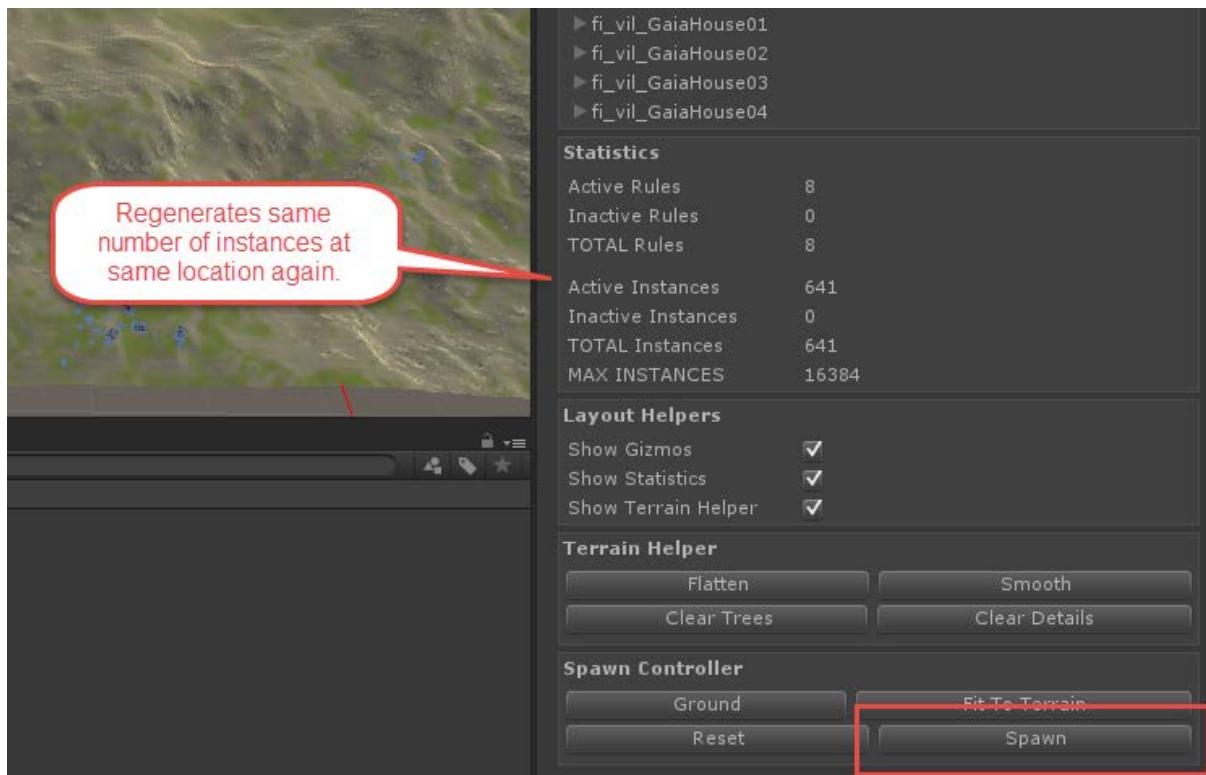


Clicking Spawn more times will spawn more game objects.



Clicking the Reset button will delete all the Game Objects that were instantiated and Reset the random number generator to its seed value again.





Clicking Spawn again a few times will regenerate the same number of instances at the same location again. See the number of instances on the previous page to verify this.

Note: Gaia is driven by rules in conjunction with random numbers. If nothing else has changed in a scene then a Spawner will always generate exactly the same output, regardless of the number of time it is iterated on after a Reset.

However, if the underlying scene has been changed, Gaia is not guaranteed to generate the same output, because the iterative process may also cause a decision to spawn to go another way, which then causes the number of random numbers generated to be different, and then next location will not be generated in the same sequence.



A randomly spawned village.

## Spawner Settings

For an overview on how to use Spawners check out the Work Flow in the previous section.

### Spawner Settings

High level spawner settings.

#### Resources

The object that contains the resources that these rules will apply to.

#### Seed

The unique seed for this spawner. If the environment, resources or rules don't change, then hitting Reset and respawning will always regenerate the same result.

#### Range

Distance in meters from the centre of the spawner that the spawner can spawn in. Shown as a red box or sphere in the gizmos.

#### Shape

The shape of the spawn area. The spawner will only spawn within this area.

*Box:* Box shaped.

*Sphere:* Sphere shaped.

#### Collision Layers

Controls which layers are checked for collisions when spawning. Must at least include the layer the terrain is on. Add additional layers if other collisions need to be detected as well. Influences terrain detection, tree detection and game object detection. Will always automatically add the terrain layer here.

#### Execution Mode

The way this spawner runs.

*Design time:* At design time only.

*Runtime Interval:* At run time on a timed interval.

*Runtime Triggered Interval:* At run time on a timed interval, and only when the tagged game object is closer than the trigger range from the centre of the spawner.

#### Spawn Interval

The time in seconds between spawn iterations. Only valid for interval based execution modes.

## Trigger Range

Distance in meters from the centre of the spawner that the trigger will activate. Only valid in Runtime Triggered Interval execution mode. Shown as a yellow box or sphere.

## Trigger Tags

The tags of the game objects that will set the spawner off. Multiple tags can be separated by commas e.g. Player,Minion etc. Only valid in Runtime Triggered Interval execution mode.

## Rule Selector

The way a rule is selected to be spawned.

*All*: All rules are selected.

*Fittest*: Only the rule with the fittest spawn criteria is selected. If multiple rules have the same fitness then one will be randomly selected.

*Weighted Fittest*: The chance of a rule being selected is directly proportional to its fitness. Fitter rules have more chance of selection. Use this to create more natural blends between objects.

*Random*: Randomly selected.

## Location Selector

How the spawner selects locations to spawn in.

*Every Location*: The spawner will attempt to spawn at every location.

*Every Location Jittered*: The spawner will attempt to spawn at every location, but will offset the location by a random jitter factor. Use this to break up lines.

*Random Location*: The spawner will attempt to spawn at random locations.

*Random Location Clustered*: The spawner will attempt to spawn clusters at random locations.

## Locations Per Spawn

The number of locations that will be checked every Spawn interval. This does not guarantee that something will be spawned at that location, because lack of fitness may preclude that location from being used. Only available for Random based location selection.

## Max Cluster Size

The maximum individuals in a cluster before a new cluster is started. Only available for Random Location Clustered selection.

## Location Increment

The distance from the last location that every new location will be incremented in meters. Only available for Every Location based location selection.

## Max Jitter Percent

Every new location will be offset by a random distance up to a maximum of the jitter percentage multiplied by the location increment. Only available for Every Location Jittered location selection.

## *Distance Mask*

Mask fitness over distance. Left hand side of curve represents the centre of the spawner. Use this to alter spawn success away from centre e.g. peter out towards edges.

## *Area Mask*

An area mask is applied across the entire spawner.

None - No area mask will be applied.

Image Grey Scale - the mask is sourced from the grey scale conversion of the image mask.

Image Red Channel - the mask is sourced from the red channel of the image mask.

Image Green Channel - the mask is sourced from the green channel of the image mask.

Image Blue Channel - the mask is sourced from the blue channel of the image mask.

Image Alpha Channel - the mask is sourced from the alpha channel of the image mask.

Terrain Texture 0 - the mask is sourced from texture slot 0 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 1 - the mask is sourced from texture slot 1 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 2 - the mask is sourced from texture slot 2 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 3 - the mask is sourced from texture slot 3 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 4 - the mask is sourced from texture slot 4 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 5 - the mask is sourced from texture slot 5 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 6 - the mask is sourced from texture slot 6 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Terrain Texture 7 - the mask is sourced from texture slot 7 of the terrain.  
Allows you to paint your mask on the terrain and then use it.

Perlin Noise - the mask is sourced from the output of a Perlin Noise function.

Billow Noise - the mask is sourced from the output of a Billow Noise function.

Ridged Noise - the mask is sourced from the output of a Ridged Noise function.

## *Image Mask*

The image to be used as the source of area mask.

### *Smooth Mask*

The number of smooth iterations to be applied to the mask before using it. Can be used to soften noisy masks, and to soften mask edges.

### *Normalise Mask*

The mask will be normalised to use the full dynamic range of the mask.

### *Invert Mask*

The mask will be inverted to reverse its area of effect.

### *Flip Mask*

The mask will be flipped on its x & y axes before applying it. Useful sometimes to match it with the unity terrain which internally is also flipped.

### *Noise Seed*

The seed passed through to the noise function.

### *Octaves*

The amount of detail in the noise. More octaves mean more detail and longer calculation times.

### *Persistence*

The roughness of the noise. Controls how quickly amplitudes diminish for successive octaves. 0..1.

### *Frequency*

The frequency of the first octave.

### *Lacunarity*

The frequency multiplier between successive octaves. Experiment between 1.5 - 3.5.

### *Zoom*

The zoom level of the noise. Larger zooms display the noise over larger areas.

### *Enable RT Collider Cache*

Leaves the collider cache that was created by the spawner active at run time. This only applies to game objects.

## Spawner Rules

Spawner rules create the link between the spawner and the resources they represent.

After the spawner has chosen a location, it will get the resource referenced by the rule to assess that location for fitness, and depending on how it has been configured will optionally attempt to spawn one or more resources at that location.

The spawn rule contains a reference to the resource and the criteria used to control whether the resource can be spawned.

The spawner chooses which rules to execute via the Rule Selector:

### Rule Selector

**The way a rule is selected to be spawned.**

**ALL:** All rules are selected.

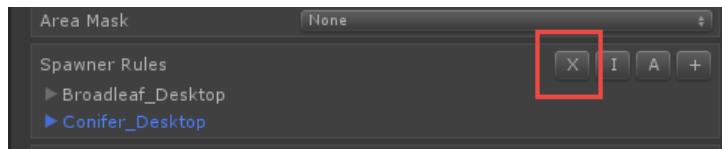
**Fittest:** Only the rule with the fittest spawn criteria is selected. If multiple rules have the same fitness then one will be randomly selected.

**Weighted Fittest:** The chance of a rule being selected is directly proportional to its fitness. Fitter rules have more chance of selection. Use this to create more natural blends between objects.

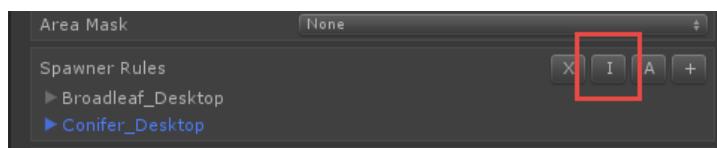
**Random:** Randomly selected.

You would typically only ever run 'All' rules when spawning textures or details, because they are multi layered and can have multiple active instances of a resource in the same location.

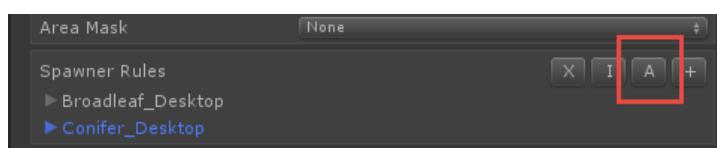
To delete all rules hit the [X] button.



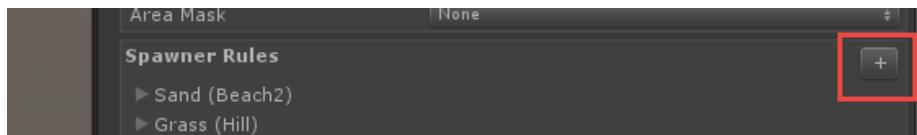
To deactivate all rules hit the [I] button.



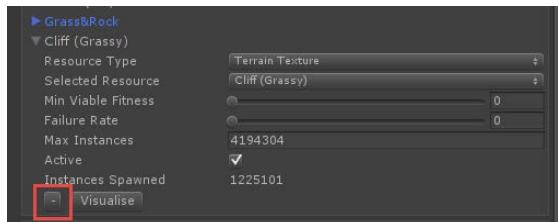
To activate all rules hit the [A] button.



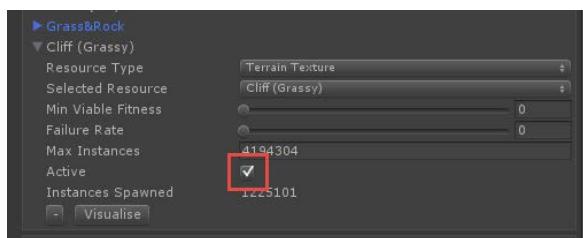
To add a new rule hit the [+] plus button.



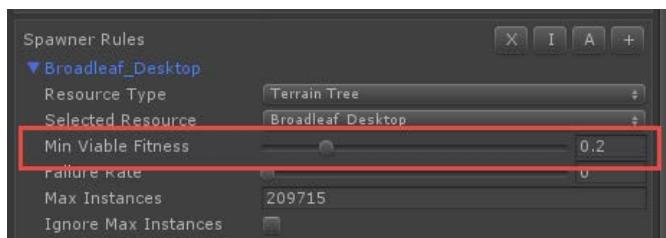
To delete a Rule hit the [-] minus button.



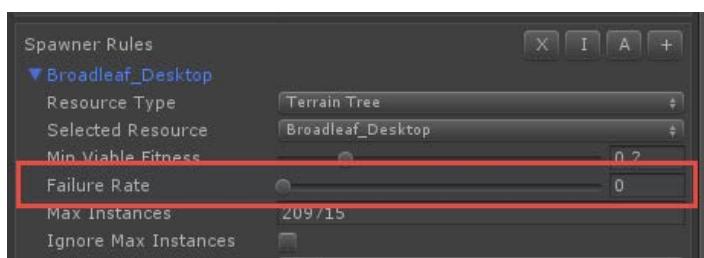
To deactivate a rule, de select the Active button.



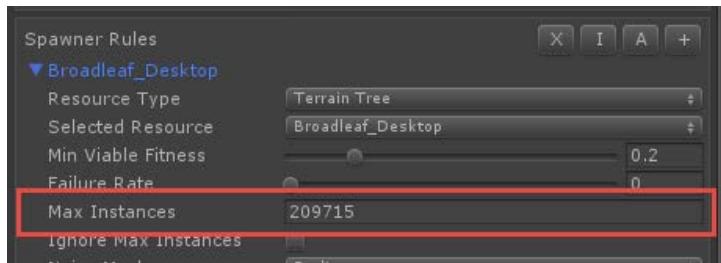
To choose the fitness cutoff point below which a location is not deemed fit for spawning use the minimum viable fitness slider.



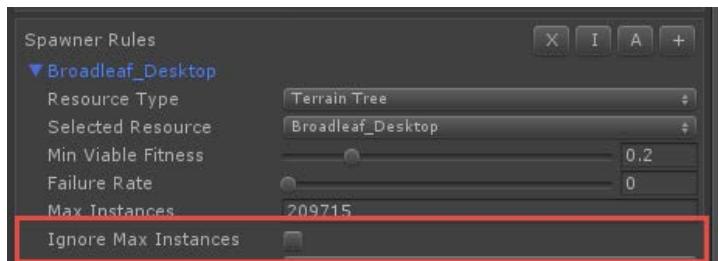
To arbitrarily fail an otherwise good location use the Failure rate slider. This is good for thinning things like grass out.



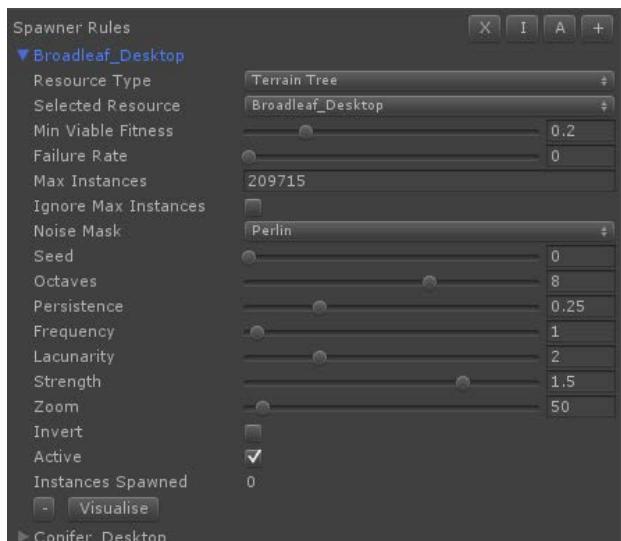
To limit the maximum number of instances of an rule that can be spawned use the max Instances setting.



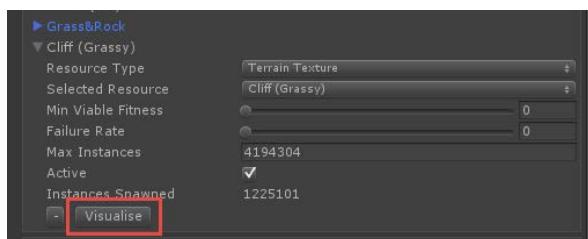
To ignore this completely use the Ignore Max Instances setting.

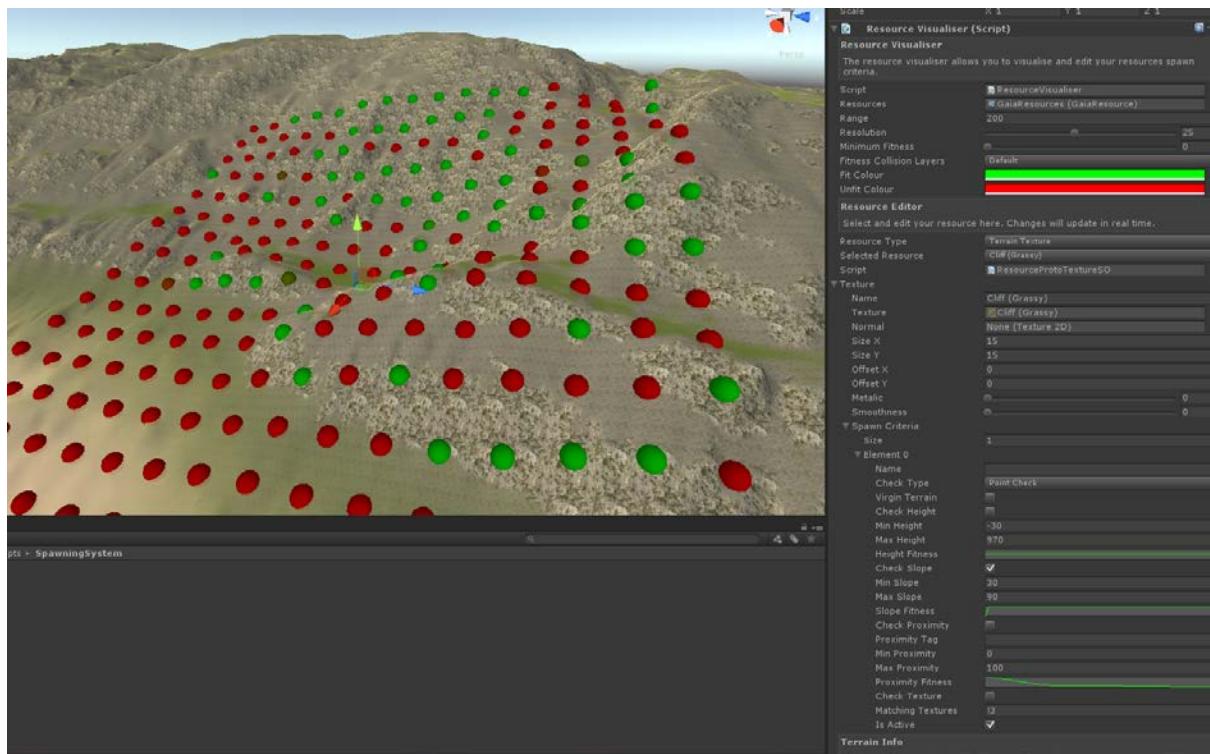


To provide an additional mask on where a rule can spawn select one of the noise masks.



To visualise fitness and tweak the spawn criteria of the resource associated with the rule hit the Visualise Button.





To learn more about the Visualiser check out the Visualiser section in Resources.

## *Resource Type*

The type of resource this rule will apply to.

## *Selected Resource*

The resource this rule applies to. To modify how the resource interprets terrain fitness change its spawn criteria.

## *Min Viable Fitness*

The minimum fitness needed to be considered viable to spawn.

## *Failure Rate*

The amount of the time that the rule will fail even if fit enough. 0 means never fail, and 1 means always fail. Use this to thin things out.

## *Max Instances*

The maximum number of resource instances this rule can spawn. Use this to stop over population.

## *Ignore Max Instances*

Will ignore the maximum instances limitation. Use this for very large instance counts e.g. large terrain texturing.

## *Active*

Whether this rule is active or not. Use this to disable the rule.

## *Instances Spawned*

The number of times this resource has been spawned.

## *Noise Mask*

An noise mask is applied across the entire terrain further limiting where a resource can spawn.

None - No noise mask will be applied.

Perlin Noise - the mask is sourced from the output of a Perlin Noise function.

Billow Noise - the mask is sourced from the output of a Billow Noise function.

Ridged Noise - the mask is sourced from the output of a Ridged Noise function.

## *Seed*

The seed passed through to the noise function.

## *Octaves*

The amount of detail in the noise. More octaves mean more detail and longer calculation times.

## *Persistence*

The roughness of the noise. Controls how quickly amplitudes diminish for successive octaves. 0..1.

### *Frequency*

The frequency of the first octave.

### *Lacunarity*

The frequency multiplier between successive octaves. Experiment between 1.5 - 3.5.

### *Zoom*

The zoom level of the noise. Larger zooms display the noise over larger areas.

### *Invert*

Invert the output off the noise.

## **Statistics**

You can enable or disable the display of statistics in the Layout Helpers section of the spawner. The statistics display helps provide a sense of how the spawner is working.

Statistics	
Active Rules	4
Inactive Rules	0
TOTAL Instances	13231502
MAX INSTANCES	16777216

## **Layout Helpers**

Handy helpers.

### *Show Gizmos*

Show the spawners gizmos.

### *Show Statistics*

Show spawner statistics.

### *Show Terrain Helper*

Shows the Terrain Helper buttons.

### *Flatten*

Flattens all terrains in the scene - not reversible so treat with care.

### *Smooth*

Smooths all terrains in the scene - not reversible so treat with care. It is generally good practice to run a single smooth iteration over your scene as the last step - it can have a remarkably positive impact on frame rate.

### *Clear Trees*

**Clears terrain trees in all terrains in the scene – not reversible so treat with care.**

### *Clear Details*

**Clears terrain details (grass) in all terrains in the scene – not reversible so treat with care.**

## **Spawn Controller**

Controls the spawner.

### *Ground*

**Position the spawner at ground level on the terrain.**

### *Fit To Terrain*

**Fits and aligns the spawner to the terrain.**

### *Reset*

**Resets the spawner, deletes any spawned game objects, and resets the random number generator.**

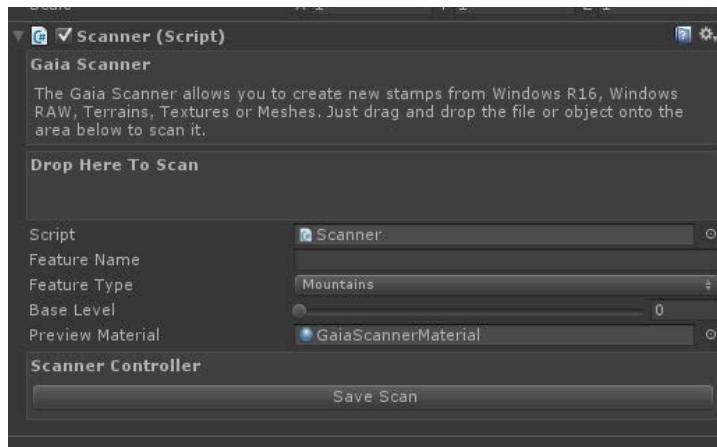
### *Spawn*

**Run a single spawn iteration. You can run as many spawn iterations as you like.**

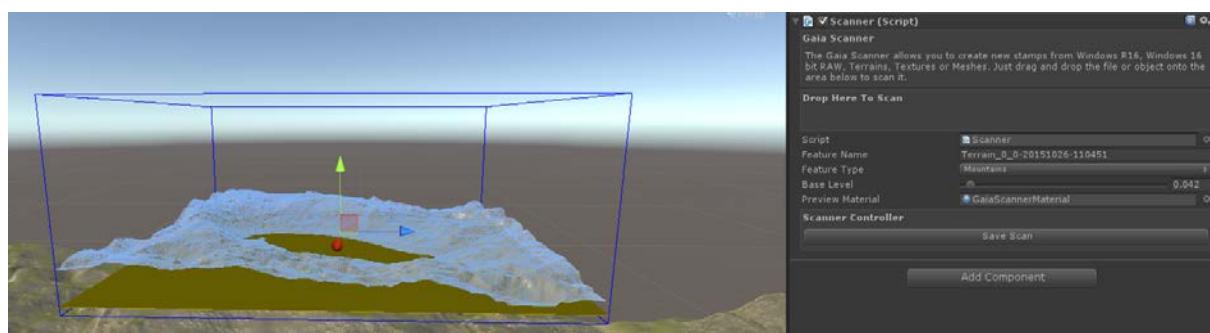
## Scanning System

The Scanning system allows you to easily extend the repertoire of stamps that can be leveraged by Gaia.

The scanner component provides a drop pad onto which you can place Terrains, Windows 16 Bit Raw Files, World Machine .16 (16 bit windows raw files), Meshes and 2D Textures, which the scanner will then scan.



Scan names are automatically picked up from the name of the asset that was dropped onto the Scanner pad, and can be overridden if desired.



You can then assign the scan a feature type to enable. This will control which directory it will be saved into to enable it to be categorised and retrieved.

The scan also has a base level assigned. This base level is passed through to the stamper to facilitate stamp placement. The base level is visualised by the yellow plane.

The scan will only be saved after you have hit the Save Scan button.

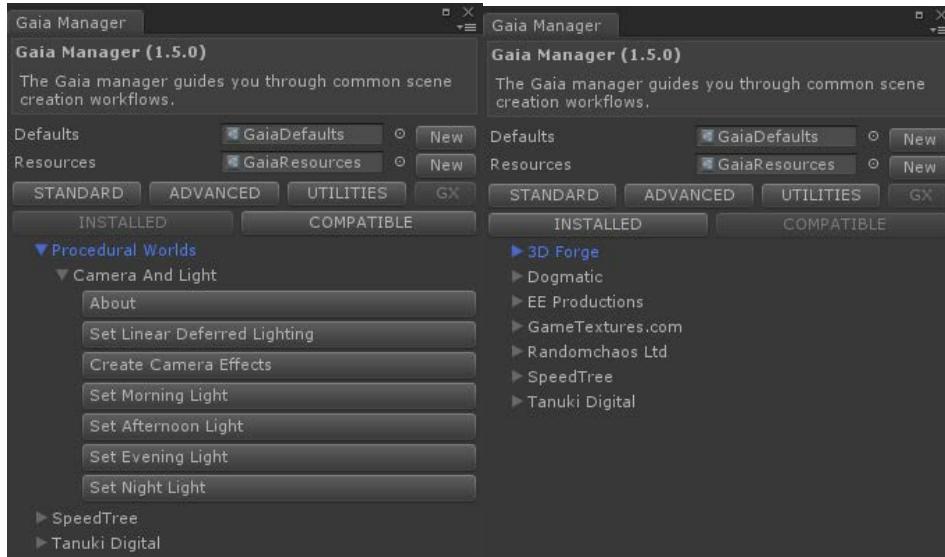
Two files are saved – a small jpg preview file, which shows the stamp as a grey scale height map, and a data file in the Data subdirectory. You need both of these files in order to be able to apply that stamp into the terrain.

If the scanner cannot read a raw file, then see if you can import that raw file into the terrain itself, and then the terrain instead.

NOTE: If scanning images (Texture2D), or Meshes, then they must have previously been imported into Unity and Unity must be able to successfully read them.

## Extension System (GX)

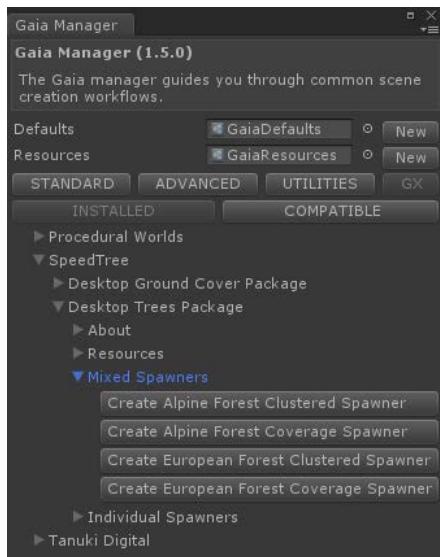
The Gaia eXtension (GX) system allows Gaia to be extended to easily leverage and take advantage of other art and tool based assets.



It consists of two tabs – an INSTALLED tab, and a COMPATIBLE tab.

### Installed Extensions

When an asset is installed that is Gaia Compatible, and has the correct integration script included, it will be exposed in the INSTALLED tab, and provides a number of menu options that can make leveraging those assets a simple and convenient.



For example – in this SpeedTree integration, add and leverage the selected SpeedTree assets is a simple one operation. Click once to add a preconfigured spawner, and click Spawn on that spawner to add the trees to the terrain and spawn them.



In another example Gaia comes preconfigured with some defaults for lighting and camera effects. Adding these effects into your scene is a few clicks rather than the fairly tedious and normal way that they would be set up.

## Compatible Extensions

Compatible extension are extensions that are known to be of high quality and are compatible with Gaia.

They provide a bit of information about the resource, and a link to the resources web page to make accessing it easier.



Asset producers who are interested in creating Gaia eXtensions should contact me directly. It's a great cross promotion opportunity and delivers better outcomes for everyone.

## Extension Exporter

You can create your own extensions with the extension exporter.

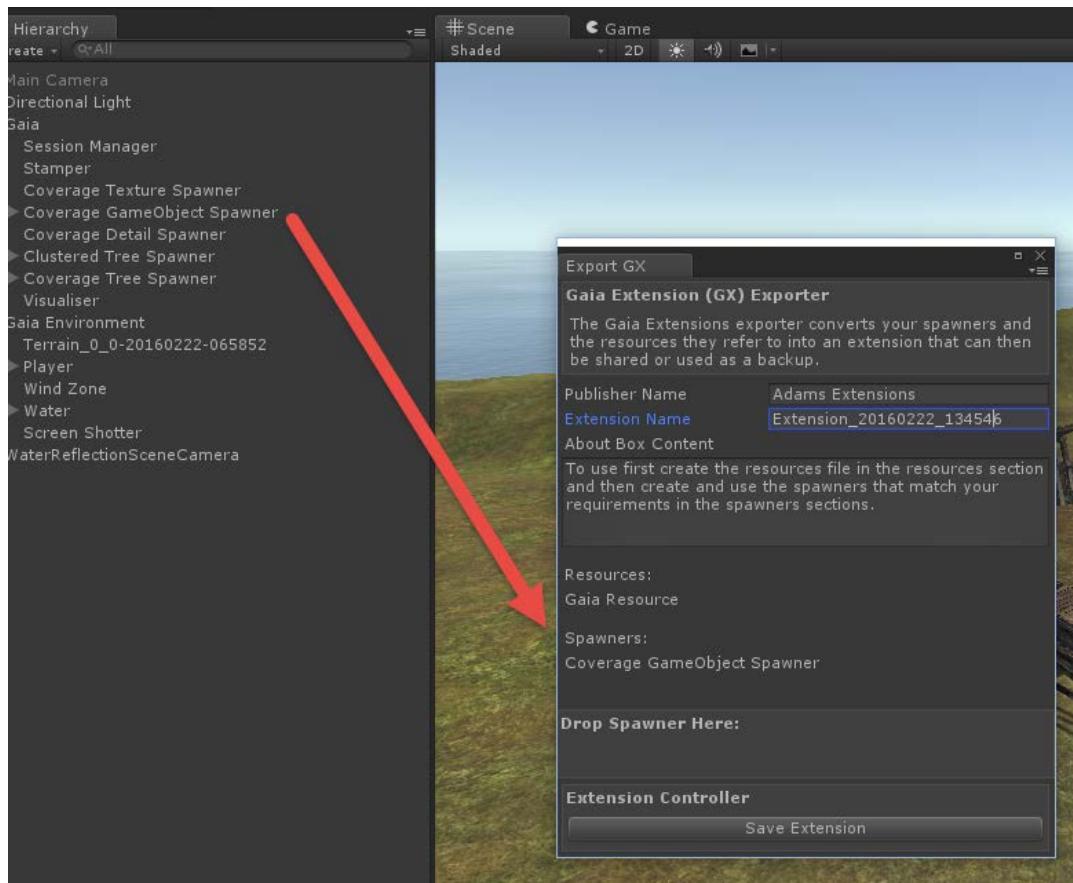
To access the extension exporter select Show Extension Exporter from the Utilities Menu.

You then drag and drop the spawners you wish to export into the exporter. The exporter will automatically pick up the related resources files.

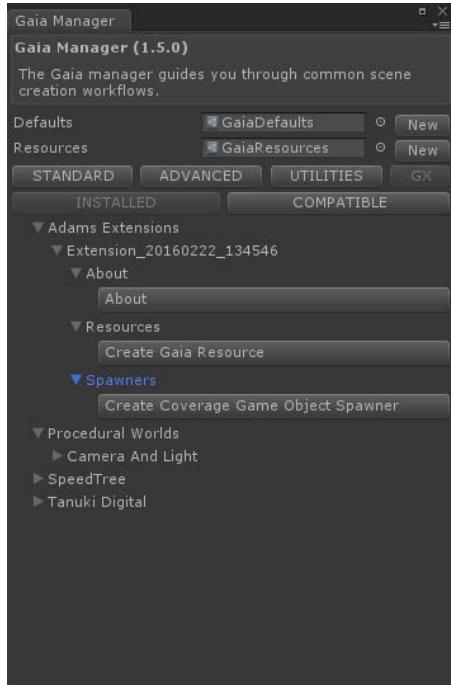
You then add in your publisher name and the extension name, and hit Save Extension.

The extension exporter will then generate a C# file in the GaiaExtensions directory that will allow you to encapsulate those spawners, their configuration, and their resource contents into a simple file that can easily be shared.

It is a great way of backing up and sharing valuable resource and spawner configurations.



When the extension has finished compiling it will now be available as a new Gaia Extension and can be dragged and dropped into any project. One of the great things about this is that it will automatically pick up the references to the original assets in the target project – even if they are in a different directory.



## Extensions API For 3<sup>rd</sup> Party Developers

The extension system works using .Net reflection. It consists of two scripts – an INSTALLED script and a COMPATIBLE script.

The INSTALLED script is included in the target assets project, and will only be made available to the end user when that asset is installed into a project that also has Gaia installed in it.

The COMPATIBLE script is included in Gaia, and used by Gaia to show off a directory of assets along with information and links back to the publishers web site. I will be paying close attention to what is included here and will only consider assets that are somehow synergistic with the process of environmental generation.

### Installed Script Structure:

Gaia provides two examples of different ways to generate an installed script – the first is via the extension exporter – and is an example of how to do art extensions – and the second is via the Camera And Light script and is an example of how to do tools extensions.

Regardless of how they are generated, the same basic structure is followed :

First – wrap your script in a define to ensure that the script is only compiled if Gaia is present and we are in the Unity editor. This stops asset store submission fails, and also run time build fails.

```
#if GAIA_PRESENT && UNITY_EDITOR  
...  
#endif
```

Next – in order to be detected the script needs to be appropriately namespaced as Gaia will scan the project for these specific name spaces. This scan happens when the Gaia Manager is opened, and whenever a build is detected.

The namespace should be Gaia.GX.YourPublisherName. Anything with Gaia.GX. is considered as an installed extension.

```
namespace Gaia.GX.ProceduralWorlds
```

Next name your class after your asset name.

```
public class CameraAndLight : MonoBehaviour
```

Next – implement the GetPublisherName() and get PackageName() methods. The content of these methods is used to configure your publisher and package names in the Gaia menu.

Finally to get a button to display in the installed extension menu, create a public static void method, and prefix it with GX\_. The button will take the following Camel Text format and break the words up for the button.

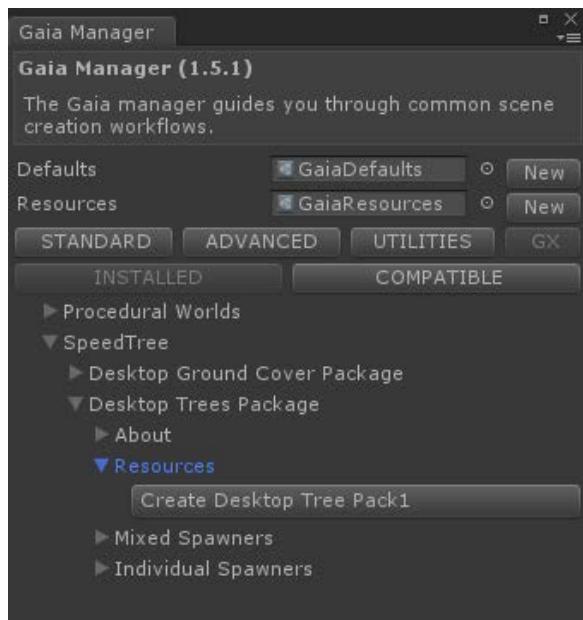
For example the following will generate an About button at the top level:

```
public static void GX_About()
```



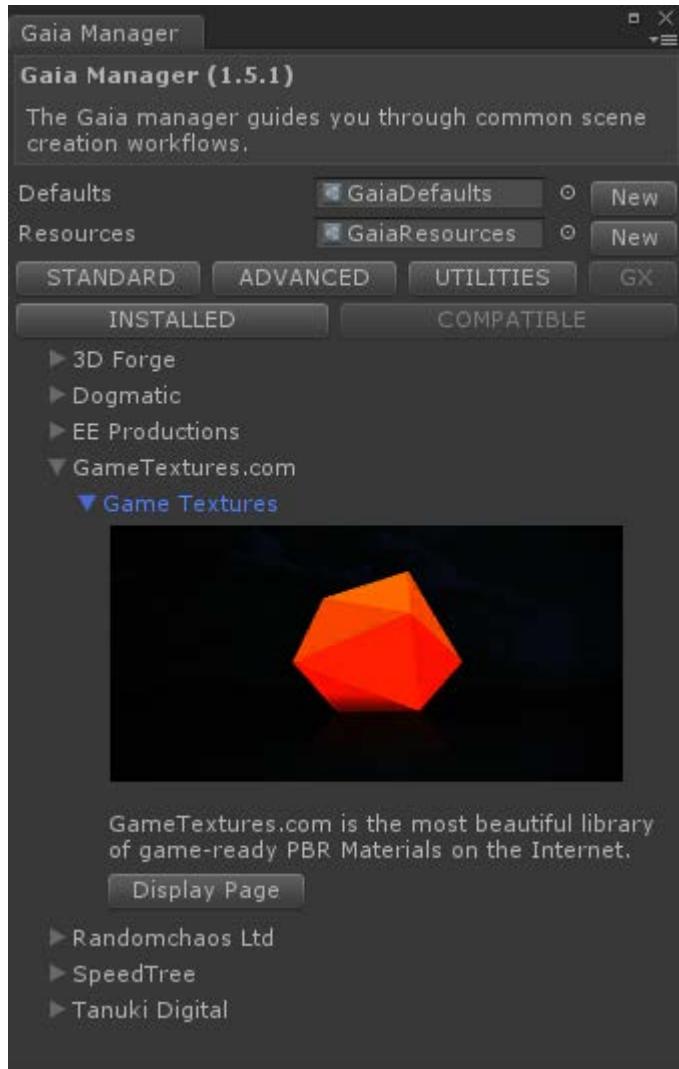
If you would like to create sub categories then add another underscore into the name of the method. The format is GX\_SectionName\_ButtonName.

```
public static void GX_Resources_CreateDesktopTreePackage1()
```



## Compatible Script Structure:

From a scripting perspective a compatible script is similar but slightly different to support its more informational nature.



The script provides a publisher name and package name which must match the installed script described in the previous description.

It then needs an image (optional), a description and a URL.

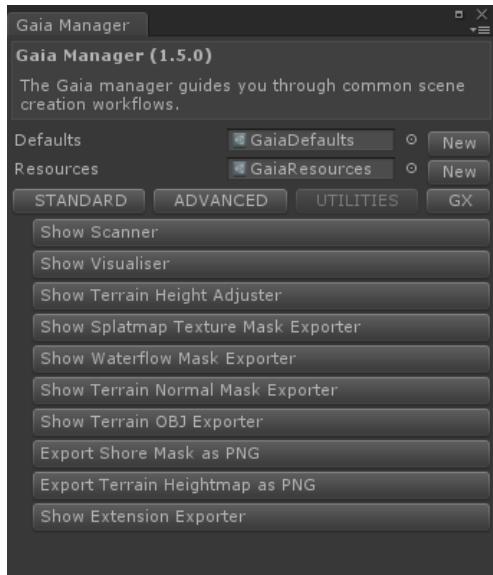
In order to be added as a compatible asset into Gaia it needs to be a quality asset and synergistic to the environmental experience. I will also get involved to ensure that the integration is tested and that a decent experience is provided to the end user. The benefit to the publisher is that the end user can take advantage of the full system in an easy way, and the publisher can benefit from the Gaia user base.

If you would like to be considered then please contact me adam@btek.com.au.

# Utilities

Gaia provides a number of handy utilities to make using the system easier.

To access Utilities open up Gaia Manager, and then select the Utilities menu.



## Show Scanner

Shows the scanner object.

## Show Visualiser

Shows the visualiser object.

## Show Terrain Height Adjuster

Shows the terrain height adjuster object. Allows you to change the height of the terrain to a physical value.

## Show Splatmap Texture Mask Exporter

Shows the splatmap texture mask exporter. This allows you to create a texture mask from the specified texture layer or all layers.

## Show Grassmap Texture Mask Exporter

Shows the grassmap texture mask exporter. This allows you to create a texture mask with one channel per grass layer in your scene.

## Show Waterflow Mask Exporter

Shows the waterflow mask exporter. This exports a mask based on where water would flow in the terrain. Very nice when used to do additional masked refinements to texturing.

### **Show Terrain Normal Exporter**

Shows the terrain normal exporter. Allows you to export a normal map of the terrain.

### **Show Terrain OBJ Exporter**

Shows the terrain obj file exporter. This allows you to export your terrain as an OBJ mesh file.

### **Export Shore Mask as PNG**

Exports a shoreline mask as a PNG file and puts it in your Assets directory.

### **Export Terrain Heightmap as PNG**

Exports the terrain height map as a PNG file and puts it in your Assets directory.

### **Show Extension Exporter**

Shows the extension exporter that allows you to export resource and spawner configurations as new extensions.

## Support

To get support for Gaia go to:  
<https://proceduralworlds.freshdesk.com/support/home>