

USER MANUAL



For the latest updated documentation go to www.awesometech.no

Vegetation Studio

Table of content

Features.....	3
Getting started.....	11
Vegetation Studio Manager.....	22
Vegetation Package.....	26
VegetationSystem Component.....	30
Settings tab.....	31
Render tab.....	37
Editor tab.....	44
Vegetation tab.....	47
Terrain textures tab.....	67
Mask tab.....	70
Real-time Mask tab.....	73
Debug tab.....	75
TerrainSystem Component.....	77
Vegetation Masks.....	84
VegetationMaskArea Component.....	85
VegetationMaskLine Component.....	92
VegetationBeacon Component.....	101
ColliderSystem Component.....	106
TouchReactSystem Component.....	113
BillboardSystem Component.....	123
RuntimePrefabSpawner Component.....	137
Persistent Vegetation storage.....	143
Settings Tab.....	145
Stored Vegetation Tab.....	146
Bake Vegetation Tab.....	148
Edit Vegetation Tab.....	149
Paint Vegetation Tab.....	151
Precision Painting Tab.....	154
Import Tab.....	157
Terrain Tree Importer.....	158
Terrain Detail Importer.....	159
Scene Vegetation Importer.....	160
Persistent Vegetation storage package.....	162
GrassPatchGenerator Component.....	163
VegetationColorMaskCreator Component.....	171
MaskBackgroundCreator Component.....	175
SceneVegetationBaker Component.....	177
ObstacleMaskCreator Component.....	179
Guide - Vegetation color mask and microsplat.....	181
Guide - Custom grass/plants.....	190
Guide - Map magic infinite terrain.....	195

VEGETATION STUDIO FEATURES

We are happy to introduce Vegetation Studio, coming to Unity Asset Store this fall.

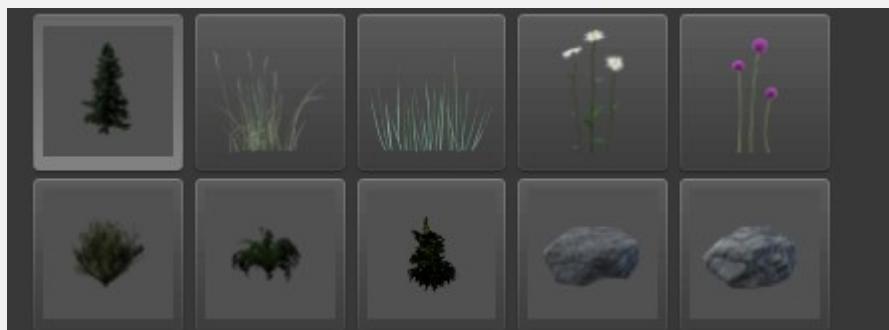
Vegetation Studio is a vegetation placement and rendering system designed to replace the standard tree and detail system in the Unity terrain component. Vegetation is spawned on the terrain based on a flexible set of rules, and controlled with both texture and polygon masks.

Use your existing trees (SpeedTree, Tree Creator etc.), rocks and grass textures. Rendering is done using Unity's new Instancing and Instancing Indirect system. This means no overhead on culling and handling GameObjects. Vegetation Studio should work with most vegetation shaders that support instancing.

You create a Vegetation Package that references your selected trees and plants. Configure the spawn rules and apply this to a terrain. The same package can be re-used on terrains in multiple scenes. Any changes done to the rule set will update the terrain directly.

With Vegetation Studio you can use denser vegetation on much larger terrains than before. Visible range, not terrain size, will limit your performance.

RULE BASED VEGETATION



Seed	67
Sample distance(meter)	16.2547
Density	1
Randomize distribution	<input checked="" type="checkbox"/>
Random distance	1
Collision Detection	<input type="checkbox"/>
Position offset	X 0 Y 0 Z 0
▼ Scale/Rotation	
Scale type	Simple
Min/Max scale	0.8 1.2
Rotation	Rotate Y
Rotation offset	X 0 Y 0 Z 0
▼ Height/Steepness	
Use height level	<input checked="" type="checkbox"/>
Selection type	Simple
Min/Max height	0 1000
Use steepness cutoff	<input checked="" type="checkbox"/>
Selection type	Simple
Min/Max steepness	0 25
▼ Perlin noise	
Use perlin noise	<input checked="" type="checkbox"/>
Perlin noise cutoff	0.57740
Perlin noise scale	9
Perlin noise offset	X 0 Y 0
Inverse perlin noise	<input type="checkbox"/>

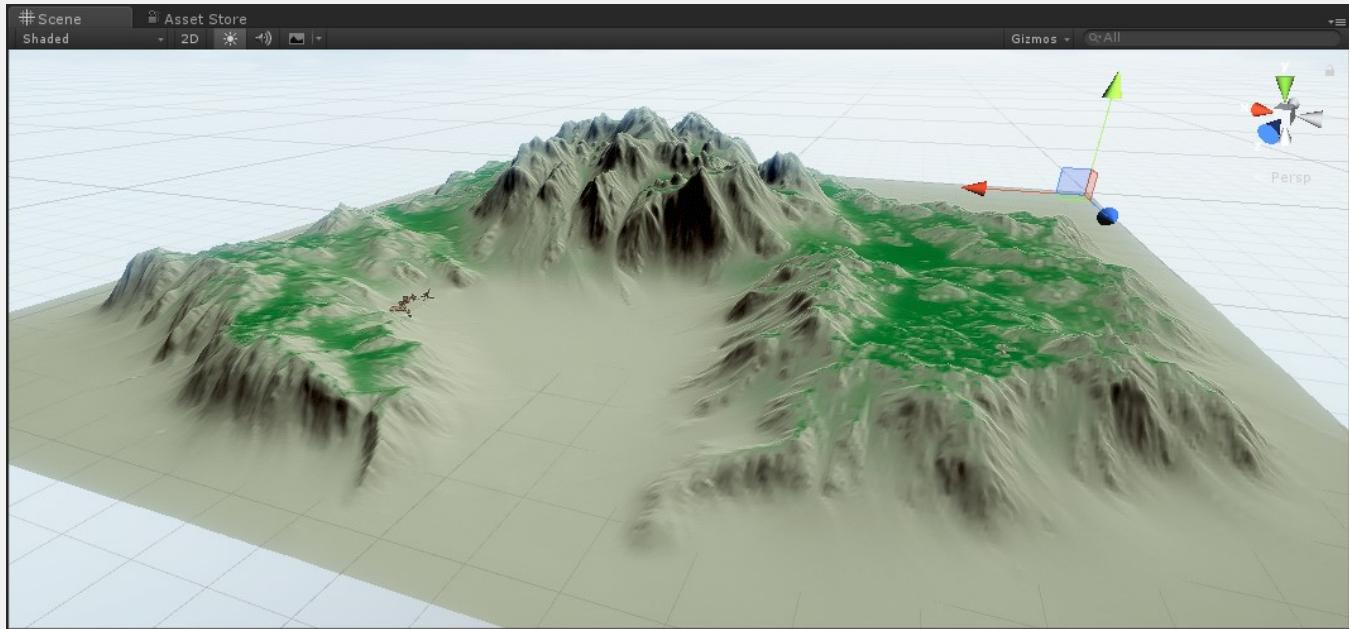


Image shows a distribution map for the spawning of a tree type. Area is based on a height and steepness curve.

Vegetation Studio uses a system for rule based vegetation. You add your grass, plants and trees to the system and they are placed according to your parameters. Configure the rules and the vegetation will spawn on the terrain accordingly. Parameters for vegetation growth are things like height above ocean level, steepness of the terrain, Perlin noise, rotation, scale and texture-based rules like "Do not grow grass on path texture", "only spawn this plant on sand area", etc. In addition to this, the run-time masking system controls the removal or spawning of any vegetation. Any changes are updated in the terrain directly.

See **VegetationSystem** component for more info.

EXTENSIVE RUN-TIME MASKING SYSTEM

Vegetation masks are a set of Components designed to control vegetation. These can be used both during design and run-time. By adding areas with polygons or lines with a width you can remove, add or modify vegetation within the area. Common use cases are roads, houses, city areas etc. See **VegetationMaskArea** and **VegetationMaskLine** for more info.





INSTANCED RENDERING

Vegetation Studio uses Unity's new Instanced and InstancedIndirect system to render the vegetation. Combined with an internal cell structure and the Culling Group API we are able to render large numbers of grasses, plants, trees and rocks with fewer draw calls and better speed than with normal rendering. In addition to this we do not have the overhead of processing and culling individual GameObjects.

RULE BASED TERRAIN SPLAT MAP

Vegetation Studio includes an optional system for splat map generation of the terrain. You can set up a rule set based on any number of textures in your terrain. Based on height, steepness and other parameters it will generate your terrain splat map. In addition to this it allows for live editor updates and can update the splat map and vegetation as you edit your terrain in the terrain inspector.

The textures not used for automatic splat map generation can be used to paint the terrain normally and will be kept even if you change the rule set and generate the splat map again. This way the manual work you do painting roads or other features will not be affected.

See **TerrainSystem** for more details.



TOUCH BEND GRASS



GRASS PATCH EDITOR

Grass Patch Generator is a tool to produce mesh grass and plant mesh patches with level of detail (LOD) from standard Texture2D images with alpha.

Size, resolution and a range of settings will be saved with the prefab. The resulting prefab can be

Vegetation Studio

used with Vegetation Studio as instanced indirect rendered grass with Touch Bend support. Or used as normal GameObject in Unity with a LOD Group.

See **Grass Patch Generator** for more info.



UNIVERSAL BILLBOARD SYSTEM

Vegetation Studio includes a billboard system that can create a billboard on any tree model added to the VegetationSystem Component. These are used for long range when displaying trees in the terrain. Billboard creation is automatic when adding a tree prefab in editor mode. See **BillboardSystem** for more details.

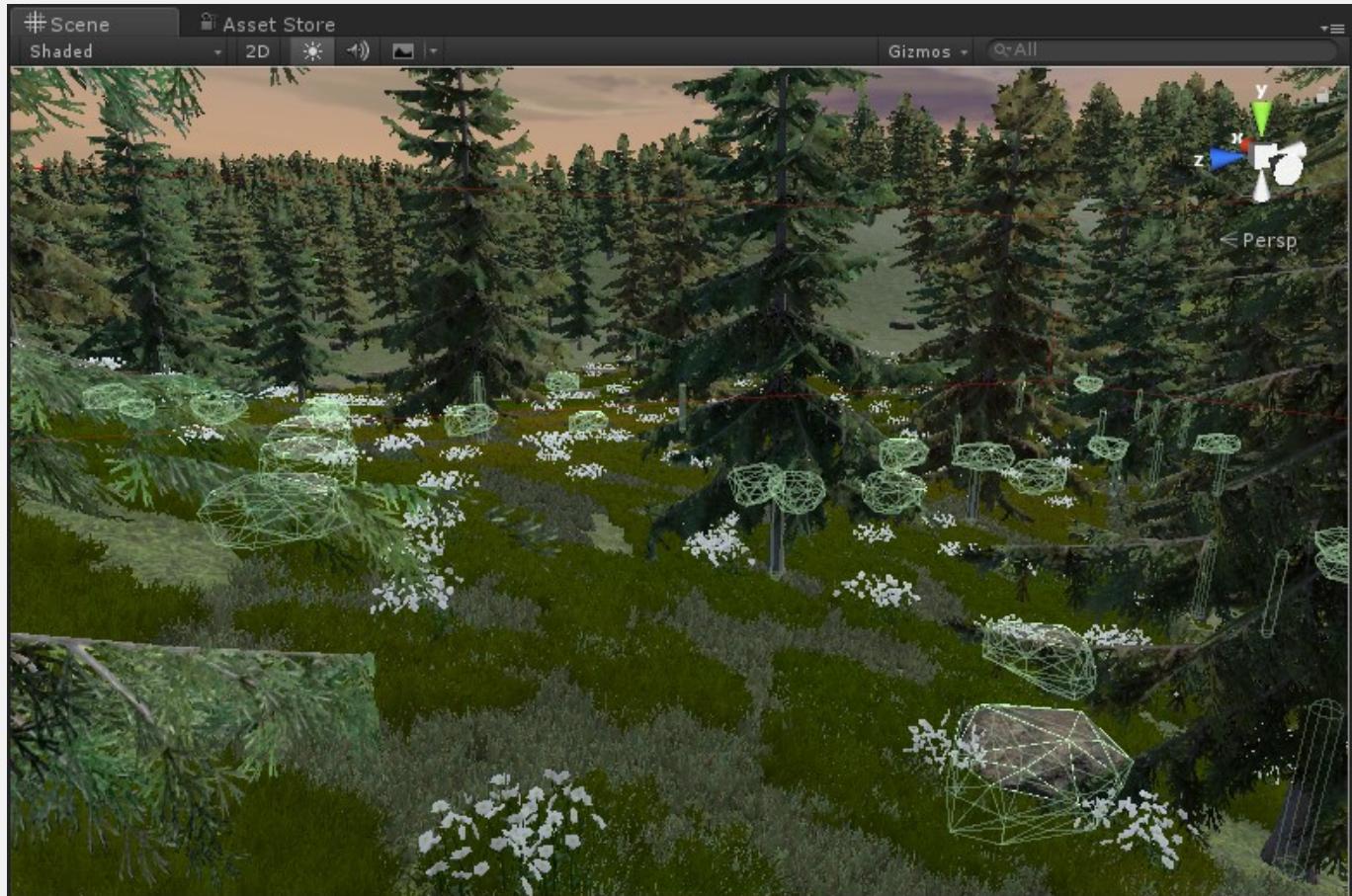




COLLIDER SYSTEM

Vegetation Studio has a built in run-time collider system. When set up it will instantiate colliders for trees, rocks etc in proximity of the camera. These colliders are instantiated from a pool to increase performance. See **ColliderSystem** for more info.

There is an API with event calls for access to the run-time colliders.



REAL-TIME EDITOR UPDATES

Any changes to the Vegetation Packages will be updated directly in the Terrain. In addition to this, changes you do with the Terrain Inspector will affect vegetation as you make them. The splat map can be updated from rules, and vegetation changes as you paint texture details. All based on the ruleset you have configured in the Vegetation Package.



MULTIPLE TERRAINS/CAMERAS

Vegetation Studio has support for multiple cameras and terrains. To add multiple terrains to the scene just include a `VegetationSystem` object for each terrain. Assign the `VegetationPackage` and

terrain, and it will work. Use the same approach for multiple cameras.

There is also floating origin support for large terrains.

VR-SUPPORT

Vegetation Studio supports VR rendering. The custom billboards system supports single pass VR and does not have the strange rotation and bend behavior of some other billboards.

SUPPORTS SPEEDTREE, TREE CREATOR AND OTHERS...

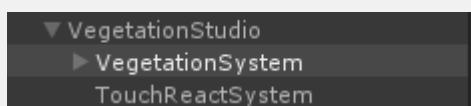
Vegetation Studio is designed to work with your existing vegetation assets. Except for those used in the demo scene, the asset does not include Vegetation meshes. You can use your normal vegetation meshes and textures. Most vegetation shaders that support instancing will work directly with Vegetation Studio.

GET STARTED WITH VEGETATION STUDIO

This guide will explain how to do your first setup of Vegetation Studio in your scene.

First you need to set up an instance of the Vegetation Studio Components. To add this select the menu “Window/Awesome Technologies/Add Vegetation Studio to scene”

This will add the needed objects to the open scene.



The system has three main GameObjects. VegetationStudio has a **Vegetation Studio Manager** Component. There can be only one of these in the scene. The **VegetationSystem** GameObject holds four components and is the object you will configure in this getting started guide.

When adding Vegetation Studio to the scene it will do some configuration on the terrain. Turn off Unity grass and tree rendering, minimum setting on pixel error etc. To override these settings disable this update on the settings tab of VegetationSystem component.

Next step is to create a Vegetation Package. This is a scriptable object that will save all your vegetation rules and settings. Create a new folder to save the object in. It is recommended to place this folder outside of the “AwesomeTechnologies” folder as it makes upgrading to a new version easier.

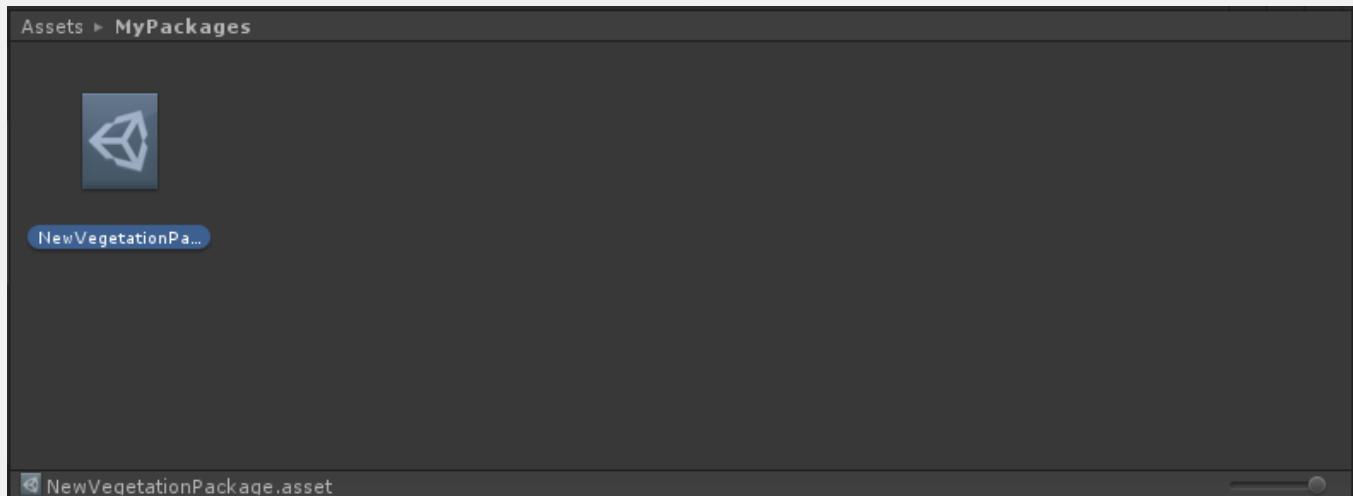
To create the Vegetation Package right click in a folder in project view and select “Create/Awesome Technologies/Create Vegetation Package”.

If you want Vegetation Studio to handle terrain textures and generate splat map from them, select the number of textures you would like to use on the terrain. See **TerrainSystem** Component for more info on splat map generation. If you have a vegetation package with textures it will replace the textures on the terrain (keeping the splat maps) when you assign it. This is done to be able to have a summer and winter package. etc. If you have a 3rd party shader like MicroSplat or CTS create a package with no textures.

For now, select “no textures”.

Name the package what you want.

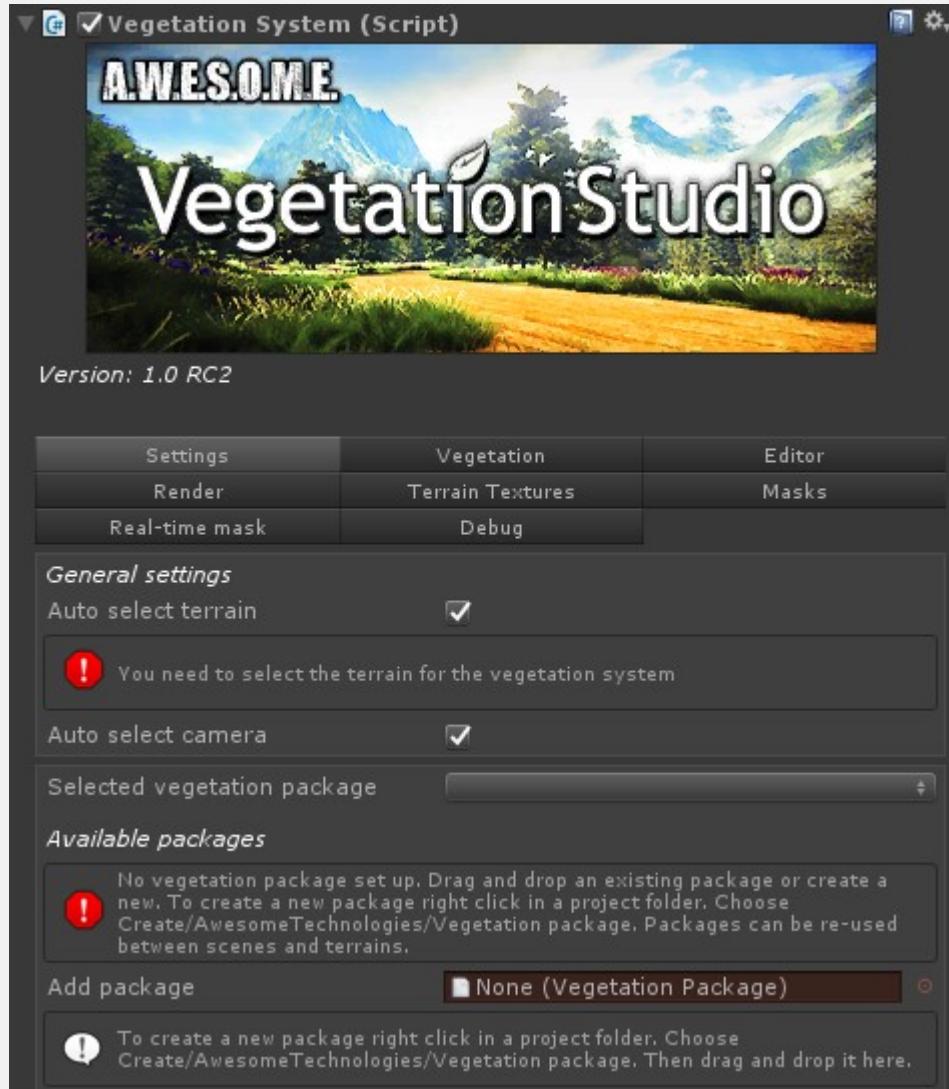
Vegetation Studio



Select the new package and also give it a package name in the inspector.



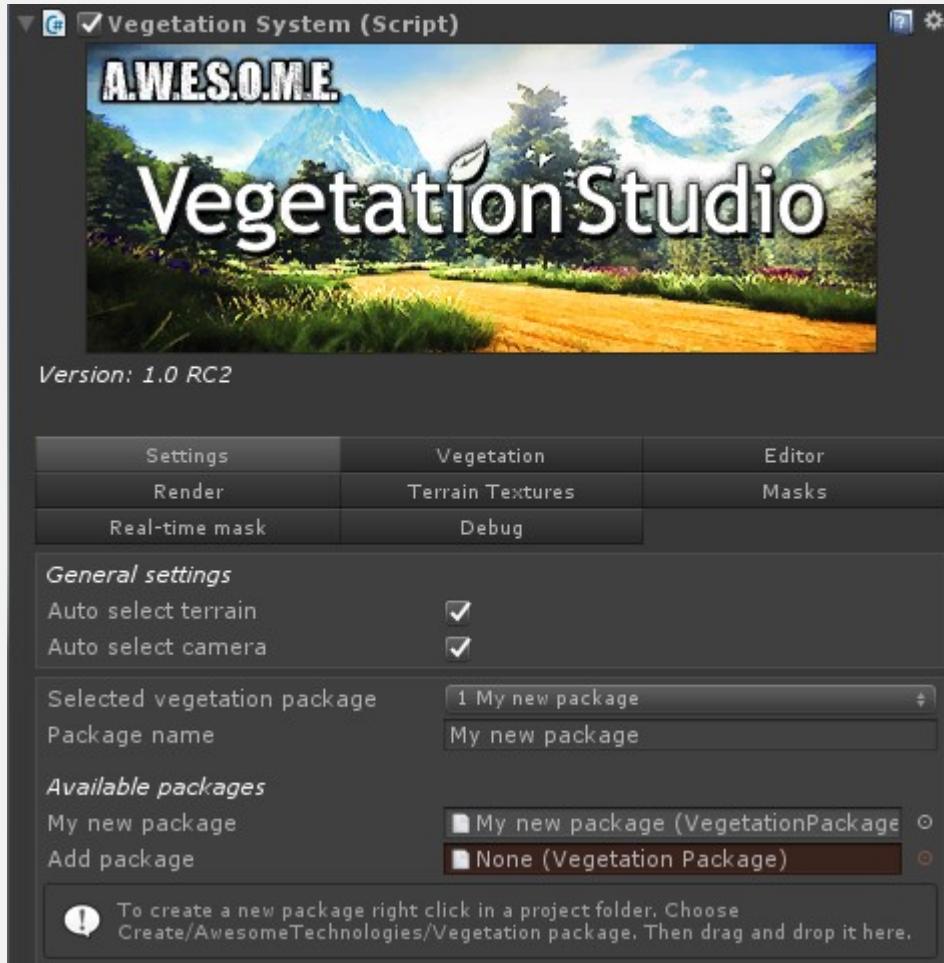
Next select the VegetationSystem GameObject and find the Vegetation Studio Component.



Add a terrain to the scene if you get a missing terrain error.

You will see it has an error for missing vegetation package. Drag and Drop your newly created vegetation package to the "Add package" field.

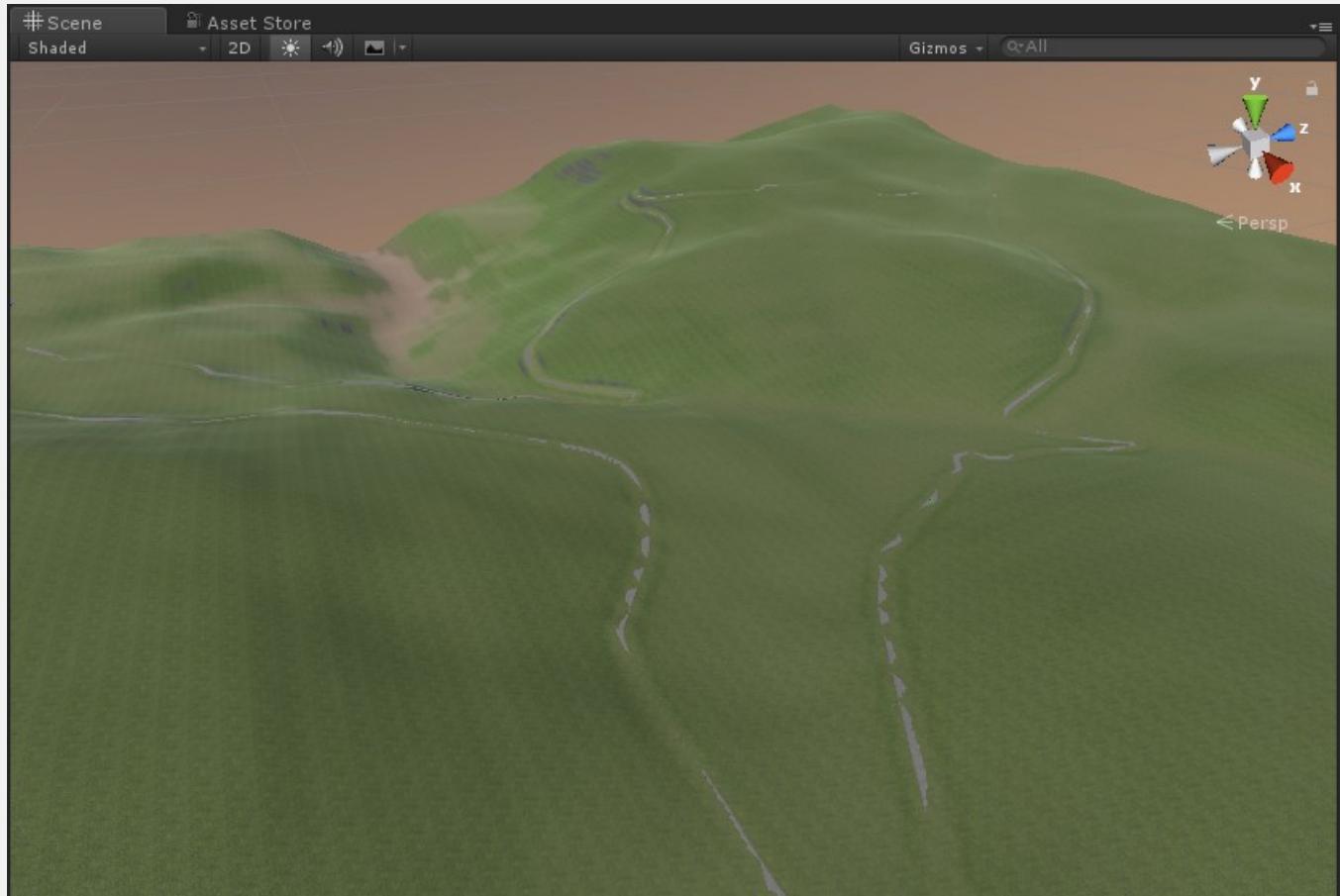
Vegetation Studio



By default Vegetation System will disable unity trees and details on the terrain. Nothing is deleted. You can enable this again if you want.

You will now see your new package listed under available packages and selected. It is possible to have several packages added and switch between them in code or inspector.

As this package is empty and has no rules you will not see any vegetation on your terrain.



Terrain with the new empty vegetation package

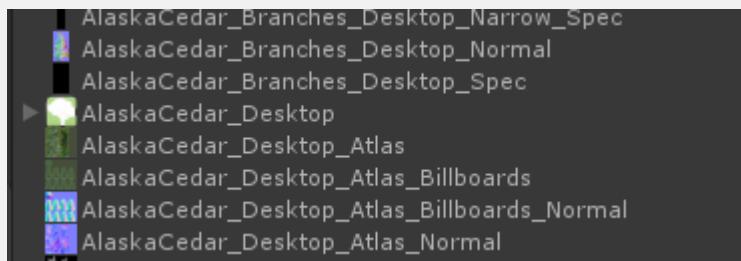
To start adding vegetation rules to the package select the vegetation tab of the Vegetation Studio Component.

Vegetation Studio



Find a vegetation prefab to add to the system. In this case we are adding a speedtree Alaska Cedar model.

Vegetation Studio

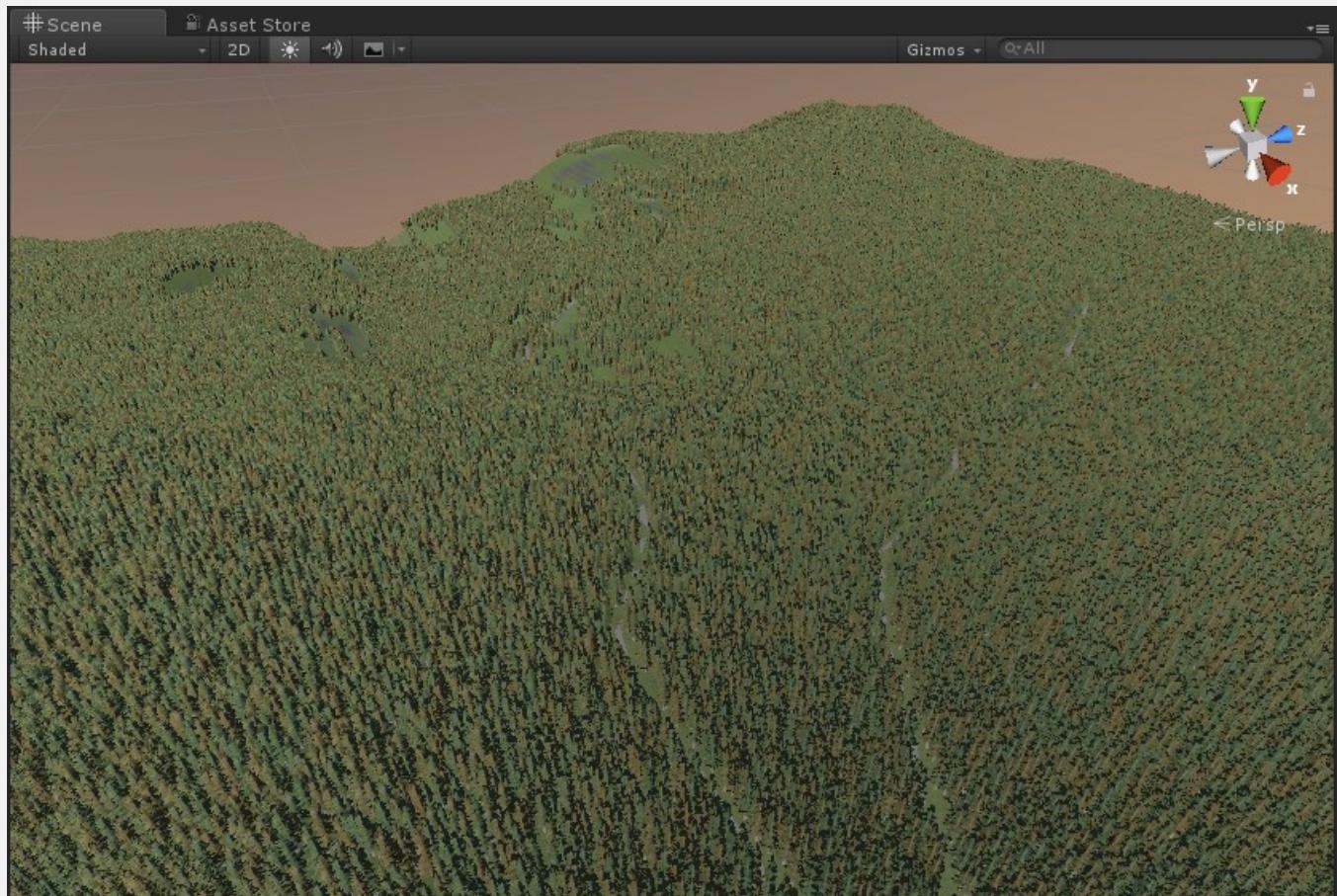


Drag and drop this to the “Add tree prefab” drop box in the Vegetation Tab. This will create a new default tree rule.

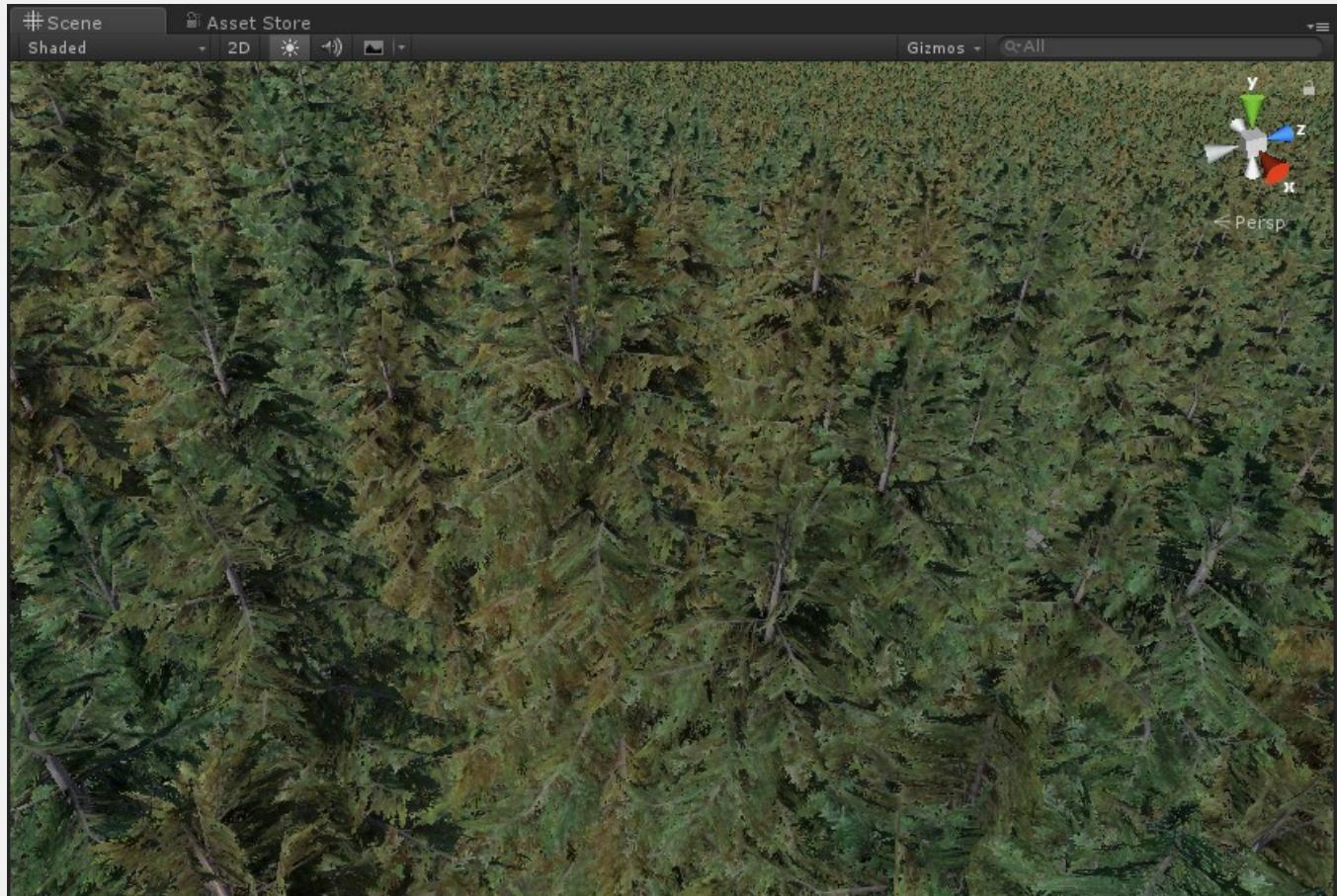


The terrain will populate with trees based on the default rules.

Vegetation Studio



Overview of the terrain with a mix of mesh trees and billboards



Closer view of the added trees

Next step is to adjust these rules. Add more trees, plants or grass.

Select Vegetation Item



Visualize height/stEEPNESS in terrain

Show heatmap

! Enabling heatmap will show the spawn area of the selected vegetation item based on the current height and steepness setting for that item.

Item Name	Broadleaf/Desktop
Type: Tree	
Vegetation prefab	<input type="text" value="Broadleaf_"/> <input type="radio"/> Refresh prefab
Selected initial LOD	<input type="text" value="LOD0"/>
Render mode	<input type="text" value="Instanced"/>
Enable run-time spawn	<input checked="" type="checkbox"/>

► Billboards
 ► Colliders
 ► NavMesh Obstacle
 ► Shadows
 ► LODs
 ► Speedtree
 ▼ Position

Seed	<input type="text" value="67"/>
Sample distance(meter)	<input type="text" value="16.2547"/>
Density	<input type="text" value="1"/>
Randomize distribution	<input checked="" type="checkbox"/>
Random distance	<input type="text" value="1"/>
Collision Detection	<input type="checkbox"/>
Position offset	X <input type="text" value="0"/> Y <input type="text" value="0"/> Z <input type="text" value="0"/>

▼ Scale/Rotation

Scale type	<input type="text" value="Simple"/>
Min/Max scale	0.8 <input type="text" value="1.2"/>
Rotation	<input type="text" value="Rotate Y"/>
Rotation offset	X <input type="text" value="0"/> Y <input type="text" value="0"/> Z <input type="text" value="0"/>

▼ Height/StEEPNESS

Use height level	<input checked="" type="checkbox"/>
Selection type	<input type="text" value="Simple"/>
Min/Max height	0 <input type="text" value="1000"/>
Use steepness cutoff	<input checked="" type="checkbox"/>
Selection type	<input type="text" value="Simple"/>
Min/Max steepness	0 <input type="text" value="25"/>

Look in the documentation for the components to get detailed information on rules and settings.

VEGETATION STUDIO MANAGER

The Vegetation Studio Manager Component is a manager component designed to keep sync between all Vegetation System components in one scene. You would use multiple Vegetation System Components in scenarios with multiple terrains and/or cameras active.

It also has an API to control one or multiple Vegetation Systems run-time. Set new vegetation packages. Vegetation Density etc.

Adding and removing Vegetation Masks are also done using this component API.

To create a new Vegetation Studio Manager Component select “Window/AwesomeTechnologies/Add Vegetation Studio” from the menu in Unity or drag the “VegetationStudio” prefab from “AwesomeTechnologies/VegetationStudio/prefab” folder to the scene.

There should be only one instance of the Vegetation System Manager component per scene.

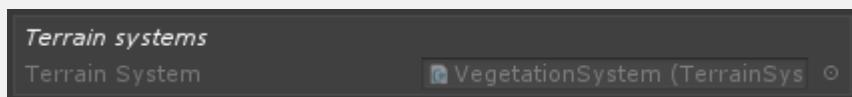
Vegetation Systems

Terrain Systems

Vegetation Masks

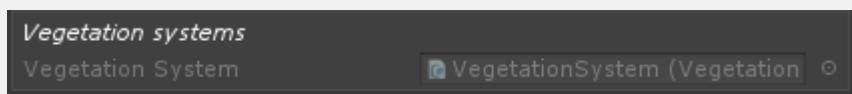


VEGETATION SYSTEMS



All Vegetation System components in the scene will auto register with the Vegetation Studio Manager Component and listed here.

TERRAIN SYSTEMS



All Terrain System components in the scene will auto register with the Vegetation Studio Manager Component and listed here.

VEGETATION MASKS

All Vegetation Masks in the scene will auto register with the Vegetation Studio Manager. The manager will then make sure all VegetationSystems have masks assigned and removed as needed.

SCRIPT API

VegetationStudioManager has an external API to refresh vegetation when Terrain Data is changed. It will manage multiple VegetationSystem and TerrainSystem components in the scene.

RefreshTerrainSplatMap is used to update vegetation if terrain splat map is changed from code or an editor tool. you can provide an optional bounds to only refresh splatmap in an area. This is much faster than a complete update

```
VegetationStudioManager.RefreshTerrainSplatMap(changedBounds);
```

RefreshTerrainHeightMap is used to update vegetation if terrain height map is changed from code or an editor tool. you can provide an optional bounds to only refresh height in an area. This is much faster than a complete update

```
VegetationStudioManager.RefreshTerrainHeightMap(changedBounds);
```

GenerateTerrainSplatMap is used generate the terrain splat map from rules configured in the TerrainSystemComponent. It takes an optional bounds as parameter and a ClearAllLayers bool. ClearAllLayers to true will clear all manually drawn textures in the splat map by terrain textures not used with automatic splat generation. See TerrainSystemComponent for more info

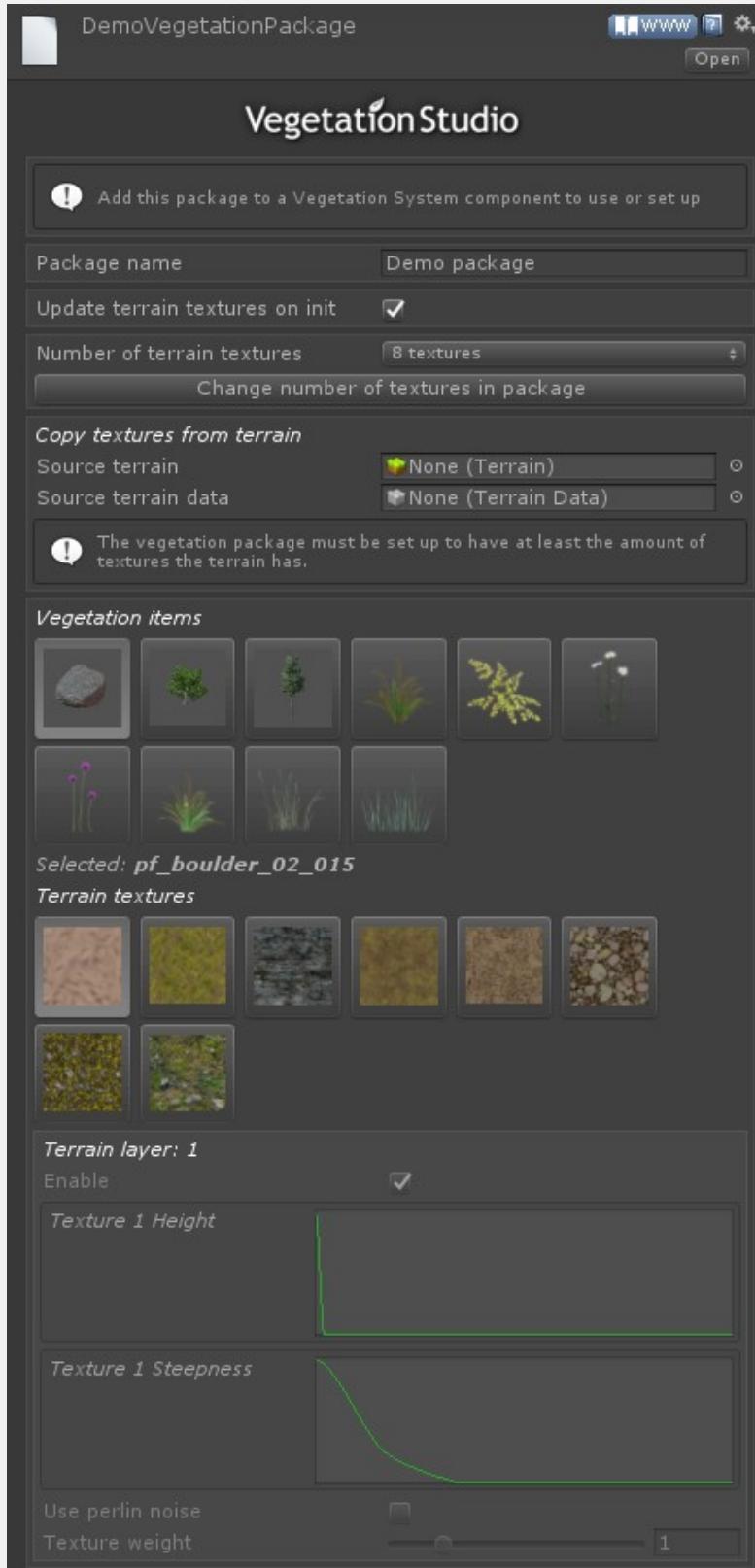
```
VegetationStudioManager.GenerateTerrainSplatMap(changedBounds, false);
```

VEGETATION PACKAGE

The Vegetation Package is a scriptable object that will save all your vegetation rules and settings. You create a package, add your grass, plants, trees and rocks and configure the spawning rules. It can also be used to manage terrain textures and splat map generation.

You can apply this package to any terrain or even multiple packages to the same terrain and switch between them run-time.

Vegetation Studio



PACKAGE NAME

Here you can set a name on the package. This is used for reference in the VegetationSystem inspector.

UPDATE TERRAIN ON INIT

With the update terrain on init checkbox enabled VegetationStudio will replace the textures in the terrain with the textures in the package. This is useful when you work with multiple vegetation packages. You can switch from a summer to a winter package and the terrain textures will change but keeping the same splatmap. It is important that you keep the summer and the winter road textures in the same index for the splats to look good.

NUMBER OF TERRAIN TEXTURES

If you create a package and want to change the number of terrain textures in the package you can do it here.

COPY TEXTURES FROM TERRAIN

If you create a vegetation package with textures and want Vegetation Studio to handle your terrain textures you can use this function to read the texture setup from an existing terrain. The package must be set to the same amount of textures as the terrain. Then drag and drop the terrain here. What happens is that the texture references from the terrain is added to the Vegetation Package.

CREATE A VEGETATION PACKAGE

Create a new folder to save the object in. It is recommended to place this folder outside of the "AwesomeTechnologies" folder as it makes upgrading to a new version easier.

To create the Vegetation Package right click in a folder in project view and select "Create/Awesome Technologies/Create Vegetation Package".

If you want Vegetation Studio to handle terrain textures and generate splat map from them, select the number of textures you would like to use on the terrain. 0,4,8,12 or 16 textures. See **TerrainSystem** Component for more info on splat map generation.

Name the package what you want.



Select the new package and also give it a package name in the inspector.

You can now assign the empty package to the VegetationSystem component. See **Settings** and **Vegetation** pages for more info on how to add and configure the package.

VEGETATION SYSTEM

VegetationSystem Component is the main component in Vegetation Studio. It handles the rules, spawning of vegetation and the main rendering loop.



The documentation for this component is divided into several pages.

Settings Tab

Render Tab

Editor Tab

Vegetation Tab

Terrain Textures Tab

Masks Tab

Real-time mask

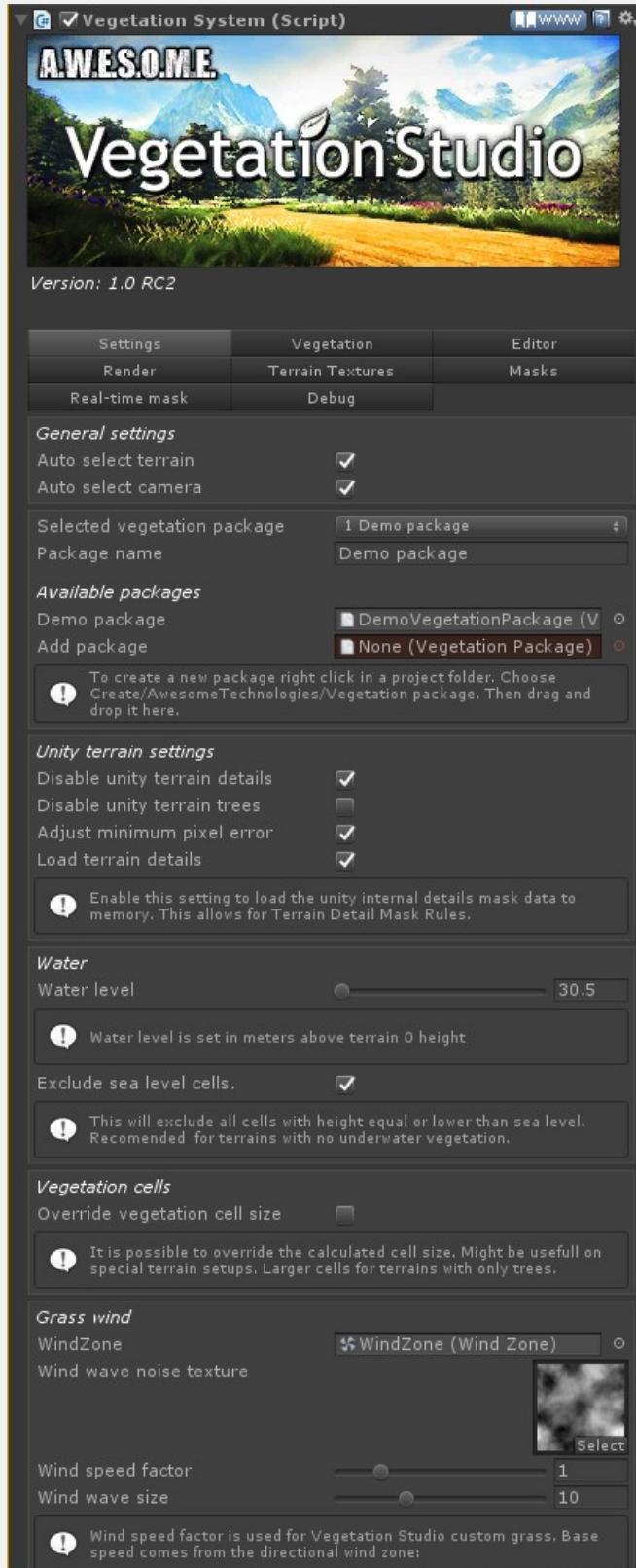
Debug Tab

VEGETATION SYSTEM - SETTINGS TAB

This section is used to configure the main settings of the VegetationSystem Component. It is here you set up references to camera, terrain. Add Vegetation Packages etc.

This page is part of the documentation of the **VegetationSystem** Component.

Vegetation Studio



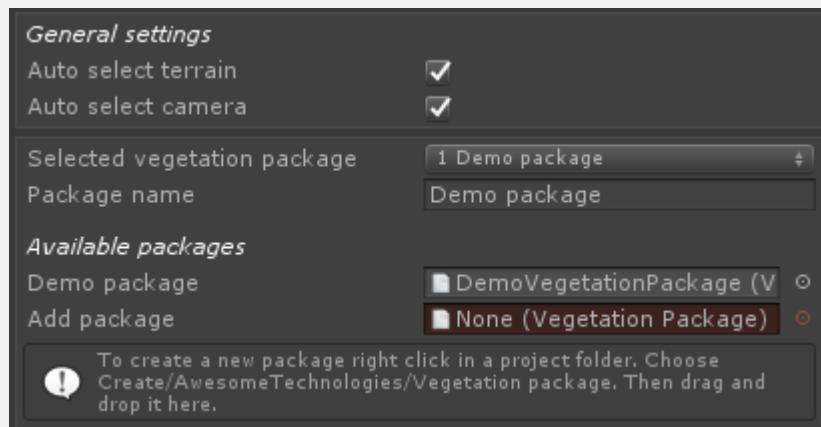
General Settings

Unity Terrain

Water

Grass Wind

GENERAL SETTINGS



AUTO SELECT TERRAIN

When enabled VegetationSystem component will use Terrain.ActiveTerrain as selected terrain. Disable to provide a terrain manually. For setups with multiple terrains and VegetationSystems this is needed.

AUTO SELECT CAMERA

When enabled VegetationSystem component will use Camera.Main (Camera tagged as MainCamera) as selected camera. Disable to provide a camera manually. For setups with multiple cameras this could be needed.

Run time created cameras will be detected if tagged as MainCamera.

SELECTED VEGETATION PACKAGE

This drop down box selects the active vegetation package. The active packages is the one you configure and use with this component.

Select from drop down list or set index from script to change.

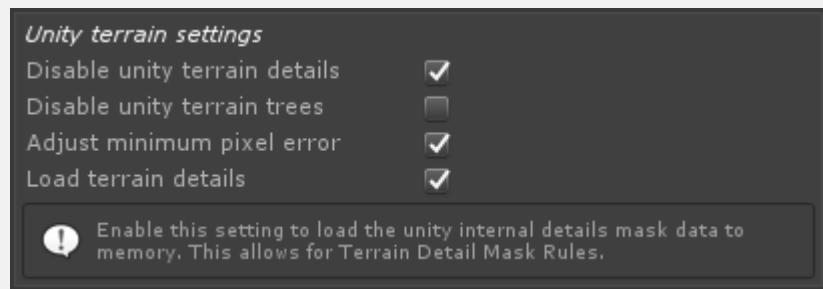
AVAILABLE PACKAGES

This is a list of all vegetation packages added to the VegetationSystem Component. To remove,

select one and press Backspace key.

To add a new package to the vegetation system drag and drop in the Add Package drop field.

UNITY TERRAIN



These settings are helper setting to configure the Unity terrain. With all vegetation in Vegetation Studio there is no need to have the tree or detail system on Unity Terrain enabled.

DISABLE UNITY TERRAIN DETAILS

Enable this to turn of unity terrain detail system at startup.

DISABLE UNITY TERRAIN TREES

Enable this to turn of unity terrain tree system at startup.

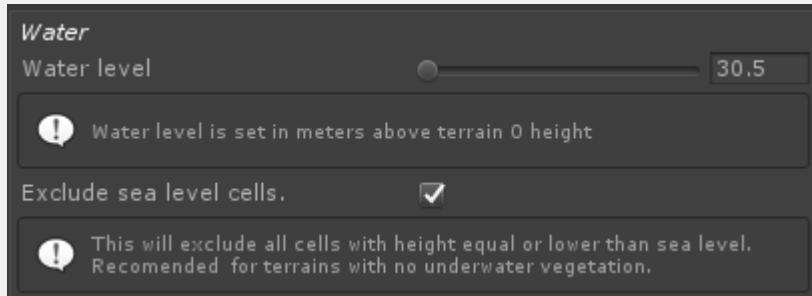
ADJUST MINIMUM PIXEL ERROR

With this setting enabled VegetationSystem will make sure the Pixel Error on the Unity terrain is at least 5. A really low pixel error will lead to a high amount of drawcalls for the terrain.

LOAD TERRAIN DETAILS

Load terrain details needs to be enabled in order to use Terrain Detail density rules for spawning. Disabled by default to save memory.

WATER



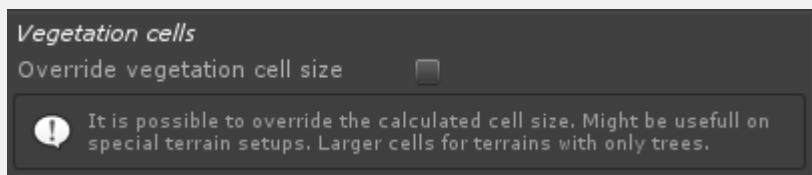
WATER LEVEL

This sets the sea level for the selected terrain. Value is in meters from terrain 0 height. This water level will be used as 0 height for spawning rules in Vegetation Tab and for splat map generation in TerrainSystem component.

EXCLUDE SEA LEVEL CELLS

for terrains with large areas of sea and no under water vegetation you can enable this setting. it will exclude the sea level(or lower) cells and speed up culling and spawning as they do not need to be evaluated.

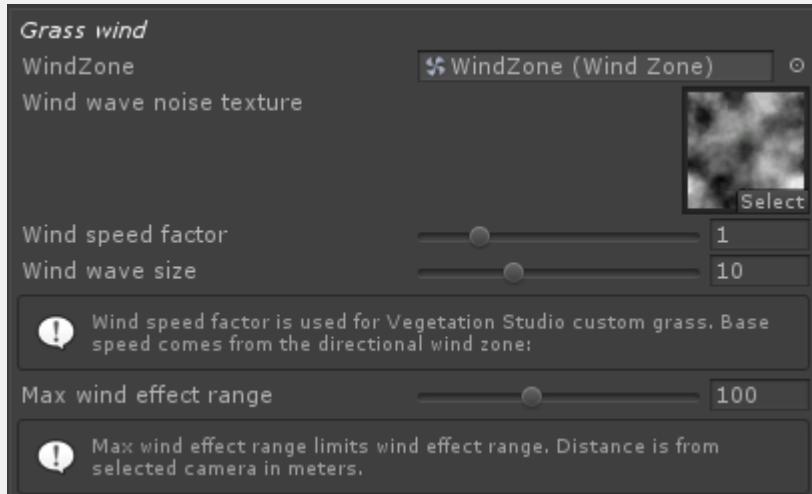
VEGETATION CELLS



OVERRIDE VEGETATION CELL SIZE

If you would like to override the vegetation cell size, enable this and set the new size. This setting will probably be removed after Beta.

GRASS WIND



WIND ZONE

VegetationSystem component will search and select the first directional Wind Zone it finds in the scene. If you have multiple and want to select what zone to use, drag and drop it here. If no wind zone is present there will be no wind for Vegetation Studio Grass.

WIND WAVE NOISE TEXTURE

Noise texture is used to calculate wind waves for Vegetation Studio Grass. It should be tilable to avoid artifacts in the wind movement.

WIND SPEED FACTOR

Wind speed factor is a multiplier for the wind speed on the selected wind zone.

WIND WAVE SIZE

This sets the scale in meters for the noise texture used to create wind waves.

LIMIT EFFECT RANGE

Checkbox to enable or disable wind effect range limit. When disabled wind will be active until max visible range.

MAX WIND EFFECT RANGE

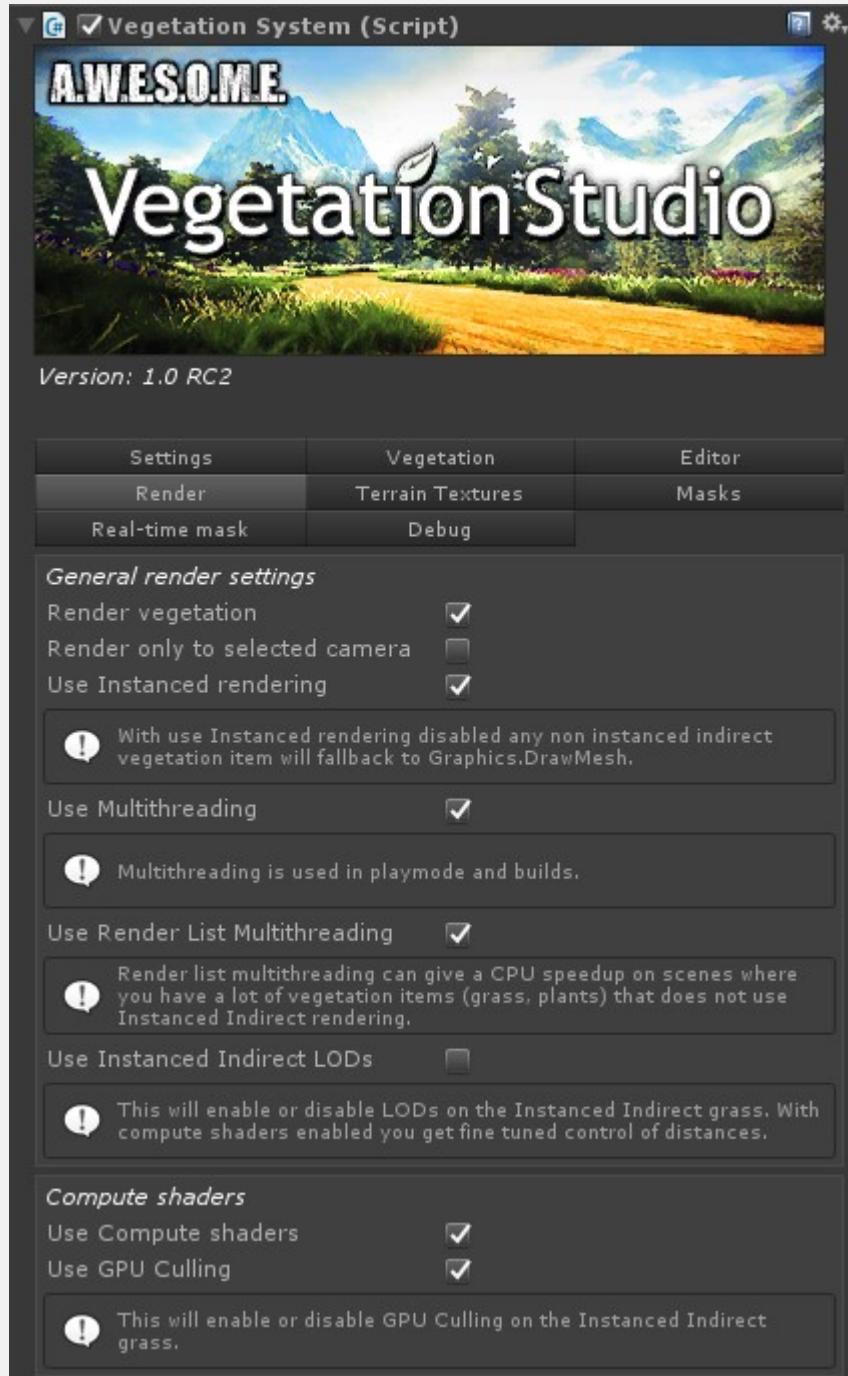
This setting sets the maximum range from camera for wind to affect grass.

VEGETATION SYSTEM - RENDER TAB

The Render Tab lets you control settings used in the render loop, layers, shadows, multithreading and more. Settings are saved in the scene on the VegetationStudio Component.

This page is part of the documentation of the **VegetationSystem** Component.

Vegetation Studio





General Render Settings

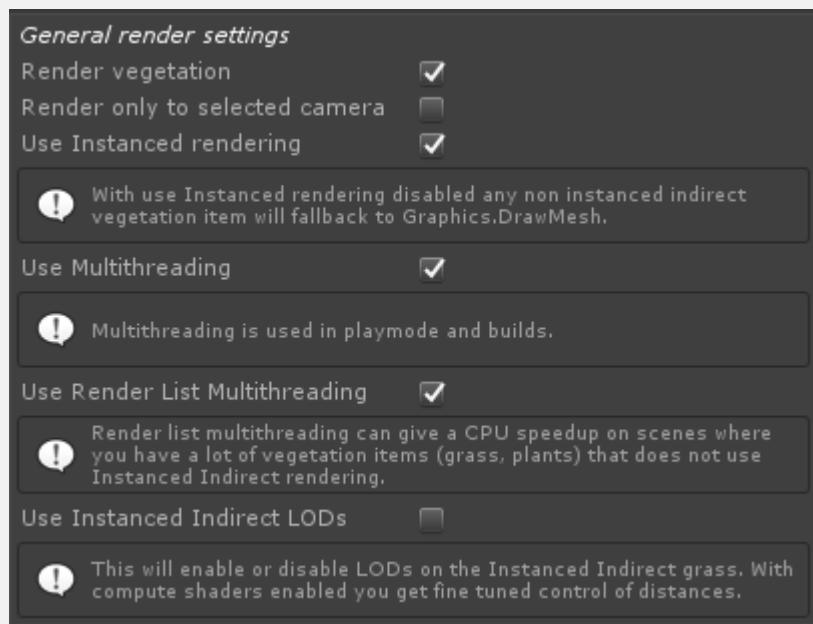
Vegetation Settings

Shadows

Editor Shadows

Layers

GENERAL RENDER SETTINGS



RENDER VEGETATION

The “Render vegetation” checkbox can enable or disable the render loop of Vegetation System. All culling, spawning of vegetation etc. will still be done

RENDER ONLY TO SELECTED CAMERA

When checked the vegetation will only be rendered to the selected camera for the VegetationStudio Component. This can be usefull if working with multiple camera setups that are watching the same areas. Binoculars looking into the same terrain etc.

USE INSTANCED RENDERING

Disabling “Use instanced rendering” will force the render loop of instanced shaders back to Graphics.DrawMesh. This is slower but can be usefull for debugging of shaders that has problem with instancing. This has no effect on Vegetation Items set to Instanced Indirect.

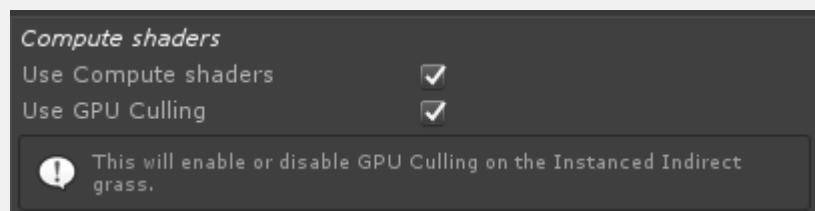
USE MULTITHREADING

when enabled multi threading will be used when spawning new areas and preparing render lists. This is only used in playmode and in builds. In editor mode single threading is used.

USE RENDER LIST MULTITHREADING

when enabled multi threading when creating new renderlists as camera rotates and visible cells change. This is only used in playmode and in builds. In editor mode single threading is used.

COMPUTE SHADERS



USE COMPUTE SHADERS

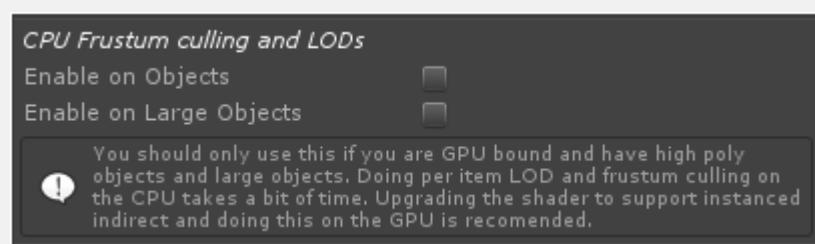
Enable this to use the compute shader renderloop for instanced indirect vegetation. This will merge all cell buffers and draw all items of a kind in a single drawcall.

Per item LODs are also calculated in the final compute shader before rendering.

USE GPU CULLING

With GPU culling enabled frustum culling will be done on the GPU in the final compute shader pass, reducing the number of instanced rendered to what is actually showing. No overhead for partial visible cells.

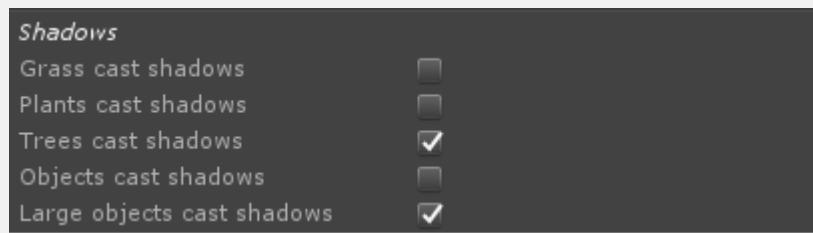
CPU FRUSTUM CULLING AND LODS



Enabling CPU frustum culling and LODs will do a 2nd level frustum culling on top of the normal cell culling. It also enables per item LODs on the objects and large objects. There is a compute cost for this and I recommend to only enable this if you have high poly objects or heavy shaders. To get individual LODs for high volume object count upgrade the shaders to Instanced indirect and do this in the compute shader loop.

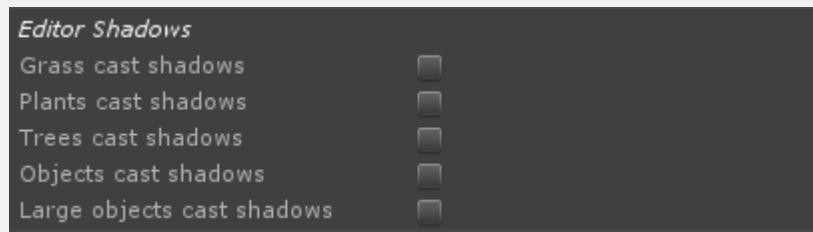
SHADOWS

Select if Vegetation Items will cast shadows or not in playmode. For grass and plants in deferred mode a full screen effect like Screen Space Shadows or Contacts Shadows is recommended as this is much faster.



EDITOR SHADOWS

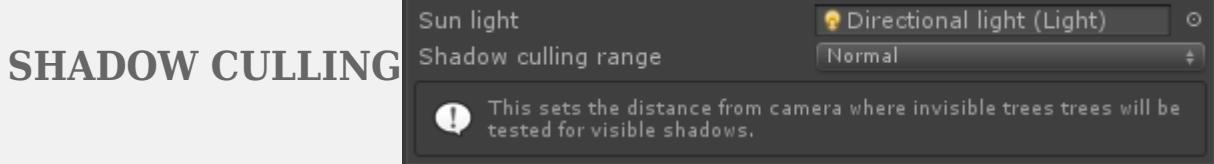
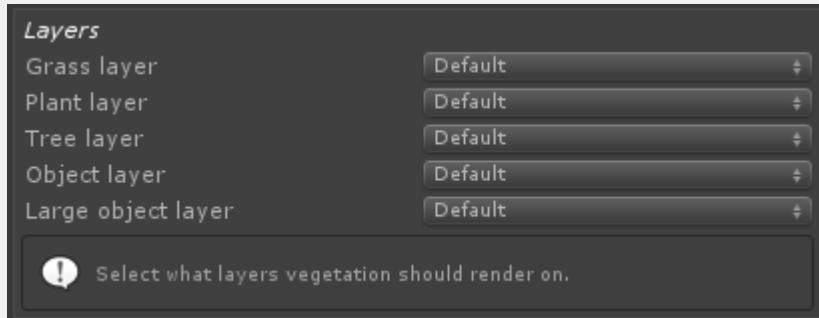
Same as Shadows option above but used for editor mode.



LAYERS

You can select what layer each Vegetation Item Type will be rendered on. This is usefull to remove grass, plants etc from ocean reflection etc to increase rendering speed.

Vegetation Studio



Assign the directional sun light here and set the distance you want to load additional trees and large objects.

Vegetation Studio will calculate what trees have visible shadows and only render these for invisible trees.

VEGETATION SYSTEM - EDITOR TAB

This page is part of the documentation of the **VegetationSystem** Component.



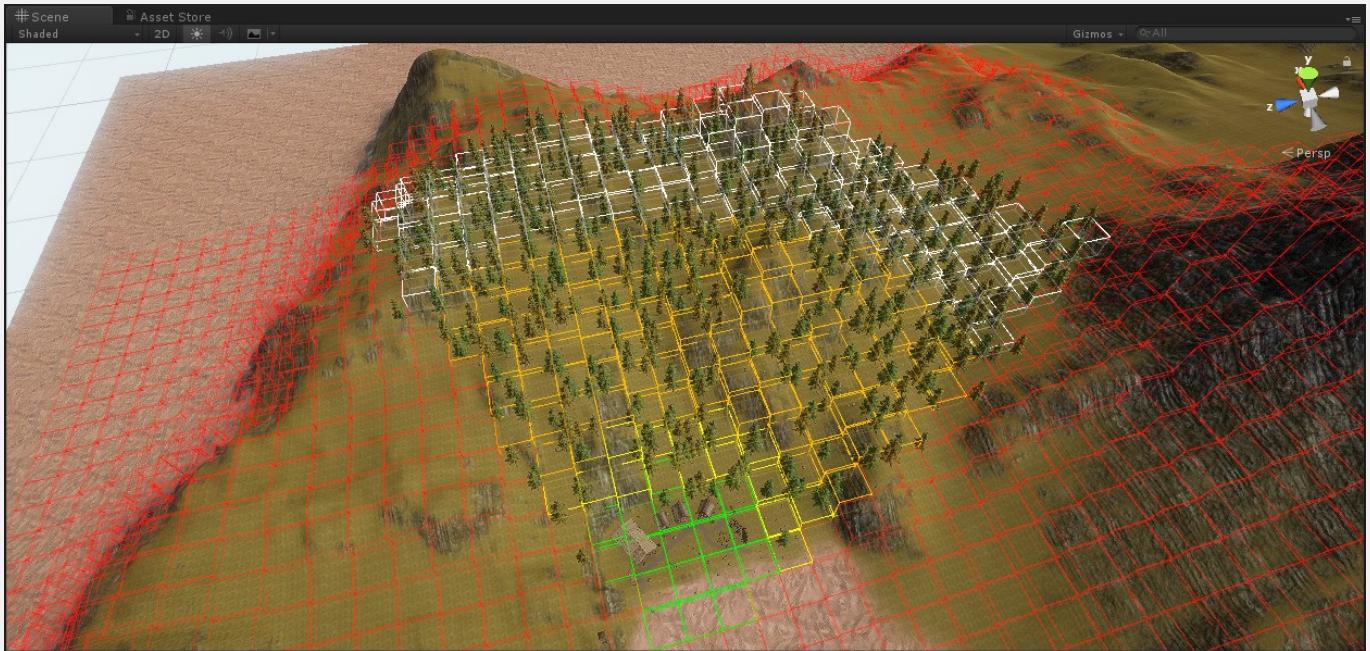
EDITOR SETTINGS

DISABLE VEGETATION IN EDIT MODE

With “Disable vegetation in edit mode” checked the VegetationSystem Component will not initialize while in editor mode. This can be useful when vegetation setup is complete and you want to save editor resources or with multiple terrains in the same scene, where you only use 1 terrain for configuring.

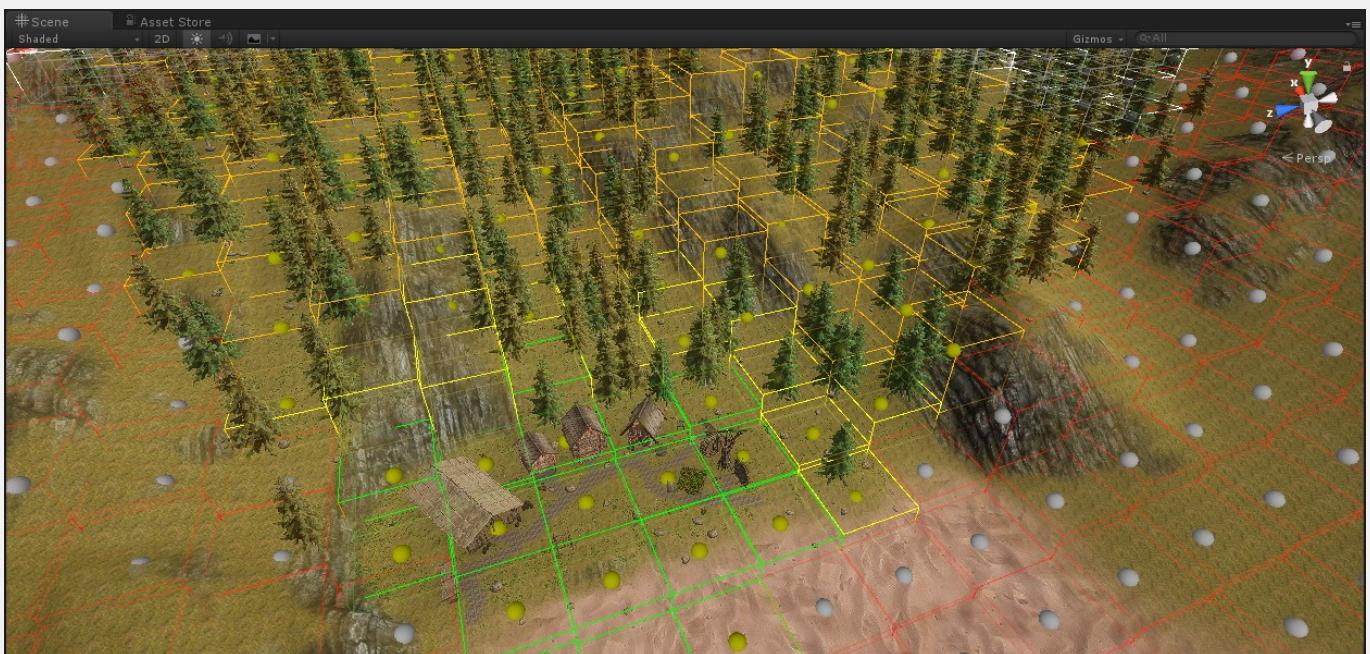
SHOW VEGETATION CELLS

Vegetation Studio



In order to handle fast culling and rendering of vegetation, VegetationSystem will split the terrain into cells. These are then culled using the CullingGroups API. Enable this option to see the current visible cells in sceneview. Cells are color coded based on visible LOD, Green, Yellow and Orange for vegetation distance and white for additional tree range.

SHOW CELLS LOADSTATE



In addition to the current LOD you can also enable loadstate for the cells. A yellow ball is shown for cells already in cache.

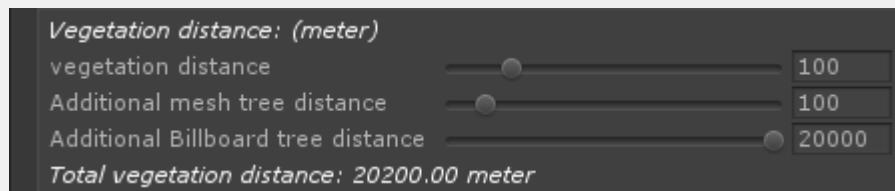
VEGETATION SYSTEM - VEGETATION TAB

This page is part of the documentation of the **VegetationSystem** Component.

Vegetation Studio



VEGETATION DISTANCE



VEGETATION DISTANCE

Vegetation Distance is the visible range for grass, plans and objects in the scene. Adjusting this does not require a respawn of cells in cache.

ADDITIONAL TREE DISTANCE

“Additional Tree Distance” is added to the “Vegetation Distance”. It will render Large Objects and Trees as meshes in this additional Range.

ADDITIONAL BILLBOARD TREE DISTANCE

“Additional Billboard Range” is distance on top of the additional tree distance. In this range billboards of trees will be rendered. See **BillboardSystem** component for more detailed info.



Tree billboards showing in the distance

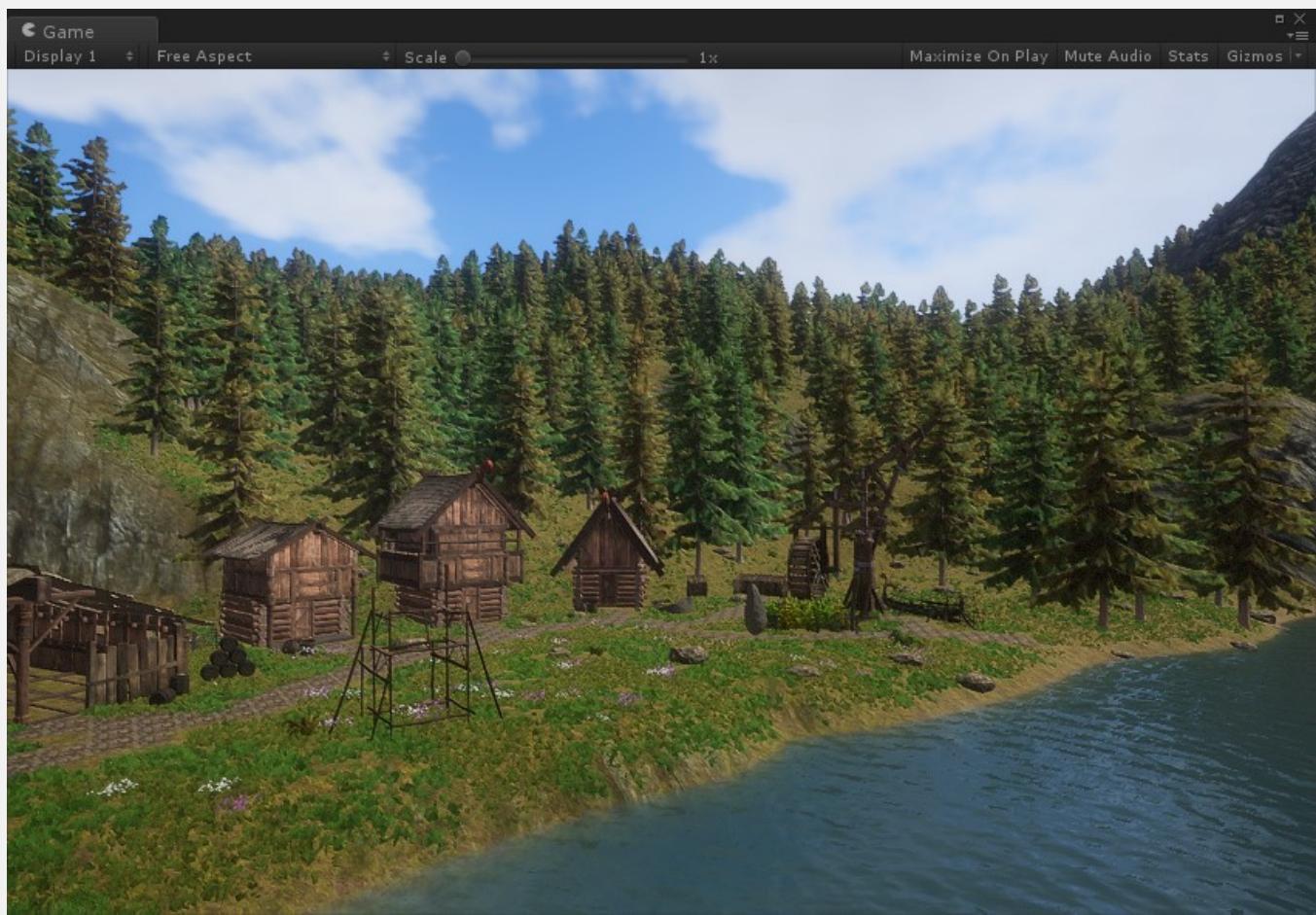
VEGETATION DENSITY

Vegetation density is designed to allow the developer to easily control the general density of vegetation. This will work as a multiplier for the sample distance of the individual Vegetation Item. This can be useful as an exposed setting for players to reduce density on lower quality computers.

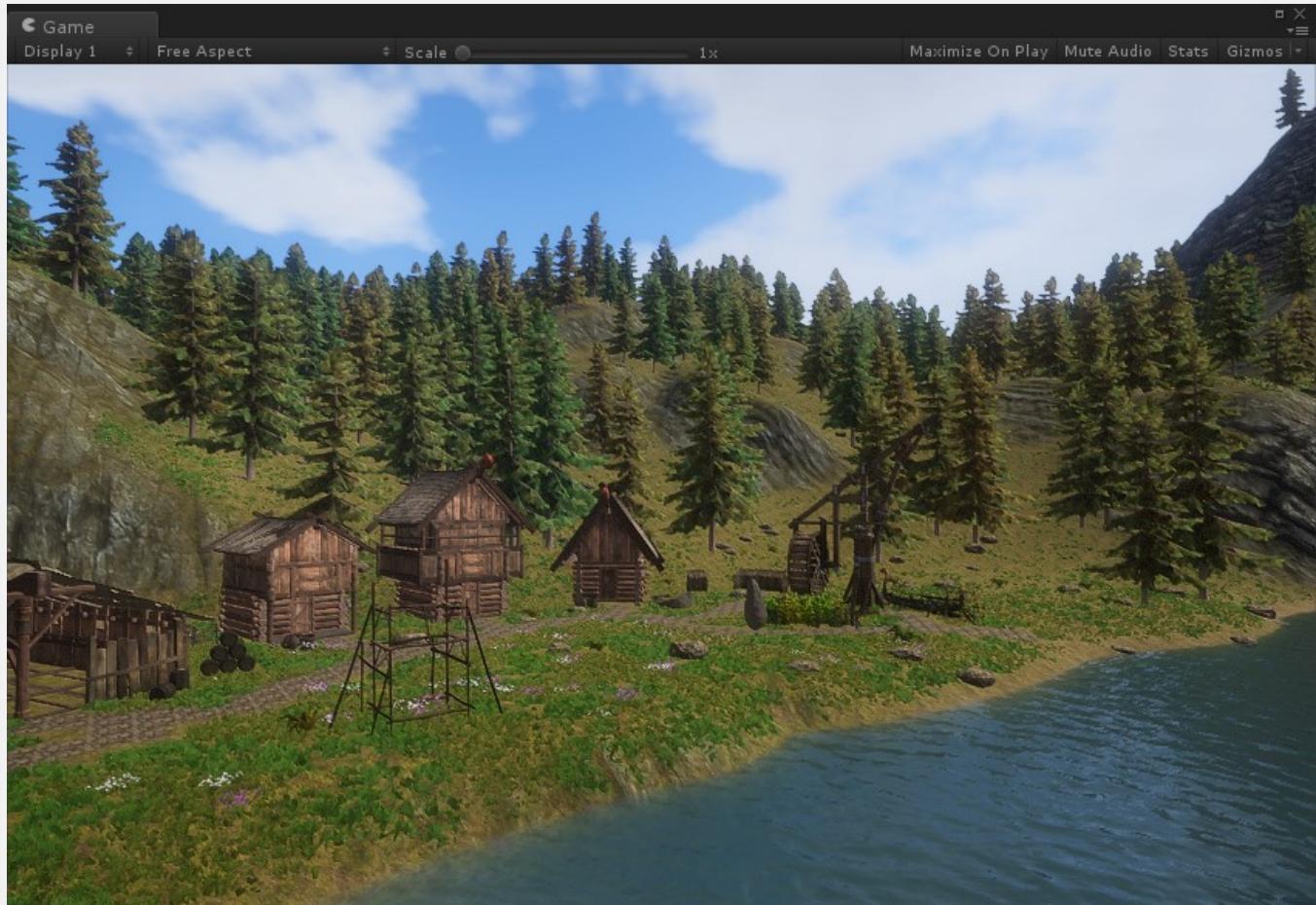
Vegetation Studio



Adjust setting to change scene tree density.



Same scene with tree density set to 1.7, 1 and 0.3



Same scene with tree density set to 1.7, 1 and 0.3

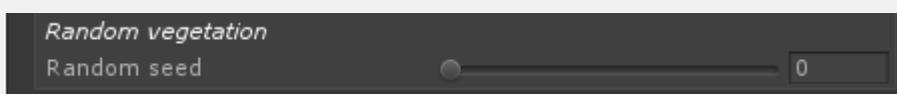


Same scene with tree density set to 1.7, 1 and 0.3

VEGETATION SCALE

Adjust this to change the scale of all VegetationItems in the vegetation package.

RANDOM VEGETATION



RANDOM SEED

All vegetation placement done by VegetationStudio is based on a random seed. Based on this the vegetation spawned will be 100% identical every time. Change the seed to get variations of vegetation.

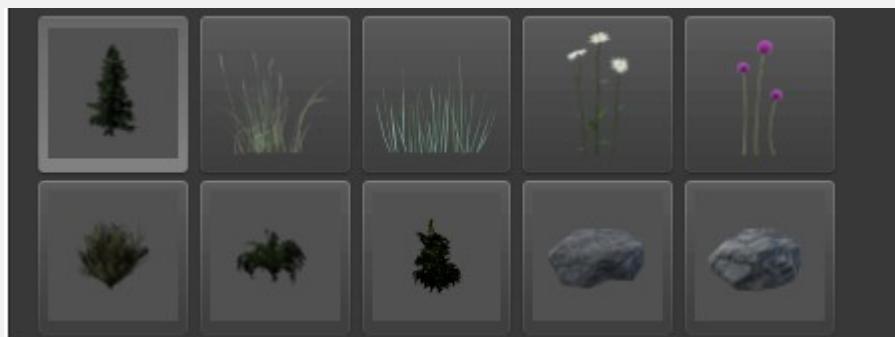
ADD VEGETATION ITEM



In order to add new vegetation to the Vegetation Package, drag and drop the prefab of the plant, tree etc to the corresponding drop area. The difference between the areas is the default configuration for each item. Sample distance, Type, rotation etc.

In addition to prefabs you can drop Texture2D grass and flowers direct. They will be used as mesh grass/plants and you have a range of settings. If you want more detailed control you can make a custom grass mesh patch prefab using the **Grass Patch Generator**.

SELECT ACTIVE VEGETATION ITEM



To remove or edit a Vegetation Item in the package, select it from the grid.

HEATMAP

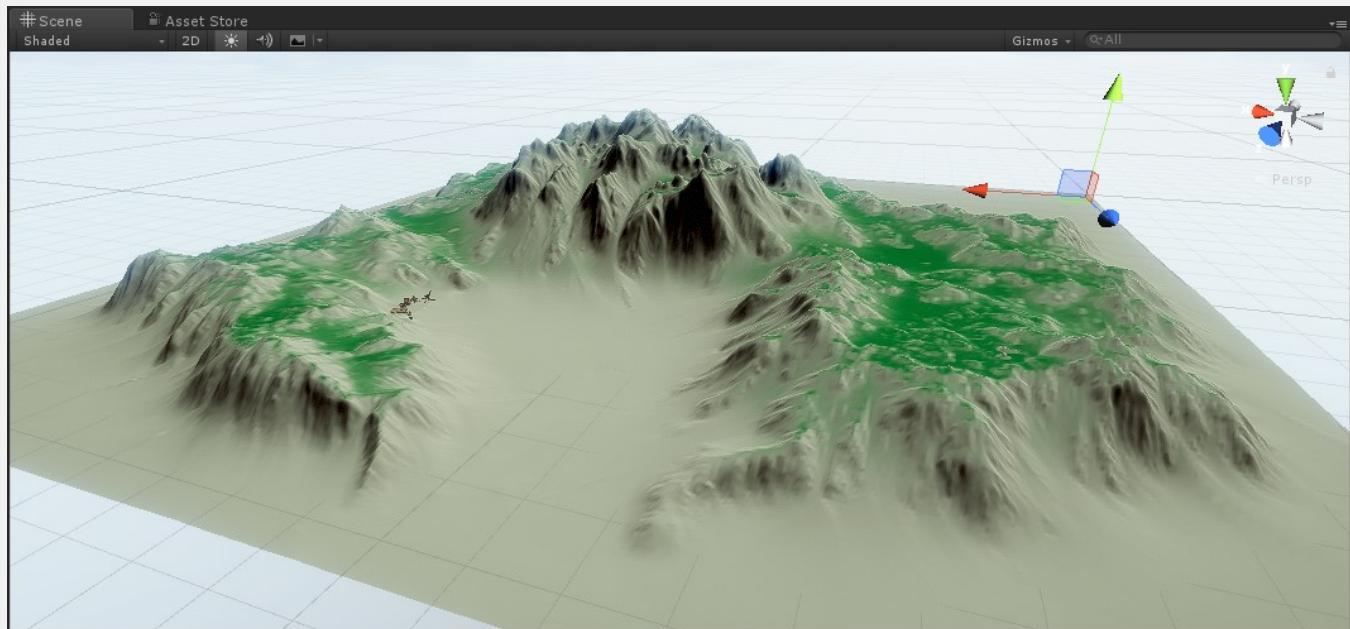


The heatmap function is designed to show you the potential spawn area for an Vegetation Item. Turn it on to see potential area based on height and steepness settings.

SHOW HEATMAP

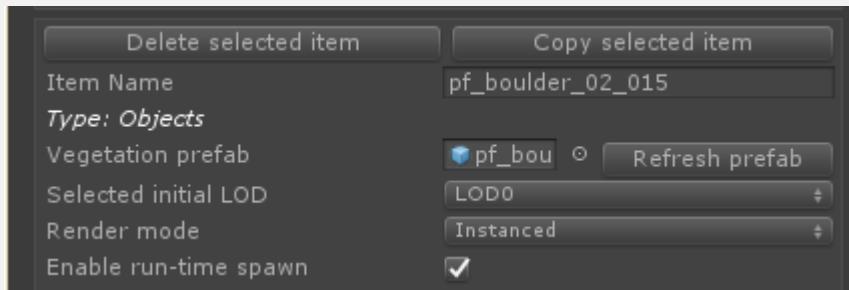
Enable to display heatmap. Heatmap will be disabled if you change selected GameObject or Tab.

Showing heatmap of vegetation using simple mode for height and steepness



For simple mode a min/max value is used but for Advanced mode a curve based system for steepness and height give much more control.

VEGETATION ITEM



DELETE SELECTED ITEM

This will delete the selected item from the Vegetation Package. Prefab or Texture2D is not touched.

DUPLICATE SELECTED ITEM

This will duplicate the selected item, with rules and prefabs.

ITEM NAME

Item name is your name for the Vegetation Item. This can be anything and is only used for your reference

VEGETATION PREFAB/TEXTURE

This is the link to the actual Prefab or Texture of the Vegetation Item. You can replace this with a new by dropping a new object.

VEGETATION TYPE

Settings like Billboards, Colliders, Grass settings, LOD groups is based on this setting. This is automatic when adding an Vegetation Item. Will be removed after Beta.

- Grass
- Plant
- Tree
- Object
- Large Object

SELECTED INITIAL LOD

Vegetation studio supports Vegetation Item prefabs with LOD groups (standard gameobject LOD naming). If you want to reduce the initial LOD used you can change this setting.

RENDER MODE

There are currently 3 different render mode for Vegetation Items.

- Normal

Using Graphics.DrawMesh. This is the slowest rendering method and a fallback for computers without instancing support.

- Instanced

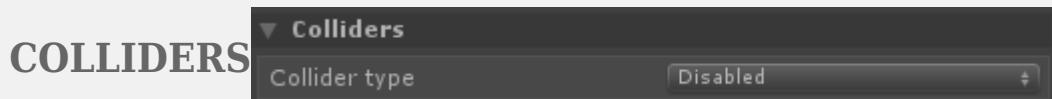
Most Vegetation Items will use Instanced rendering. They will be rendered in batches of up to 1023 per draw call at a low CPU cost.

- Instanced Indirect

At release only Vegetation Studio Grass is rendered InstancedIndirect. This is rendering done from a ComputeBuffer on the GPU. when set up the CPU use is very low and there is no 1023 batch limit like Instanced Rendering.

RUN-TIME SPAWN

Enable run-time spawn to activate the Vegetation Item run-time spawned. When you bake the item to the persistent storage it gets disabled since the item instances is loaded from the storage



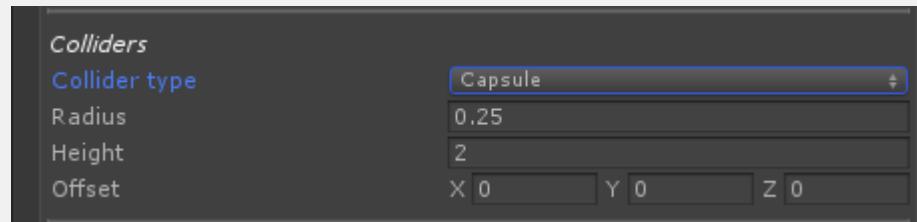
COLLIDER TYPE

Change collider type from disabled to add a collider to the Vegetation Item. This is possible on trees, objects and Large Objects.

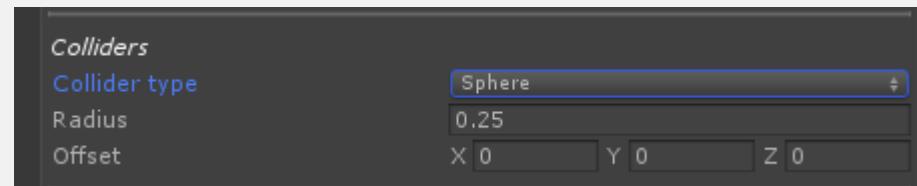
See **ColliderSystem** page for more detailed information.

- Capsule
- Sphere
- Mesh
- From Object

This option will copy all colliders on the same GameObject as the mesh used at the current LOD in the prefab.



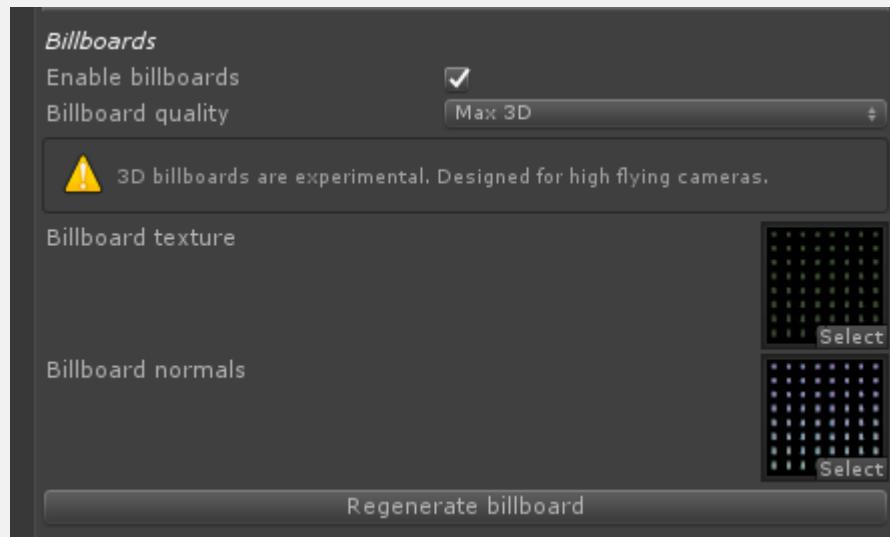
Adding a capsule collider to a Vegetation Item



Adding a sphere collider to a Vegetation Item

BILLBOARDS

See **BillboardSystem** component for more detailed info.



ENABLE BILLBOARDS

Enable toggle to use billboards on the selected tree.

BILLBOARD QUALITY

There are currently 6 different quality settings. 3 for 2D and 3 for 3D billboards.
Normal, High and Max. The settings control the size of the Atlas texture and sizes are 1024, 2048 and 4096.

BILLBOARD TEXTURE

The generated Atlas texture with the trees billboard.

BILLBOARD NORMALS

The generated Atlas normal texture with the trees billboard.

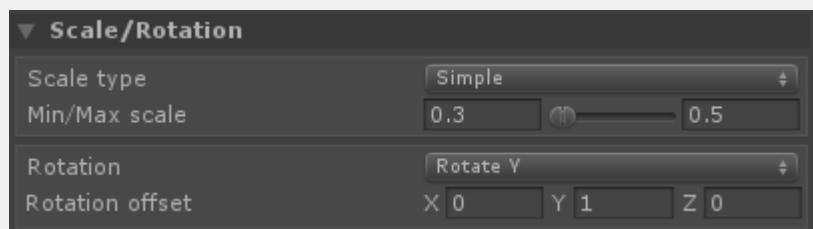
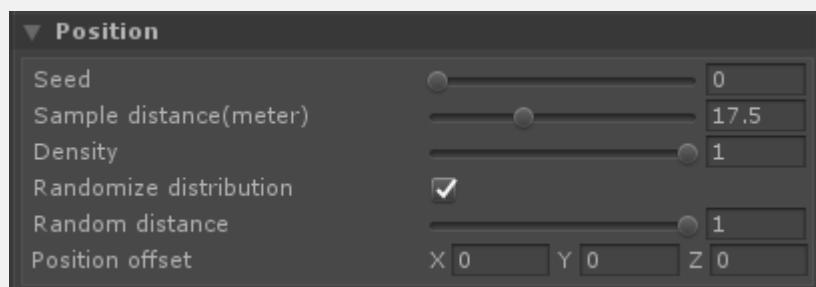
REGENERATE BILLBOARD

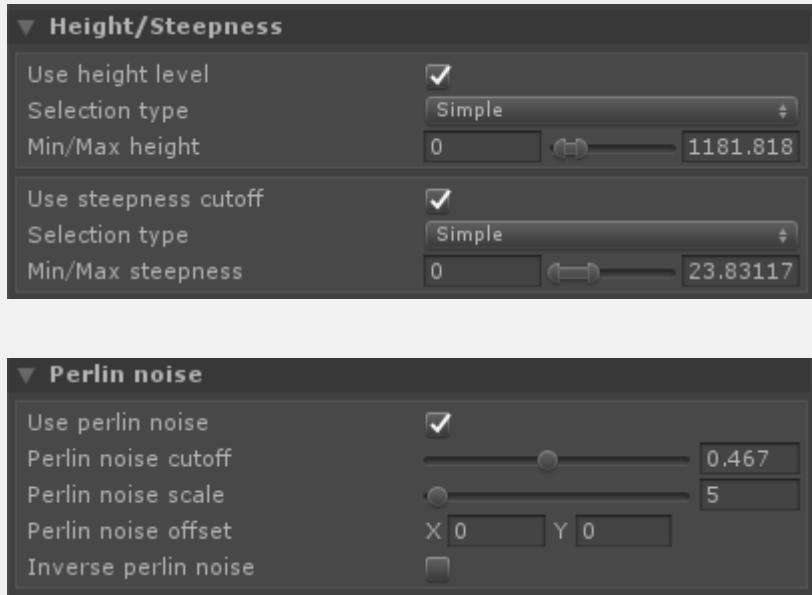
When changing lighting model in Unity you need to manually regenerate the billboards of all trees in the package.

Press regenerate to create new atlas.

SPAWN RULES

All rules enabled for Vegetation Item must be evaluated to true in order for the Vegetation Item to spawn.





POSITION

SAMPLE DISTANCE

When trying to find possible position for Vegetation Items the terrain will be sampled. Each Vegetation Item will sample at "Sample Distance" intervals in both x and z direction.

RANDOMIZE DISTRIBUTION

When enabled the sampled position will be randomized within 50% of sample distance.

MIN/MAX SCALE

The vegetation Item will be spawned with a random size based on the min/max value of the scale.

POSITION OFFSET

This Vector3 offset will be applied to the final sampled position. This is useful to move rocks down a bit more in the ground etc.

ROTATION

- Rotate around Y
- Rotate XYZ

- Follow terrain
- Follow terrain scale
- Geological Buckling. (comming soon)

When tilting of specific spawned objects in the XZ planes (compass) are needed, for instance with geological buckling, or tilting from the plane of deposition for sedimentary rocks; Vegetation Studio has XZ sliders that will tilt the models to be spawned in world space. This is independent of Y axis rotation, so that models spawned with this method can be of any Y axis rotation, but will always have the same world space XZ rotation. Ensuring a more realistic geological outcome for spawned rocks and strata. The same technique can be employed for spawning a Fraise (sharpened angled stake defensive barrier), or wind-swept trees on a high cliff; indeed any repeating object that requires tilting from the horizontal at a given compass direction

HEIGHT

Height setting is used to decide if a Vegetation Item can spawn in a location or not. It is relative to Sea level in Settings.

USE HEIGHT LEVEL

Enable to use height level as part of the rules.

SELECTION TYPE

- Simple
- Advanced

Advanced mode is still under development. Planned finished during the Beta.

MIN/MAX HEIGHT

the min and max height allowed for the Vegetation Item.

STEEPNESS

Seepness setting is used to decide if a Vegetation Item can spawn in a location or not. Value is location steepness (0-90 degrees)

USE STEEPNESS

Enable to use steepness as part of the rules.

SELECTION TYPE

- Simple
- Advanced
- Advanced mode is still under development. Planned finished during the Beta.

MIN/MAX STEEPNESS

the min and max steepness allowed for the Vegetation Item.

PERLIN NOISE

Perlin noise is used to make Vegetation Items grow in “organic” patterns.

USE PERLIN NOISE

Set toggle to enable to use perlin noise.

PERLIN NOISE CUTOFF

This sets the cutoff level for when to spawn a Vegetation Item. Perlin noise returns a value from 0-1.

PERLIN NOISE SCALE

Perlin noise scale sets the scale of the perlin noise area. A bigger scale will give bigger “patches” of vegetation.

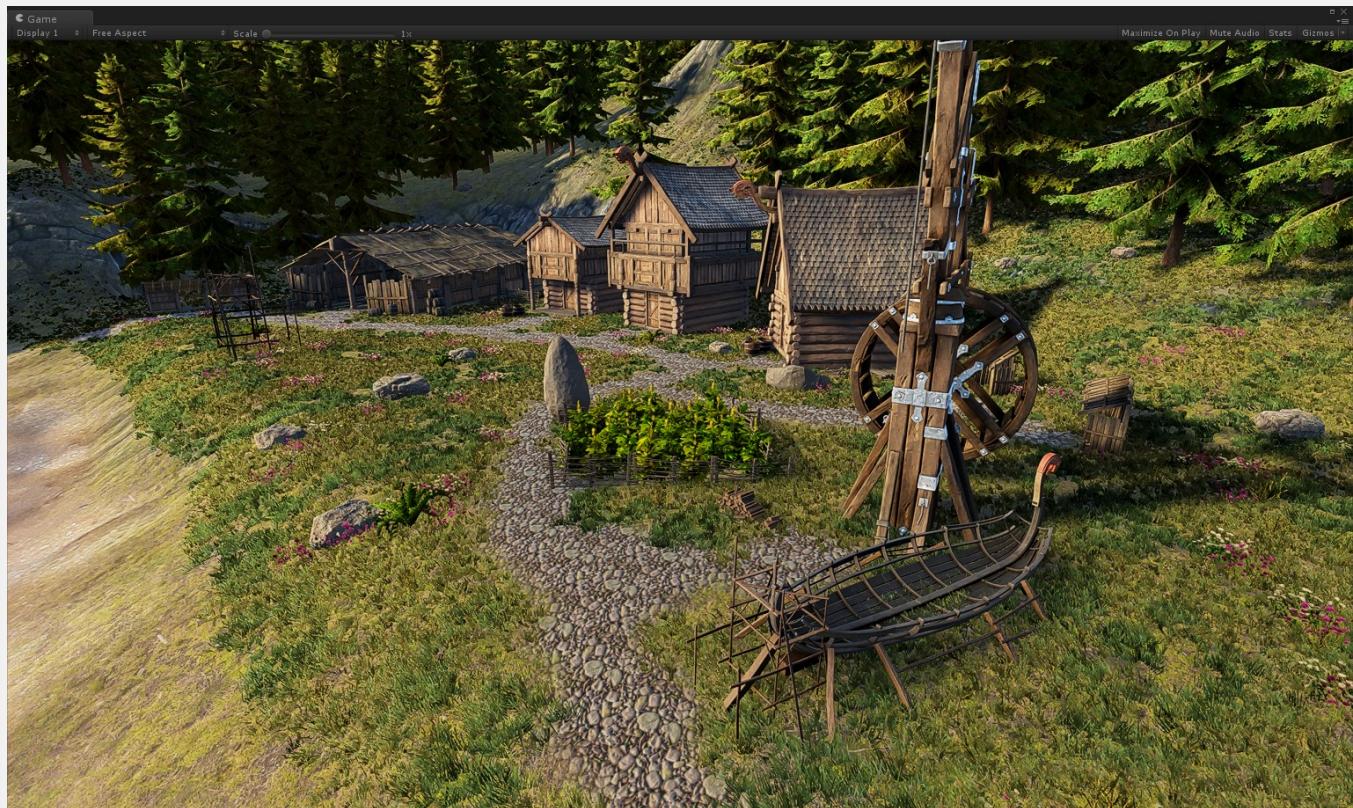
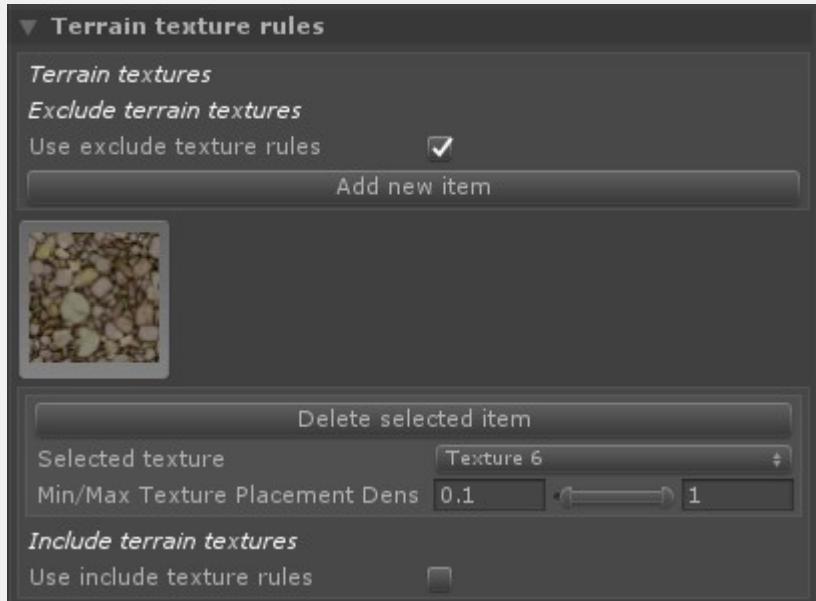
INVERSE PERLIN NOISE

Enable this toggle to inverse the perlin noise value. Example use case could be to have one normal and one inverted on 2 grass types and have them grow in oposite patterns.

TERRAIN TEXTURES

Terrain texture rules can be used both to include and exclude Vegetation Items from areas.

Vegetation Studio



The road here is masked out with a Terrain Texture exclude mask on the Vegetation Items. Any location with a density of 0.1 -1 of the path texture will have no vegetation.

EXCLUDE TERRAIN TEXTURES

Enable toggle to show UI and use exclude rules. You can add multiple Terrain textures and rules.

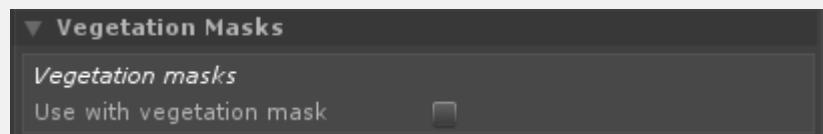
INCLUDE TERRAIN TEXTURES

Enable toggle to show UI and use include rules. You can add multiple Terrain textures and rules.

VEGETATION MASKS

Vegetation Items can be set to only spawn inside Vegetation Masks. In order to enable a rule for this enable “Use Vegetation Mask” and select a vegetation type.

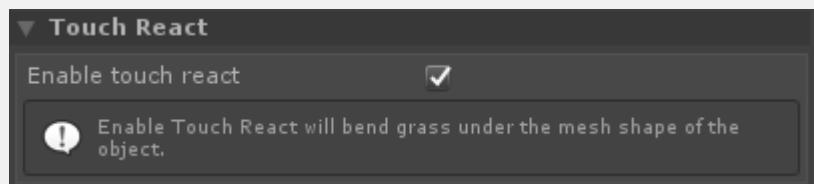
Any Vegetation Mask with Include Localized Vegetation set to the same ID will spawn this Vegetation Item. See **VegetationMaskArea** Component for detailed use.



USE VEGETATION MASKS

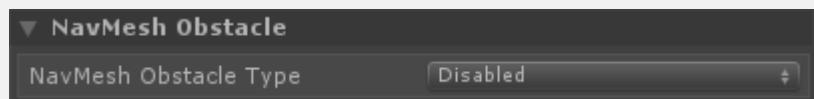
Enable toggle to show Vegetation ID dropdown selection and use vegetation mask rule.

Touch React



Enable touch reach to have the object bend grass. This is available on objects and large objects.

Navmesh Obstacle



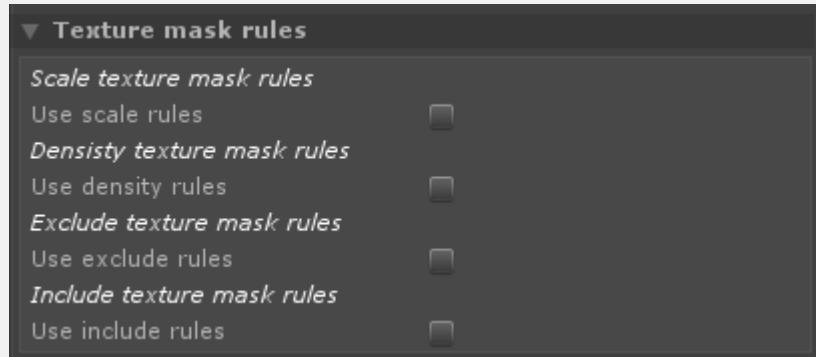
Select navmesh obstacle type. When enabled it will create a navmesh obstacle on the run-time spawned colliders to carve the navmesh.

Shadows



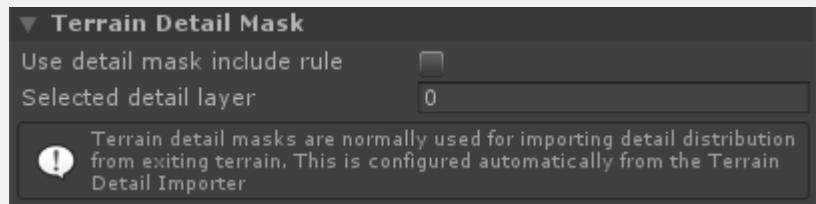
Check to disable shadows on this vegetation item. This will override any general setting.

Texture mask rules



Configure texture mask rules. Masks must be added on the Mask tab. Rules function the same way as terrain texture rules.

Terrain detail masks

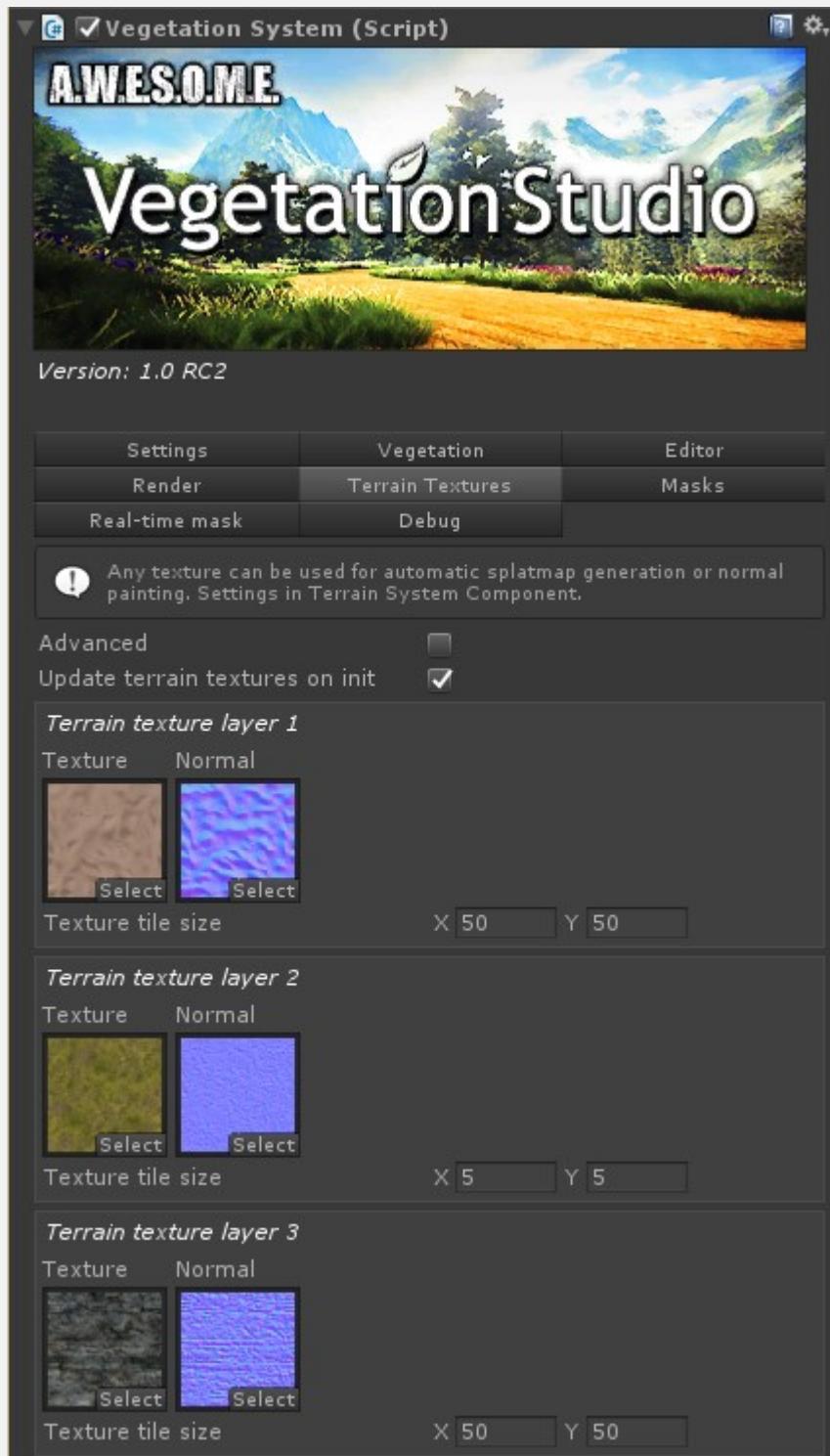


This include rule will use the unity terrain detail layers as a density mask for spawning vegetation items.

VEGETATION SYSTEM - TERRAIN TEXTURES TAB

This page is part of the documentation of the **VegetationSystem** Component.

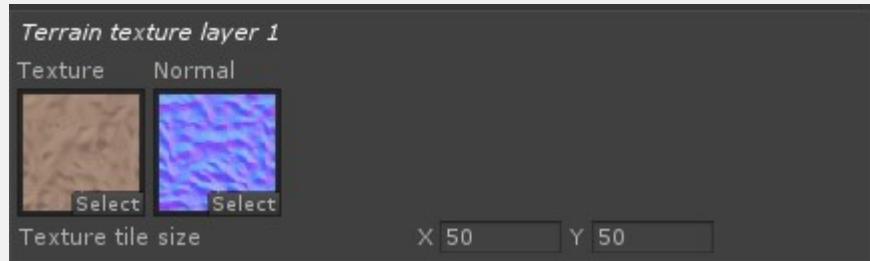
The Vegetation Package is designed to hold a reference to terrain textures. The amount of textures 0,4,8,12 or 16 is set at package creation. One of the benefits is that you can apply a package to a new terrain and have your textures and vegetation applied to it. Also when you switch between packages you will get the correct textures in the terrain.



UPDATE TERRAIN TEXTURES ON INIT

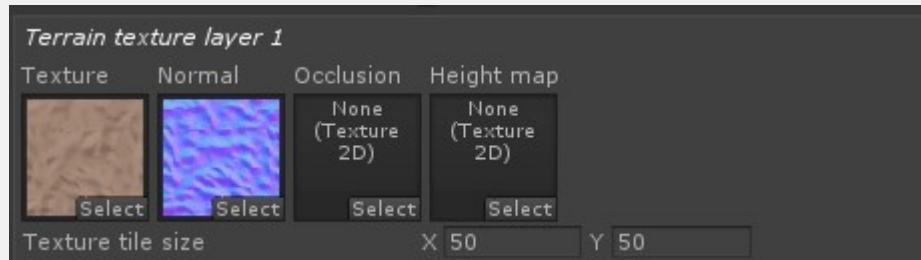
With "Update terrain textures on init" VegetationSystem will check the assigned terrain and update the textures to the textures in the Vegetation Package. This is used to change textures when you switch between packages. Summer, Winter, Desert etc. that do not have the same textures.

NORMAL MODE



In normal mode you have Albedo and Normal textures available. These will be assigned to the Unity Terrain. You can also set texture tile size.

ADVANCED MODE



Advanced mode is for use with plug-ins. It gives you the possibility to assign Ambient Occlusion and heightmap textures also.

These external plugins will get the array of changed textures and can be used to apply this to 3rd party terrain shaders.

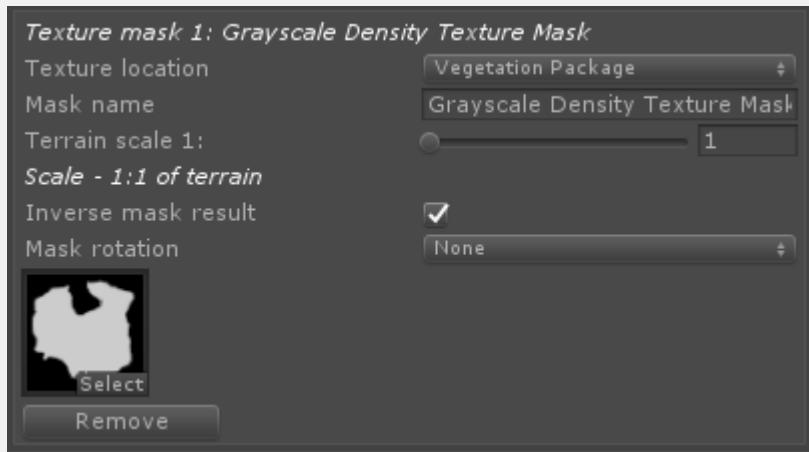
VEGETATION SYSTEM - MASKS TAB

This page is part of the documentation of the **VegetationSystem** Component.

Vegetation Studio supports texture masks as rules for VegetationItems. You can add multiple masks and then use them in the texture mask rules section of each VegetationItem. The masks are made using an interface where you can expose settings to the UI. You can create your own mask types that is exposed in the vegetation studio API.



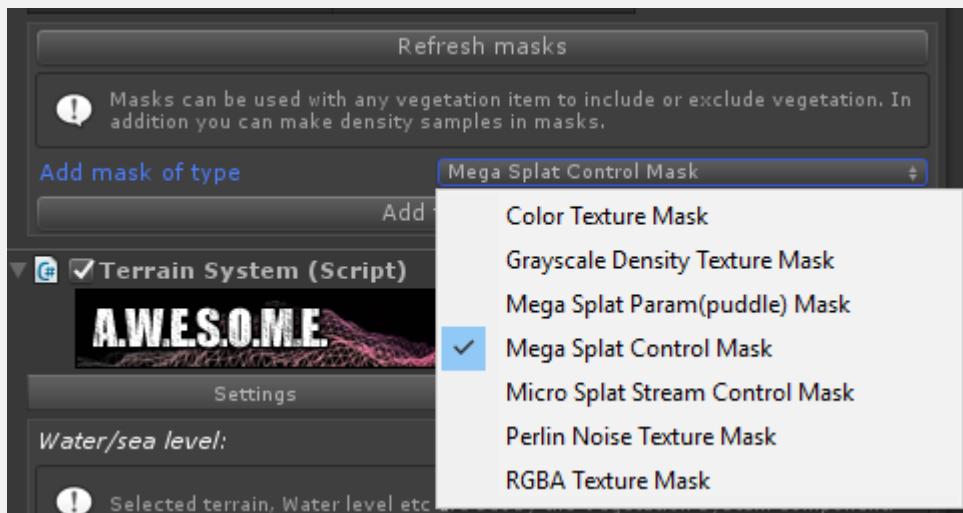
ADDING A MASK



Select the mask type and press “add texture mask”. When created the mask is now available to use as texture mask rules.

MASK TYPES

There are many mask types available for Vegetation Studio. More will probably be added.



COLOR TEXTURE MASK

This mask allows you to select a color and a tolerance for include and exclude rules.

GRAYSCALE DENSITY MASK

The average RGB value is returned from this mask.

MEGASPLAT PARAM MASK

2 layers are available from this mask. puddles and streams.

MEGASPLAT CONTROL MASK

The mega splat control mask lets you select one of the 256 mega splats layers. The coverage value for each location is returned.

MICROSPLAT STREAM CONTROL MASK

The microsplat mask lets you use stream, puddle, lava and wetness as spawn criteria.

PERLIN NOISE MASK

Sample mask for development. returns a perlin noise lookup. Same effect as the perlin noise rule in the VegetationItem

RGBA TEXTURE MASK

The RGBA mask lets you select a channel and use that for include, exclude and density rules.

VEGETATION SYSTEM - REAL-TIME MASKS

The real-time mask is a masking system that works with the compute shader render pipeline. You can add a mask texture with a 1:1 scale of the terrain and use this to mask out instanced indirect grass and plants. This happens every frame in the last compute shader pass and is updated instanced at close to 0 render cost. You can assign a texture2D or a rendertexture to the mask slot.

Select the channel and cutoff you want for the result.



This video shows a real-time mask produced by the lava flow system in microsplat. It is assigned and removes the grass real-time as lava flows.

VEGETATION SYSTEM - DEBUG TAB

This page is part of the documentation of the **VegetationSystem** Component.



SETTINGS

CACHE INFO

Cache info shows all trees and grass/plants currently spawned in cache. BillboardSystem will in many cases preload all trees on the terrain since billboard range is set large by default.

CLEAR CACHE

This will clear the current cache and reload for the visible cells the next frame

REFRESH HEIGHTMAP

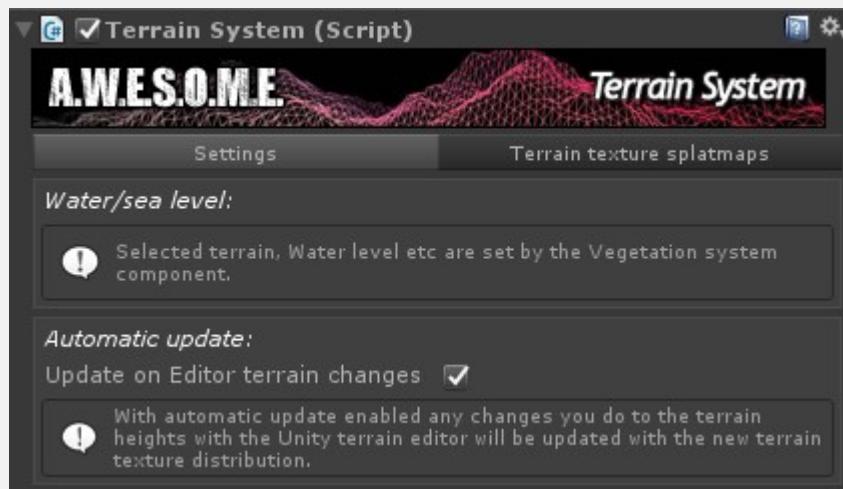
This will reload the heightmap from the terrain, clear cache and refresh the vegetation.

TERRAIN SYSTEM

The TerrainSystem Component does splat map generation for the terrain. You can setup a rule set based on any number of textures in your terrain and based on height, steepness and other settings it will generate your terrain splat map. In addition to this it allows for live editor updates and can update the splatmap and vegetation as you edit your terrain in the terrain inspector.

The textures not used for automatic splat map generation can be used to paint the terrain normally and will be kept even if you change the rule set and generate the splat map again. This way the manual work you do painting roads etc. will not be affected.

The component needs to be added to the same GameObject as VegetationSystem Component. This is done by default when setting up Vegetation Studio in a new scene. If Splat map generation is not needed, disable or remove the component.



Settings

Terrain Texture Splatmaps

SETTINGS

UPDATE ON EDITOR TERRAIN CHANGES

If this setting is enabled changes to the terrain using the Unity Terrain Inspector will be updated automatically as changes are done.



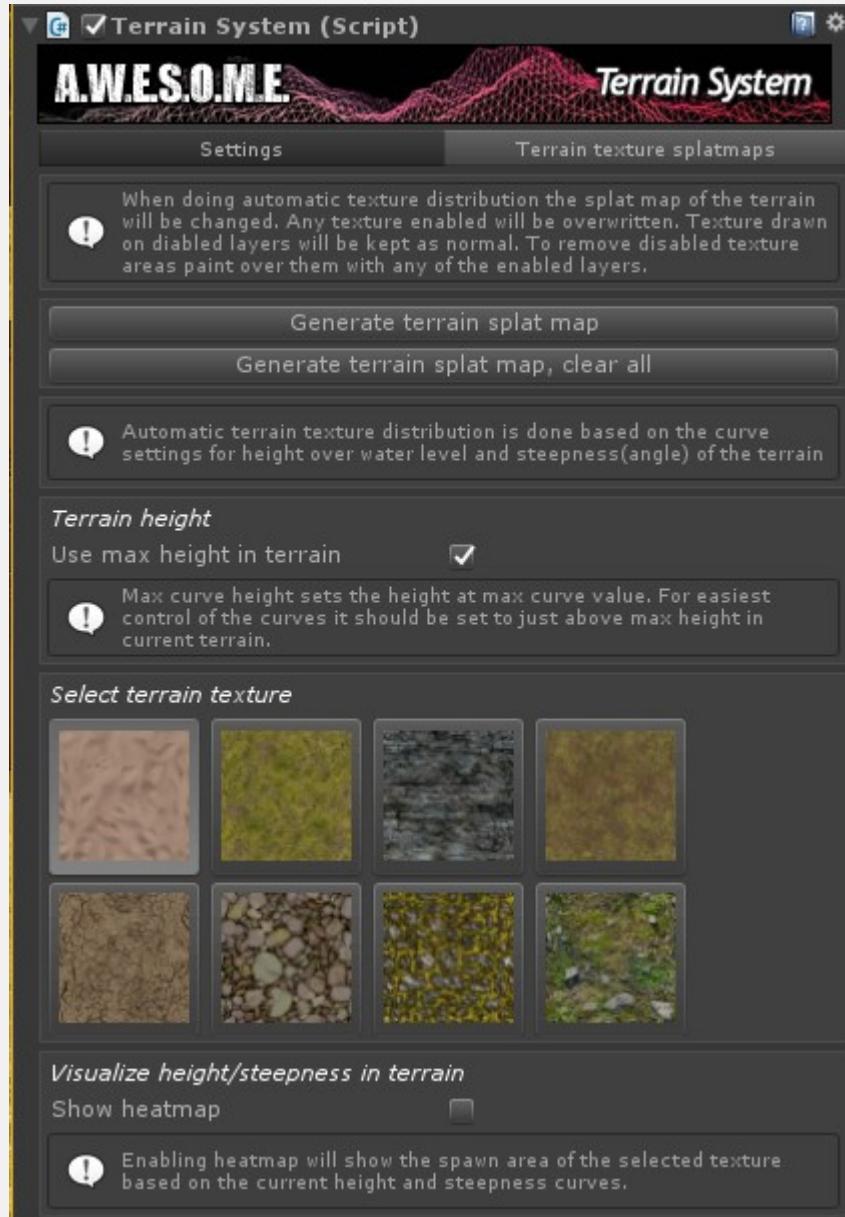
Painting terrain using the normal Terrain inspector. Vegetation has a rule set for no vegetation on road texture.

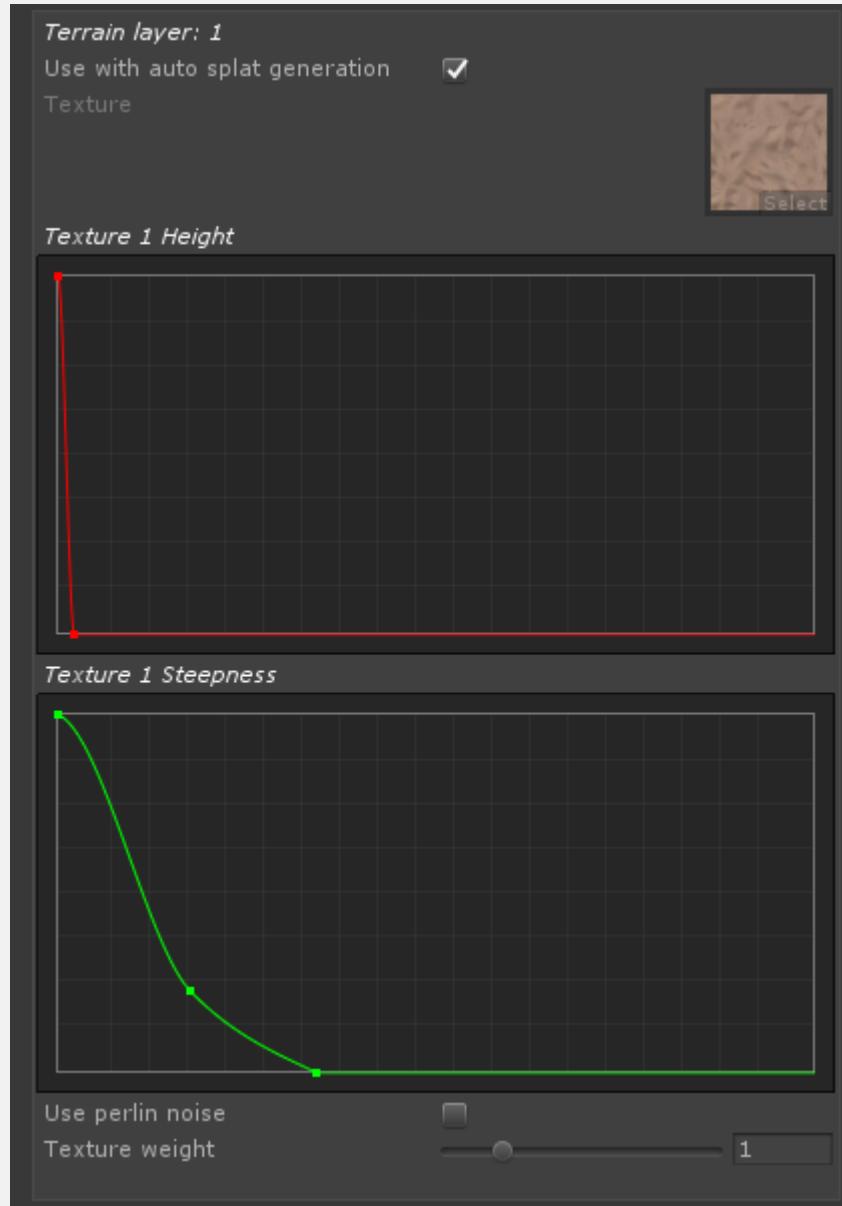


Automatic splat map generation on terrain height change. Vegetation updates automatically

TERRAIN TEXTURE SPLATMAPS

Vegetation Studio





GENERATE TERRAIN SPLAT MAP

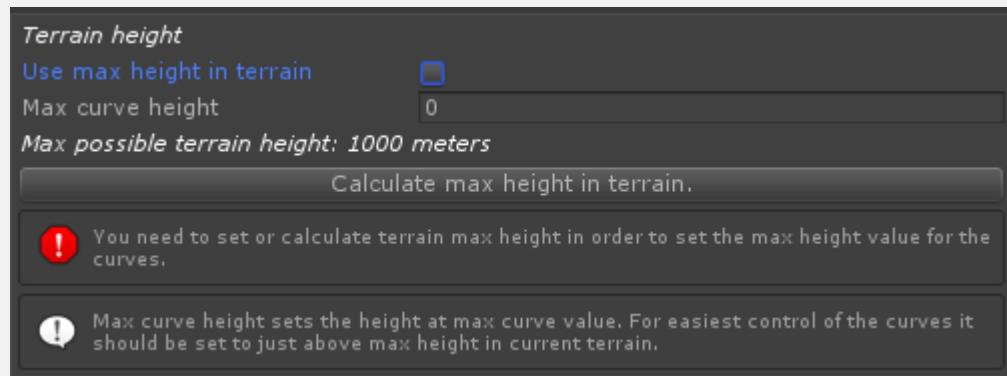
This option will regenerate the splat map from the current rules. Textures not used for automatic splat map generation painted on the terrain will be kept.

GENERATE TERRAIN SPLAT MAP, CLEAR ALL

This option will regenerate the splat map from current rules, clearing all painted textures in the terrain.

USE MAX HEIGHT IN TERRAIN

With the setting enabled the maximum height value in the terrain will be used when calculating the splat map. To set this to a custom value deselect and enter the value you want. This value is used as the max height value for the curves you configure for each terrain texture rule.



SELECT TERRAIN TEXTURE

Click on the thumbnail of the available terrain textures to select current texture.

SHOW HEATMAP

Enable show heatmap to display the result of the height and steepness rules of the selected terrain texture. If you select another GameObject it will be disabled.



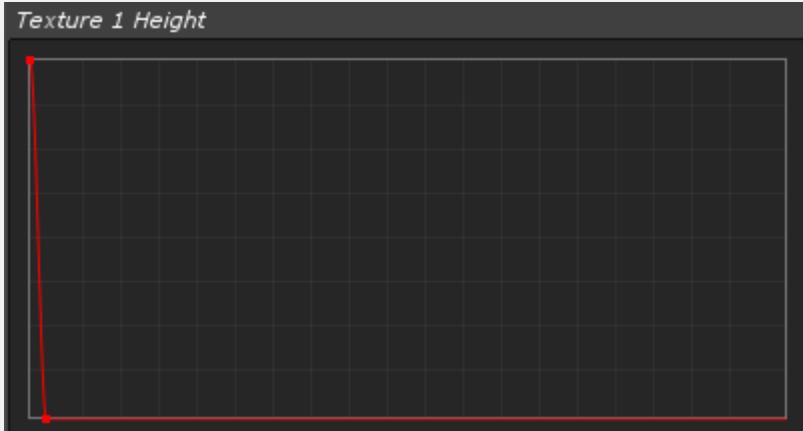
Using heatmap to show the coverage area of terrain textures.

USE WITH AUTO SPLAT GENERATION

Enable this setting to use texture with automatic splat generation. Textures not enabled will be kept in the terrain when splat map is generated.

HEIGHT CURVE

The height curve setting sets the use of the texture based on height. Vertical axis is amount and horizontal height where 1 is max.

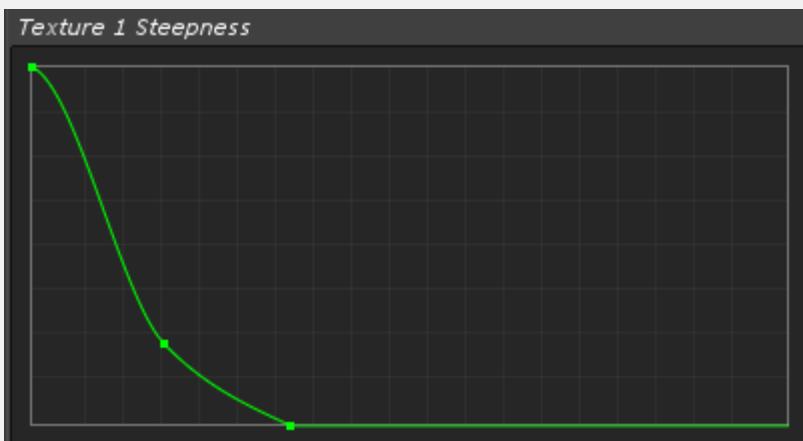


Height value goes from 0 to the set Max

terrain height.

STEEPNESS CURVE

The steepness curve setting sets the use of the texture based on terrain steepness. Vertical axis is amount and horizontal steepness where 1 is max.



steepness value goes from 0 to 90

degrees.



Adjusting steepness curve for a terrain texture splat map rule.

USE PERLIN NOISE

Enable "Use Perlin Noise" to introduce a randomness in the transitions between textures.

TEXTURE WEIGHT

Use Texture weight to adjust the general weight/importance of a Texture. If 2 textures have are both

spawned for a position they will be blended on the terrain. A higher weight on one of the textures will add more of that texture in the blend.

VEGETATION MASKS

Vegetation masks are a set of Components designed to control vegetation. They can be used both design and run-time. By adding areas with polygons or lines with a width you can remove, add or modify vegetation within the area. Common use cases are roads, houses, city areas etc.

See the components linked below for a more detailed description.



Example with a house with a vegetation mask. Vegetation will adapt to the house/mask as it is moved in the scene.

MASK TYPES

See the available mask types for a detailed description on use.

Vegetation Mask Area

Vegetation Mask Line

Vegetation Beacon

VEGETATION MASK AREA

The VegetationMaskArea component will handle run-time masking of vegetation. It is designed to be added to GameObjects in the scene and will make the vegetation adapt based on the settings. A mask will follow, scale and rotate with the GameObject. It can also be saved with prefabs and instanced at run-time.

In addition to the area defined by the nodes in the component, each vegetation type, Grass, Plants, Trees, Objects and Large Objects can be filtered and can have an additional range outside of the polygon mask area.

Masks are used at spawn time when new areas are loaded and will not affect rendering speed. Masking out vegetation may result in better performance.

Node editing

Handles

Mask settings

Global vegetation removal

Localized vegetation placement

Script access

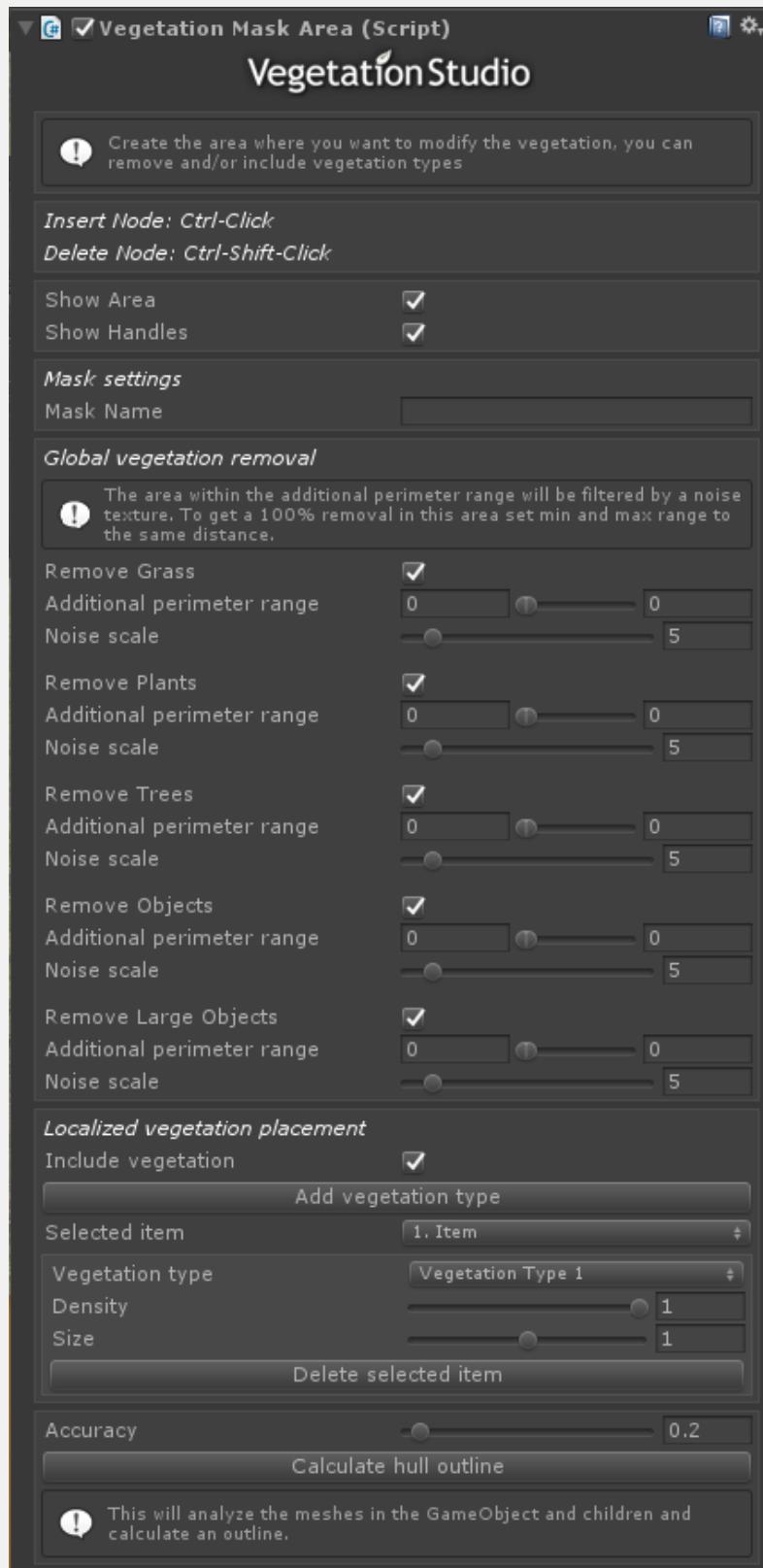


In this example an extra mask area that only removes trees is added to clear the area in front of the house. In addition to this each house has its own mask.



This image shows a VegetationMaskArea component added to a house mode. It has nodes on the house corner and in addition to this an added range to remove trees and plants.

Vegetation Studio





This house prefab has a vegetation mask. When you add it to a scene or move it, the vegetation will adapt.

NODE EDITING

You can add or delete nodes directly in the editor. Nodes will follow terrain. Ctrl-Click in terrain to add new nodes. They will position between the 2 closest nodes. Ctrl-Shift-Click to delete nodes.

*Insert Node: Ctrl-Click
Delete Node: Ctrl-Shift-Click*

HANDLES

Show area will draw a line around the polygon area in scene view in the editor.

Show handles will add movement handles to the scene view. Use them to move nodes. With high node count polygons (100+) handles at a distance will not show.

Show Area
Show Handles

MASK SETTINGS

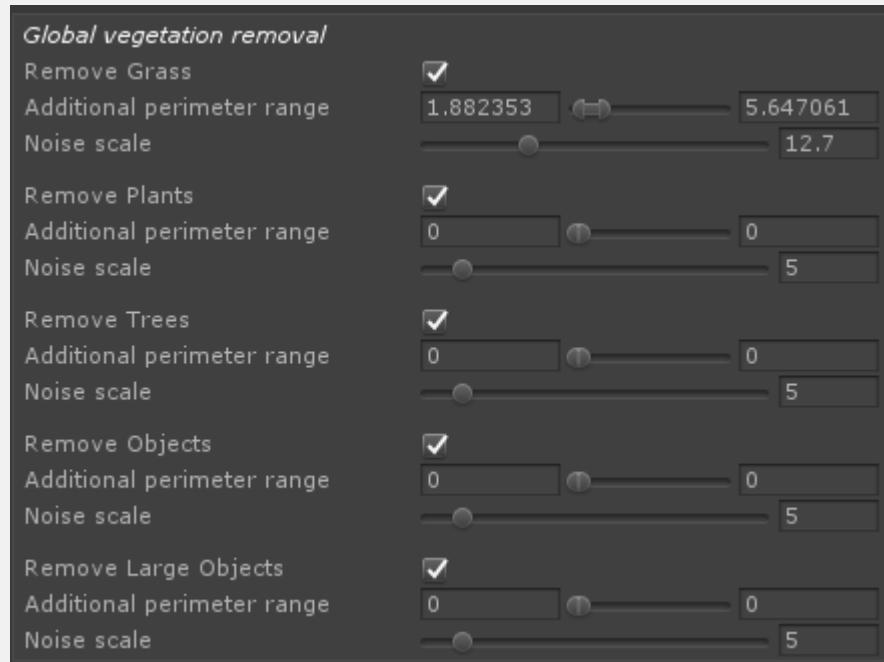
Mask name will show up as a label in center polygon in sceneview when option is turned on in VegetationStudioManager component.

Mask settings
Mask Name

GLOBAL VEGETATION REMOVAL

In order to remove vegetation within the polygon area enable Remove Grass, Plants, Trees, Objects or Large objects.

Additional perimeter ranges is in meters and can be set separate for each category. This will expand the polygon in all directions.



The additional perimeter range has a min/max value. The distance between these is using a perlin noise for the falloff to give a more organic edge to the mask. You can adjust the Noise scale to get a result you like

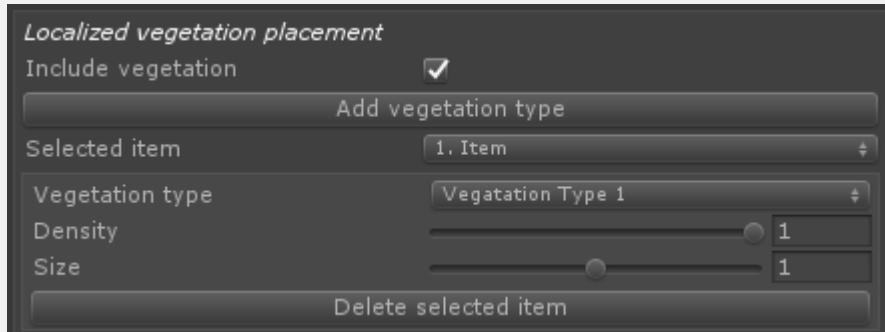
LOCALIZED VEGETATION PLACEMENT

Localized vegetation placement is used to include a vegetation type in a mask area. This could be used to introduce a new plant in a farm area, flowers in a garden etc. that does not spawn in the rest of the environment. In order to use enable the Include vegetation checkbox and add one or more Vegetation Types. The Vegetation item you want to spawn will have to be set with the same Vegetation Type ID in the Vegetation System component.

The vegetation item will be spawned inside the polygon with all normal rules set in the Vegetation System inspector.

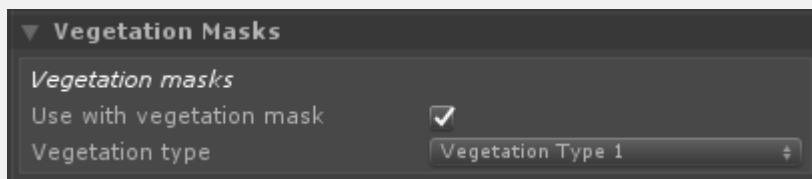
The density and size of these rules can be overridden on a mask to mask basis. This can give the effect of plants growing over time etc. These settings can be set run-time. This could allow you to switch plants in a field, make them bigger etc.

Vegetation Studio



VegetationMaskArea component set up to exclude all vegetation within the area and add a plant to the field. Here size parameter is changed.

In order to configure a plant to be used for vegetation masks you select the plant in the vegetation system inspector and at the bottom enable the Use Vegetation Mask checkbox and select an ID. Multiple plants can have the same ID.



SCRIPT ACCESS

The VegetationMaskArea component can be added to any GameObject run-time. The mask settings and points can be changed at any time. The `UpdateVegetationMask()` function must be called after changes from script.

Available mask settings and default values are:

```
public bool RemoveGrass = true;
public bool RemovePlants = true;
public bool RemoveTrees = true;
public bool RemoveObjects = true;
public bool RemoveLargeObjects = true;
public float AdditionalGrassPerimiter = 0;
public float AdditionalPlantPerimiter = 0;
public float AdditionalTreePerimiter = 0;
public float AdditionalObjectPerimiter = 0;
public float AdditionalLargeObjectPerimiter = 0;
```

Adding a new mask to a gameobject:

```
VegetationMaskArea vegetationMaskArea =
this.gameObject.AddComponent<VegetationMaskArea >();
vegetationMaskArea.RemoveGrass = true;
vegetationMaskArea.AdditionalGrassPerimiter = 5f;
vegetationMaskArea.ClearNodes();
vegetationMaskArea.AddNodesToEnd(pointListArray);
//Points in the array list needs to be in worldspace positions.
vegetationMaskArea.UpdateVegetationMask();
```

The VegetationMaskArea will internally create a object of the type PolygonMaskArea. If you have your own system to manage polygon areas you want to mask you can also do this directly. You need to create a new PolygonMaskArea object, configure it and add it to the static VegetationStudioManager.AddVegetationMask(maskArea); function. You will be responsible for keeping a reference to the PolygonMaskArea object and remove it and add a new if you want to change it. VegetationStudioManager.RemoveVegetationMask(maskArea);

```
List<Vector3> worldSpaceNodeList = GetWorldSpaceNodePositions();
//Replace GetWorldSpaceNodePositions with your own code to make a list of Vector3
positions in worldspace.
PolygonMaskArea maskArea = new PolygonMaskArea
{
    removeGrass = RemoveGrass,
    removePlants = RemovePlants,
    removeTrees = RemoveTrees,
    removeObjects = RemoveObjects,
    removeLargeObjects = RemoveLargeObjects,
    additionalGrassWidth = AdditionalGrassPerimeter,
    additionalPlantWidth = AdditionalPlantPerimeter,
    additionalTreeWidth = AdditionalTreePerimeter,
    additionalObjectWidth = AdditionalObjectPerimeter,
    additionalLargeObjectWidth = AdditionalLargeObjectPerimeter
};
maskArea.AddPolygon(worldSpaceNodeList);
VegetationStudioManager.AddVegetationMask(maskArea);
```

Look in VegetationMaskArea.cs for example on how to add localized vegetation placement from code also.

VEGETATION MASK LINE

The VegetationMaskLine component will handle run-time masking of vegetation. It is designed to be added to GameObjects in the scene and will make the vegetation adapt to them based on the settings. A mask will follow, scale and rotate with the GameObject. It can also be saved with prefabs and instanced at run-time.

In addition to the area defined by the nodes in the component, each vegetation type, Grass, Plants, Trees, Objects and Large Objects can be filtered and can have an additional range outside of the polygon mask area.

Masks are used at spawn time when new areas are loaded and will not affect rendering speed. Masking out vegetation may result in better performance.

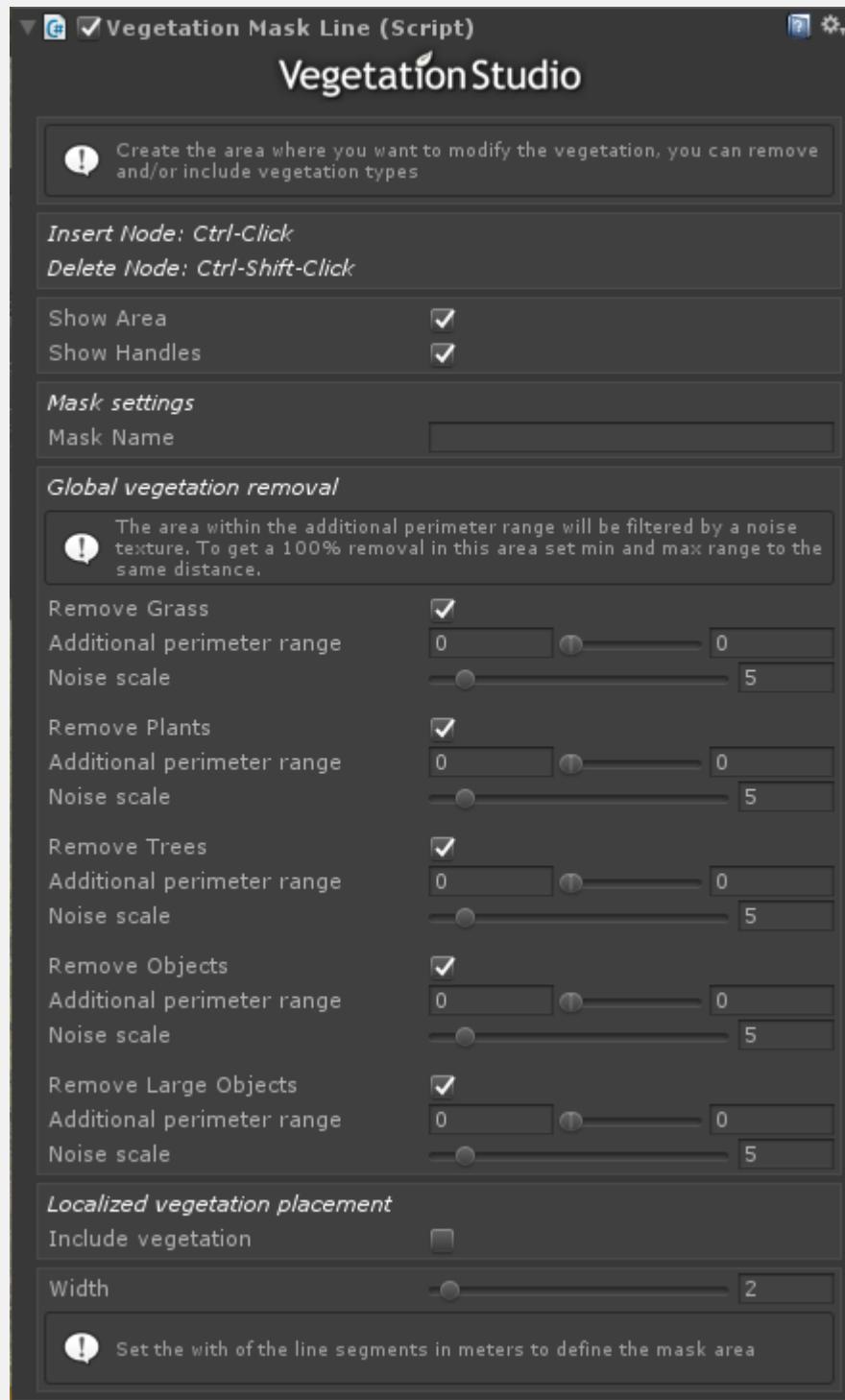




Image showing a road masked using a **VegetationMaskLineComponent**. (Roads by **Sentieri**)

Node editing**Handles****Mask settings****Global vegetation removal****Localized vegetation placement****Width****Script access**

NODE EDITING

You can add or delete nodes direct in the editor. Nodes will follow terrain. Ctrl-Click in terrain to add new nodes. They will be positioned between the 2 closest nodes. Ctrl-Shift-Click to delete nodes.

*Insert Node: Ctrl-Click
Delete Node: Ctrl-Shift-Click*

HANDLES

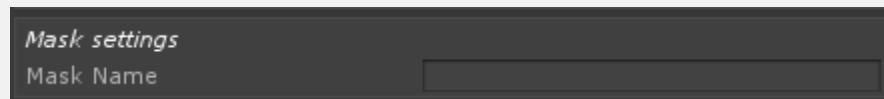
Show area will draw a line around the polygon area in scene view in the editor.

Show handles will add movement handles to the scene view to move nodes. With high node count polygons (100+) handles at a distance will not show.



MASK SETTINGS

Mask name will show up as a label in the center polygon in sceneview when option is turned on in VegetationStudioManager component.

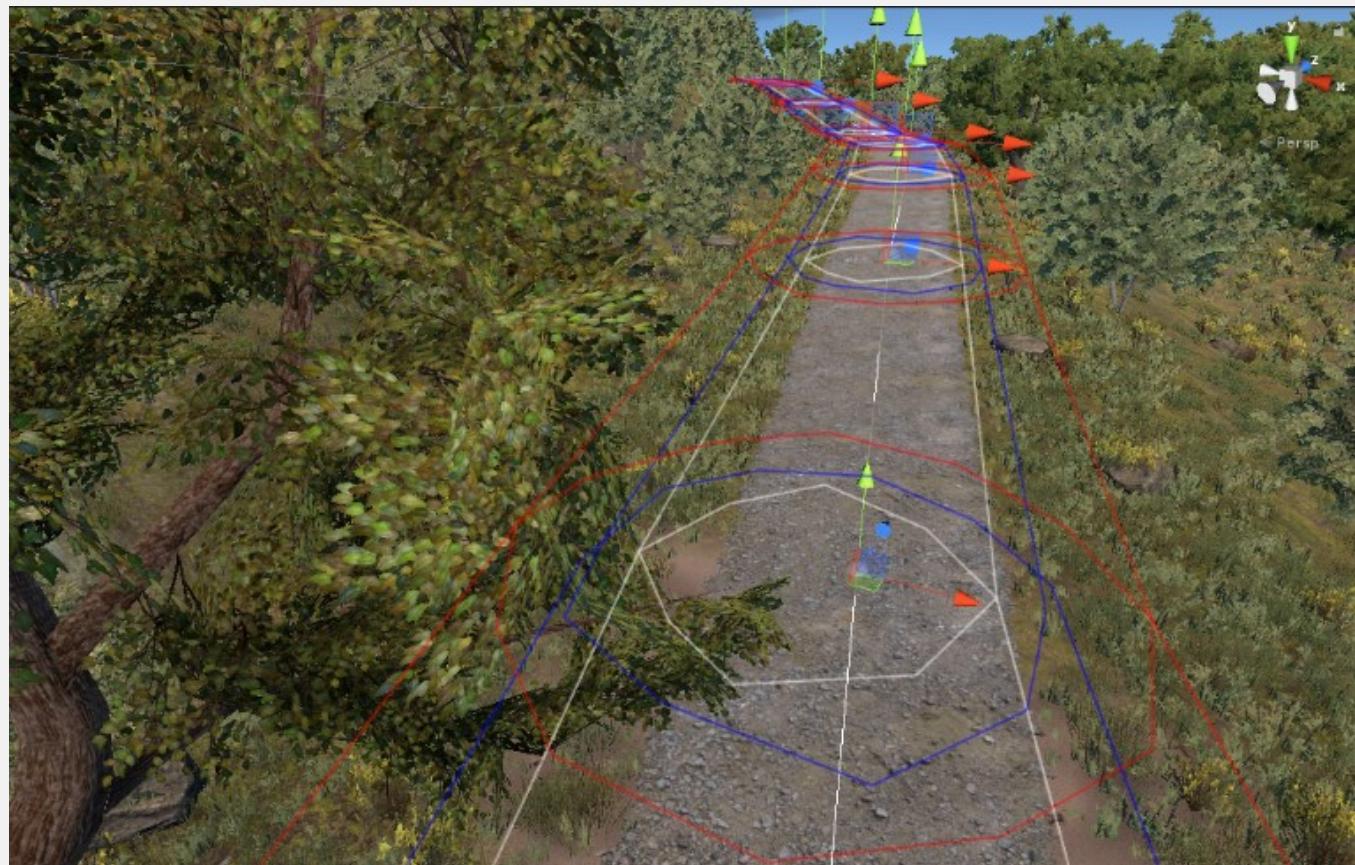
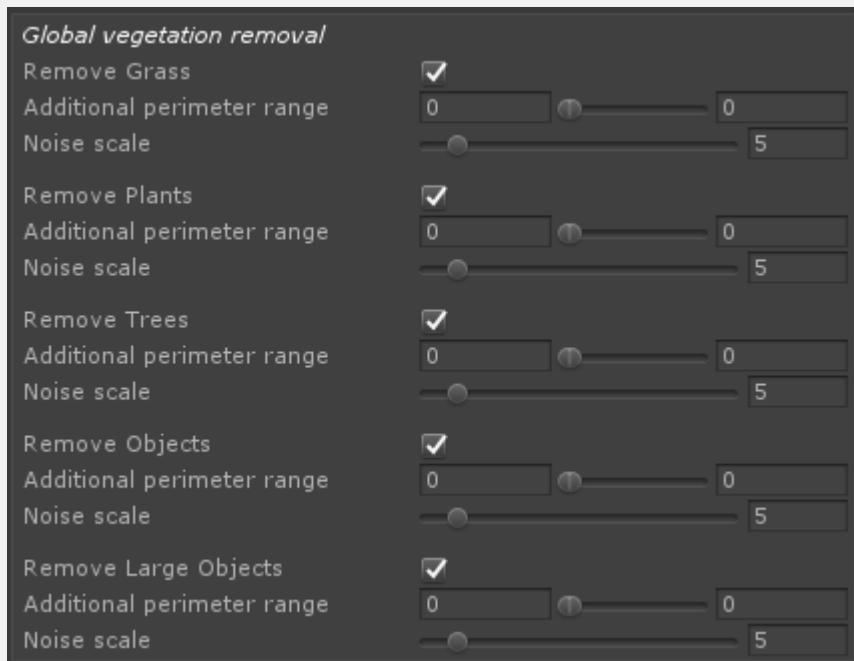


GLOBAL VEGETATION REMOVAL

In order to remove vegetation within the polygon area enable Remove Grass, Plants, Trees, Objects or Large objects.

Additional perimeter distances is in meters and can be set separately for each category. This will expand the polygon in all directions.

Vegetation Studio



It is possible to set a different additional distance on top of the line width. In this case trees are kept further away from the road.

Plants limited for a distance and grass only removed in road area.



Road masked out with a VegetationMaskLine component. (Roads by **Sentieri**)

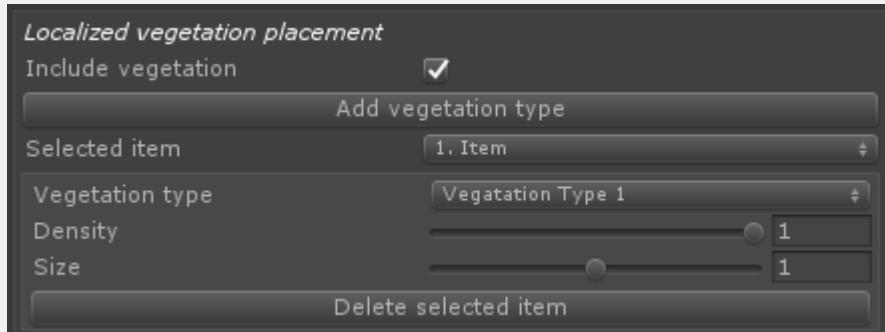
The additional perimeter range has a min/max value. The distance between these is using a perlin noise for the falloff to give a more organic edge to the mask. You can adjust the Noise scale to get a result you like

LOCALIZED VEGETATION PLACEMENT

Localized vegetation placement is used to include a vegetation type in a mask line. This could be used to introduce a new plant as a hedge, plant trees in a row by a road etc. that does not spawn in the rest of the environment. In order to use enable the Include vegetation checkbox and add one or more Vegetation Types. The Vegetation item you want to spawn will have to be set with the same Vegetation Type ID in the Vegetation System component.

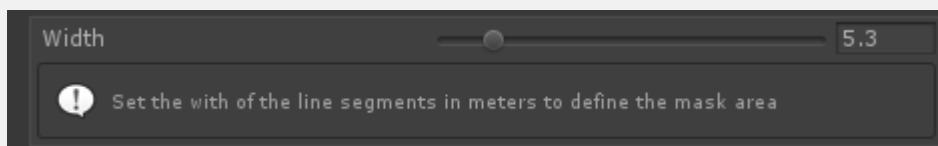
The vegetation item will be spawned inside the polygon with all normal rules set in the Vegetation System inspector.

The Density and size of these rules can be overridden on a mask to mask basis. This can give the effect of plants growing over time etc. These settings can be set run-time. This could allow you to switch plants in a field, make them bigger etc.



WIDTH

Width setting controls the width of the line mask.



SCRIPT ACCESS

The VegetationMaskLine component can be added to any GameObject run-time. The mask settings and points can be changed at any time. The UpdateVegetationMask() function must be called after changes from script.

Available mask settings and default values are:

```
public bool RemoveGrass = true;
public bool RemovePlants = true;
public bool RemoveTrees = true;
public bool RemoveObjects = true;
public bool RemoveLargeObjects = true;
public float AdditionalGrassPerimiter = 0;
public float AdditionalPlantPerimiter = 0;
public float AdditionalTreePerimiter = 0;
public float AdditionalObjectPerimiter = 0;
public float AdditionalLargeObjectPerimiter = 0;
```

Adding a new mask to a gameobject:

```
VegetationMaskLine vegetationMaskLine =
this.gameObject.AddComponent<VegetationMaskLine>();
vegetationMaskLine.RemoveGrass = true;
vegetationMaskLine.AdditionalGrassPerimiter = 5f;
vegetationMaskLine.ClearNodes();
vegetationMaskLine.AddNodesToEnd(pointListArray);
//Points in the array list needs to be in worldspace positions.
vegetationMaskLine.UpdateVegetationMask();
```

The VegetationMaskLine will internally create a object of the type PolygonMaskLine. If you have your own system to manage polygon areas you want to mask you can also do this direct. You need to create a new PolygonMaskLine object, configure it and add it to the static VegetationStudioManager.AddVegetationMask(maskLine); function. You will be responsible for keeping a reference to the PolygonMaskLine object and remove it and add a new if you want to change it. VegetationStudioManager.RemoveVegetationMask(maskLine); Internally the VegetationMaskLine component will make one PolygonMaskLine object per segment in the line.

```
List<Vector3> worldSpaceNodeList = GetWorldSpaceNodePositions();
//Replace GetWorldSpaceNodePositions with your own code to make a list if Vector3
positions in worldspace.
PolygonMaskLine maskLine = new PolygonMaskLine
{
removeGrass = RemoveGrass,
removePlants = RemovePlants,
removeTrees = RemoveTrees,
removeObjects = RemoveObjects,
removeLargeObjects = RemoveLargeObjects,
additionalGrassWidth = AdditionalGrassPerimiter,
additionalPlantWidth = AdditionalPlantPerimiter,
additionalTreeWidth = AdditionalTreePerimiter,
additionalObjectWidth = AdditionalObjectPerimiter,
additionalLargeObjectWidth = AdditionalLargeObjectPerimiter
};
maskLine.AddPolygon(worldSpaceNodeList);
VegetationStudioManager.AddVegetationMask(maskLine);
```

Look in VegetationMaskLine.cs for example on how to add localized vegetation placement from code also.

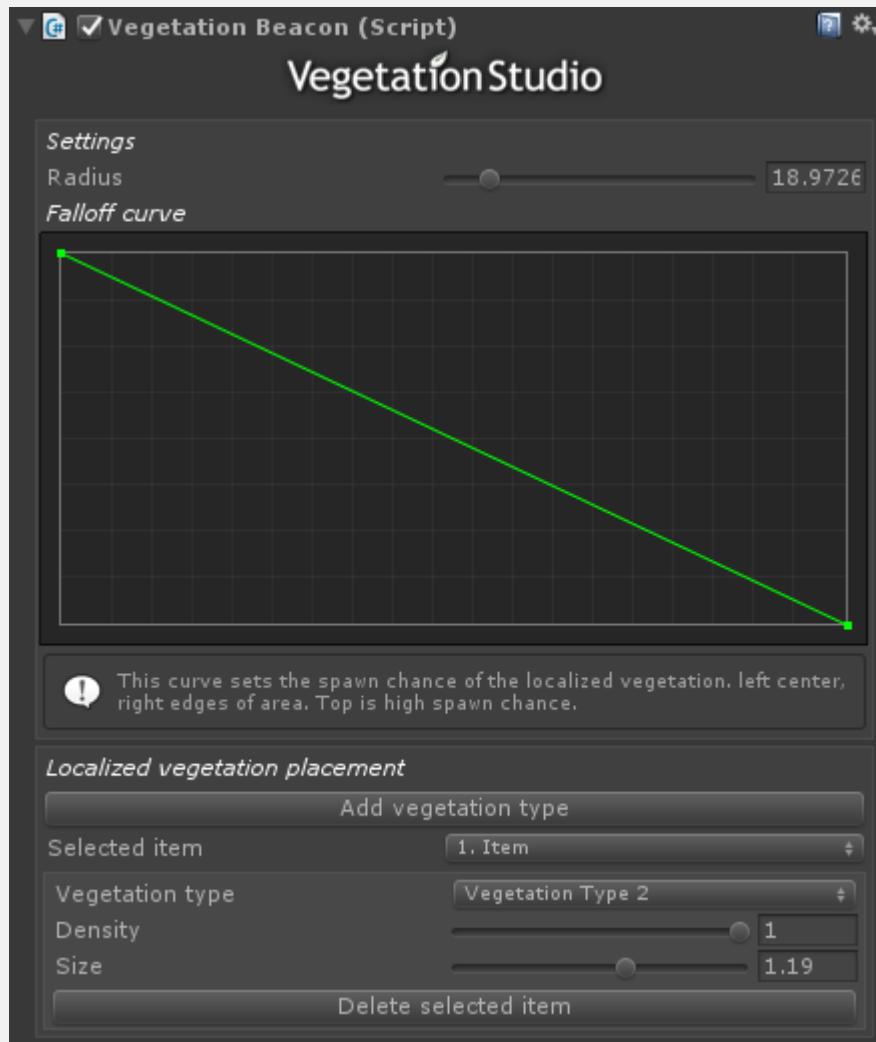
VEGETATION BEACON

The VegetationBeacon Component is designed to include/attract vegetation to areas. You can add it to any GameObject and save it with a prefab.

You configure radius and a fallout curve. The curve controls the density from the center of the area and out to the edge. This can be used to add trees, plants or objects to an area. The included vegetation will still react to any configured rules, terrain textures, height, steepness, noise etc. And can also be culled by other VegetationMasks.

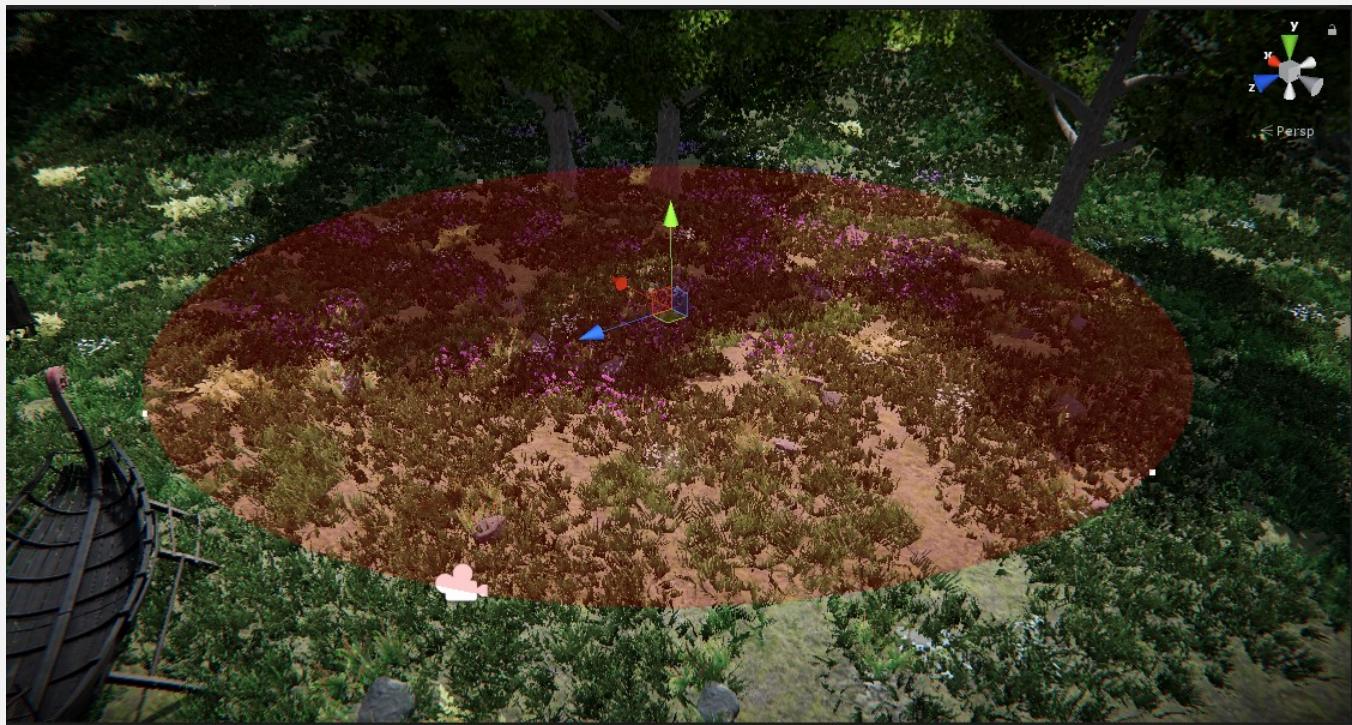
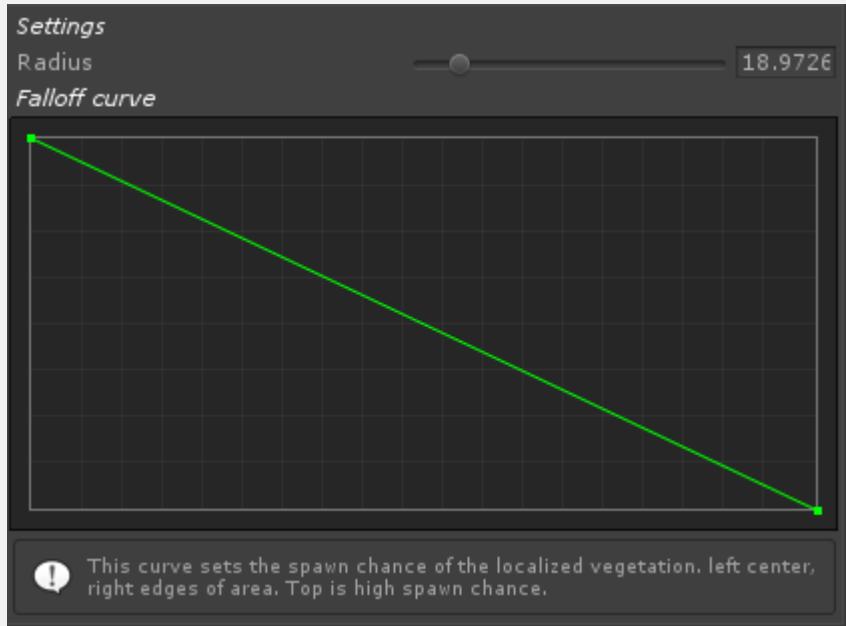


Flowers in the field here is placed using a Vegetation Beacon



SETTINGS

Vegetation Studio



Scenerview Gizmo showing the area/radius of the vegetation beacon

RADIUS

This sets the effect radius of the Vegetation Beacon.

FALLOFF CURVE

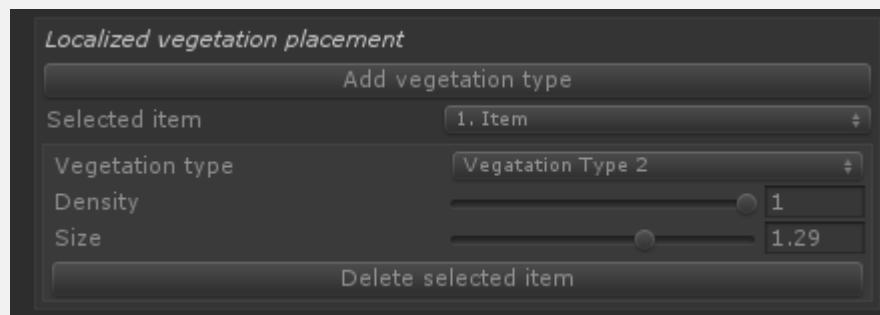
The falloff curves sets the density of the included vegetation types. Left of the curve is the center of the area and right the edges in all directions. Top is 100% of the normal density of the Vegetation Item. Bottom is 0 spawn chance.

LOCALIZED VEGETATION PLACEMENT

Localized vegetation placement is used to include a vegetation type the beacon area. This could be used to introduce a new plant in a area, flowers in a garden etc. that does not spawn in the rest of the environment. In order to use enable the Include vegetation checkbox and add one or more Vegetation Types. The Vegetation item you want to spawn will have to be set with the same Vegetation Type ID in the Vegetation System component.

The vegetation item will be spawned inside the area with all normal rules set in the Vegetation System inspector.

The density and size of these rules can be overridden on a mask to mask basis. This can give the effect of plants growing over time etc. These settings can be set run-time. This could allow you to switch plants in a field, make them bigger etc.



In order to configure a plant to be used for vegetation masks you select the plant in the vegetation system inspector and at the bottom enable the Use Vegetation Mask checkbox and select an ID. Multiple plants can have the same ID.



▼ Vegetation Masks

Vegetation masks

Use with vegetation mask

Vegetation type

COLLIDER SYSTEM

The Collider System Component creates run-time colliders for tree, objects and large objects. It needs to be added to the same GameObject as the Vegetation System Component.

Collider Info

Layers

Debug Info

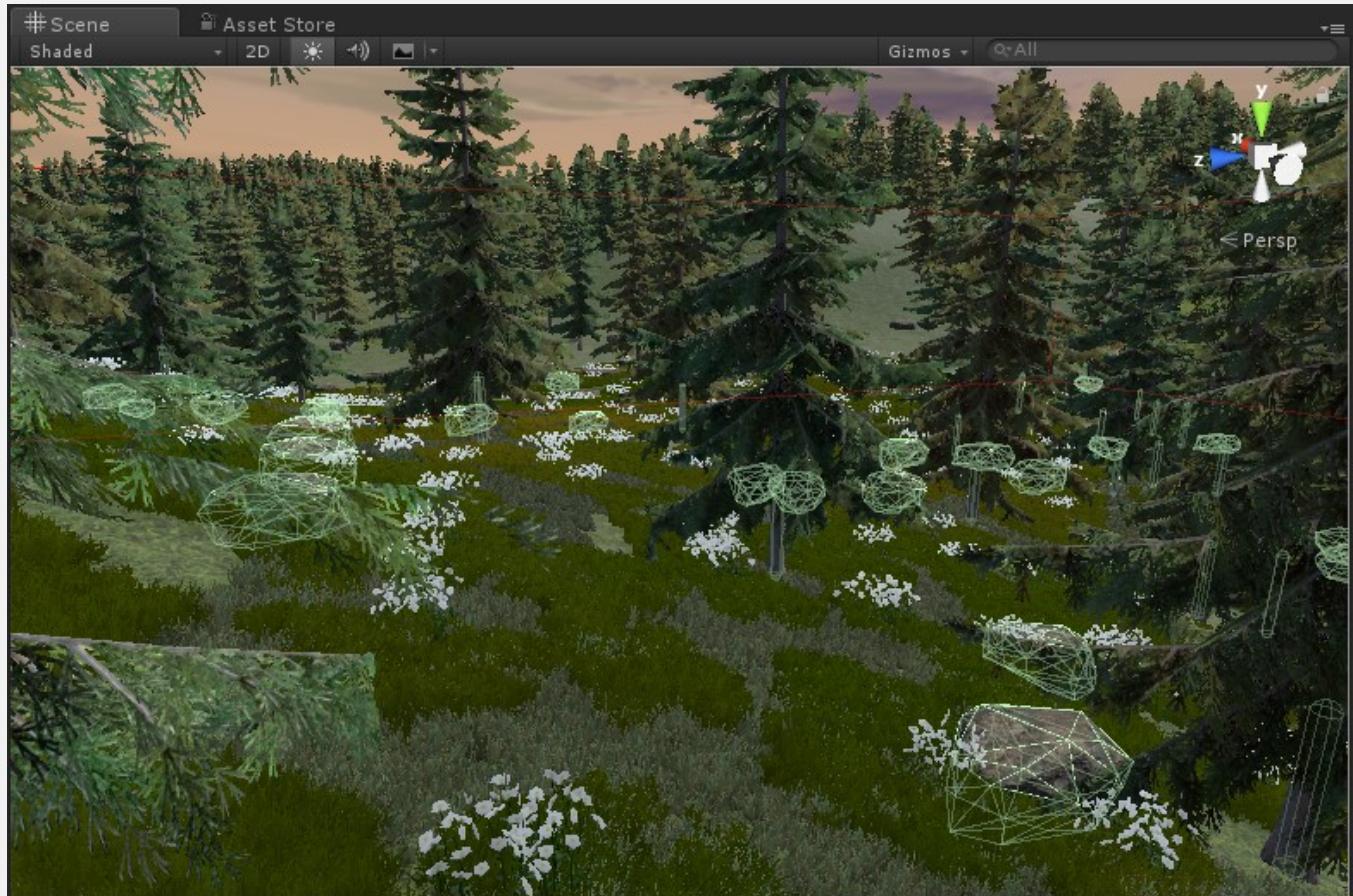
Configure colliders on Vegetation Items

Collider events



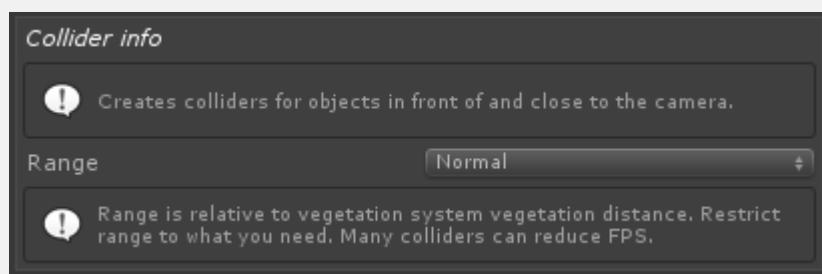
Collider System Inspector

Vegetation Studio



Scenerview showing current mesh and capsule colliders on rocks and trees.

COLLIDER INFO



COLLIDER SYSTEM

Collider system setting is used to decide the selection process for colliders. Currently only 2 modes are available.

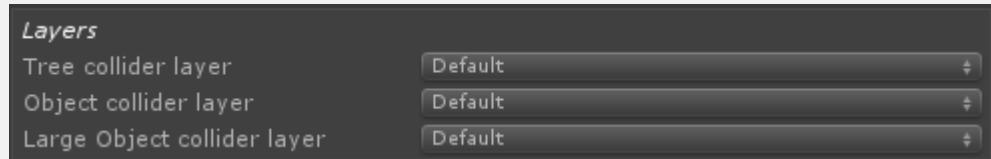
- Disabled
No colliders are instanced.
- Visible
All colliders on Vegetation Items with enabled colliders are instanced when visible. This also includes items behind camera on partial visible cells.

RANGE

Visible colliders within range are instanced/destroyed as camera moves around the scene. Distance is based on Vegetation distance on Vegetation Studio Component. Normal is 20%. Long 40% and Very Long 100% of range. Make sure you limit the range to what you need as it requires time to create and handle colliders.

LAYERS

Select the layer for the created colliders, each Vegetation Item Type can have a separate layer.

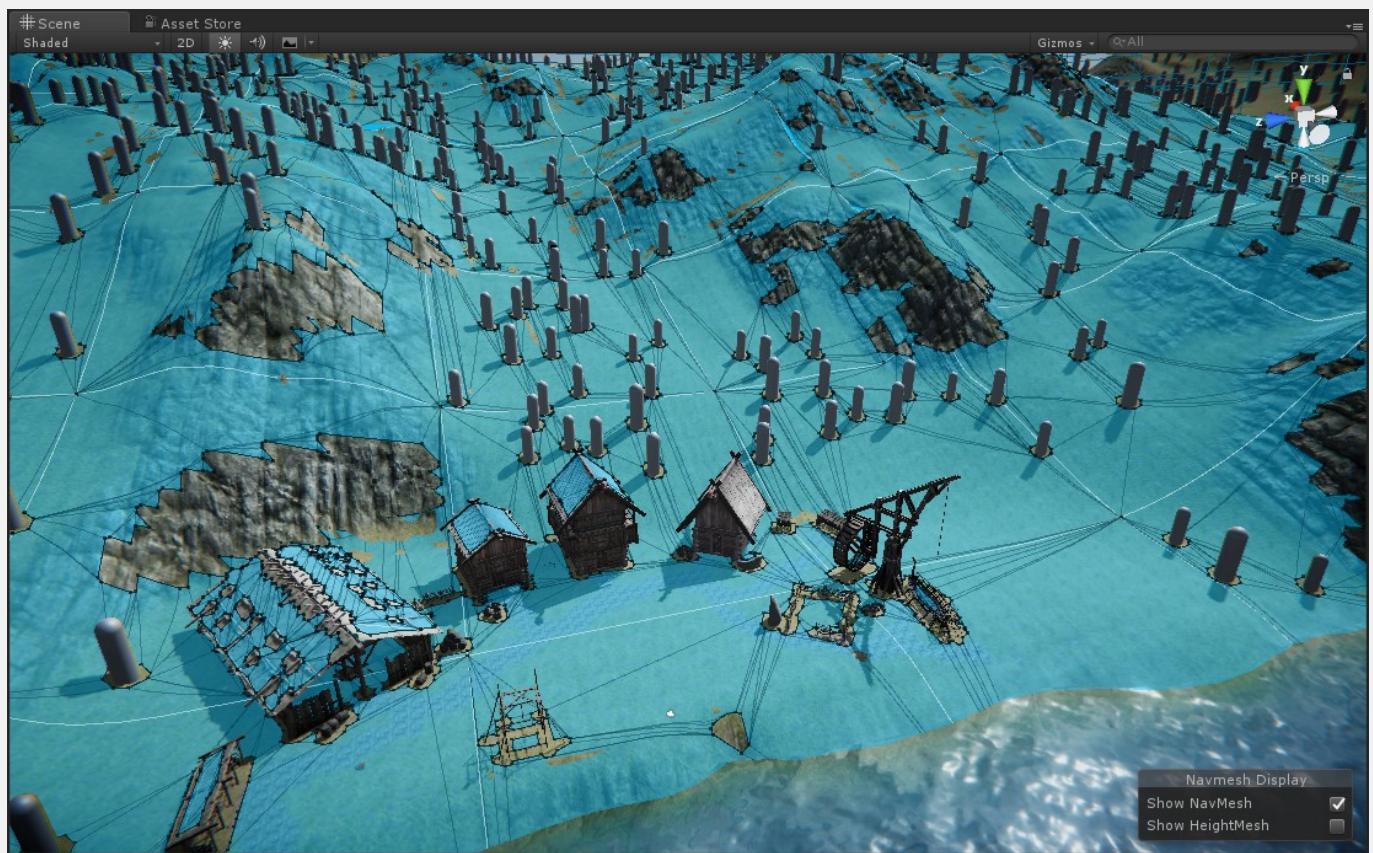


NAVMESH

It is possible to bake out all colliders configures on the individual VegetationItems. This allows you to use those to generate an Unity NavMesh. When navmesh is baked you can remove the colliders from the scene.

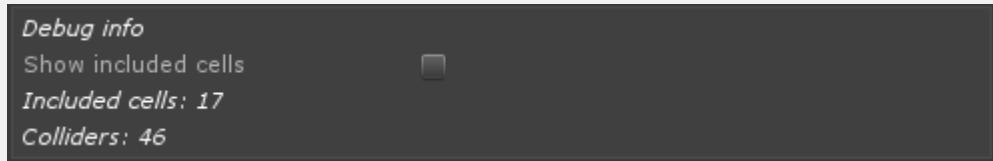
There is 2 options for exporting the colliders to scene. Bake as colliders and bake as scene mesh. The mesh export will convert all colliders to meshes. This is what the Unity Navmesh uses to bake.

Vegetation Studio



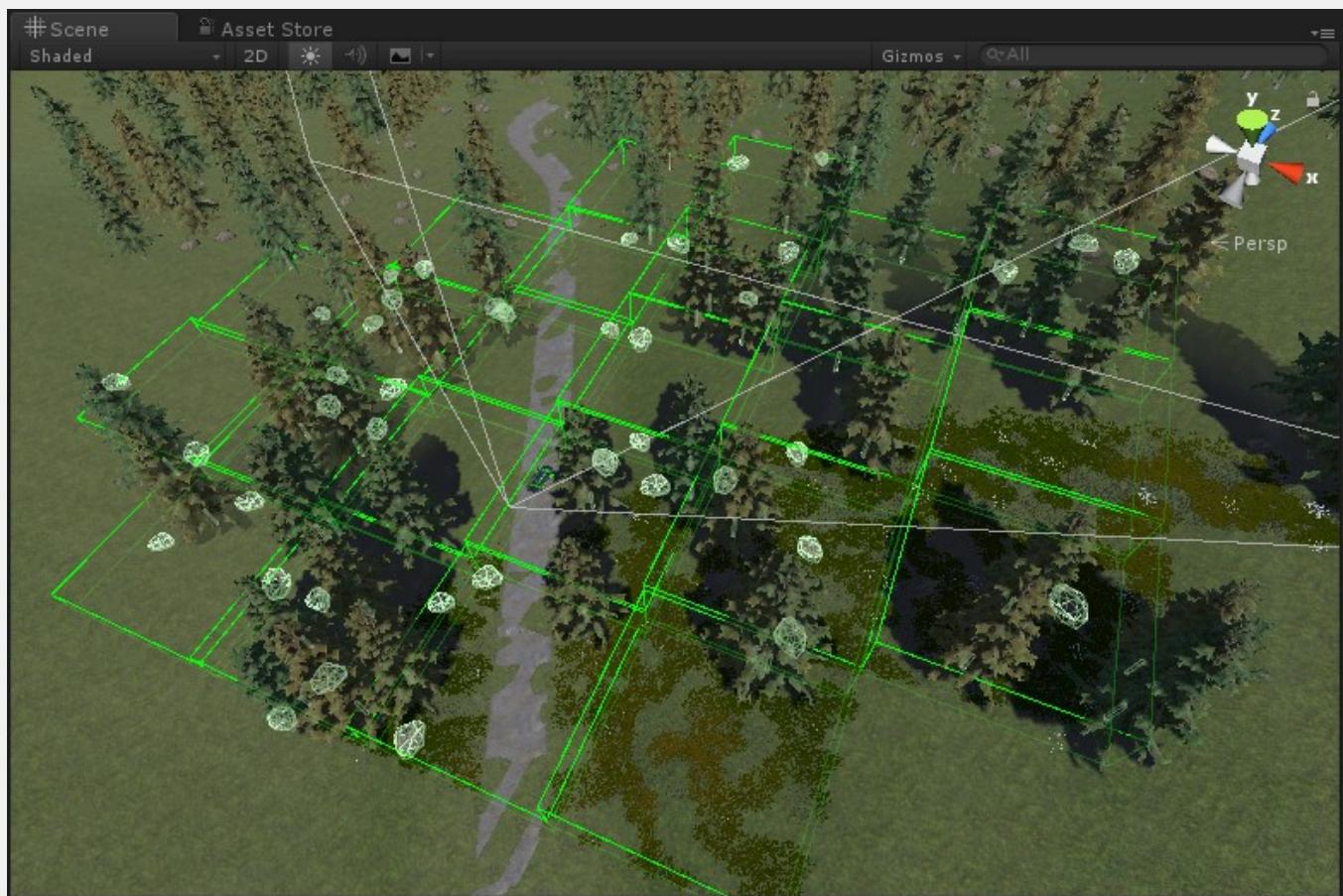
DEBUG INFO

Vegetation Studio



Option Show included cells will show the selected vegetation cells used to create run-time colliders. Selection is based on camera viewpoint and Range setting.

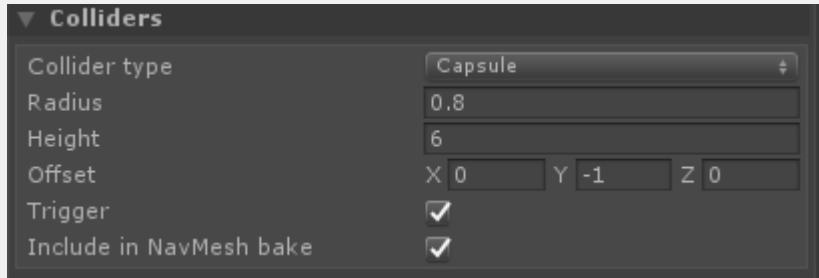
Included cells and colliders are based on current viewpoint.



Scenerview showing terrain in play-mode. Vegetation cells within Normal range.

CONFIGURE COLLIDERS ON VEGETATION ITEMS

In order to configure the colliders for each Tree, Object or large object go to the **Vegetation tab** on the Vegetation Studio Component.



COLLIDER EVENTS

There are 2 events you can connect to to get informed when a run-time collider is created. Use them to add any custom components you need to the collider GameObject. Like dynamic obstacles for path finding or resource scripts for gathering etc.

```
public delegate void MultiCreateColliderDelegate(Collider collider);
public MultiCreateColliderDelegate OnCreateColliderDelegate;

public delegate void MultiBeforeDestroyColliderDelegate(Collider collider);
public MultiBeforeDestroyColliderDelegate OnBeforeDestroyColliderDelegate;
```

To connect to the events.

```
void Start()
{
    colliderSystem.OnCreateColliderDelegate += OnCreateCollider;
    colliderSystem.OnBeforeDestroyColliderDelegate += OnBeforeDestroyCollider;
}

void OnCreateCollider(Collider collider)
{
    //Your code here.
}

void OnBeforeDestroyCollider(Collider collider)
{
    //Your code here.
}
```

TOUCH REACT SYSTEM

Vegetation Studio has a Touch React system that allows grass and flowers to bend when in contact with selected meshes or colliders in the scene. To enable this add a TouchReactSystem component to a GameObject in the scene. There should be only one of these components in the scene. When adding Vegetation Studio to the scene one is created by default.

If you are not planning to use touch react system just disable the component or remove it.

The touch react capability can be added to Objects or Large Objects spawned by the VegetationSystem component or any GameObject with a collider or Mesh.

It is also possible to do advanced rendering using a custom material to a layer invisible to the main camera. See the Car track guide linked below.

Settings Tab

Editor Tab

Debug Tab

Adding touch react to your scene

Touch React Collider Component

Touch React Mesh Component

Requirements

Video showing a car set up with a touch material on a Trail Renderer to create a persistent track in the grass. See this **guide** for a more detailed look at setting up the car tracks.



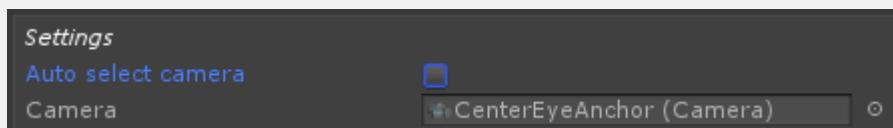
By adding a TouchReactCollider component to the boat you can force the grass to only grow up to the boat.

SETTINGS TAB



AUTO SELECT CAMERA

By default Camera.MainCamera is used for touch React area selection. Disable checkbox to select camera manually. The area around the camera is rendered to a TouchReactBuffer that is used by the custom grass shader.



TOUCH REACT LAYER

Select a layer not seen by the game cameras. This layer is used to render meshes and colliders to the touch buffer.

BUFFER RESOLUTION

Select the resolution of the touch buffer. Larger buffer gives better resolution on large areas but will take more time to render.

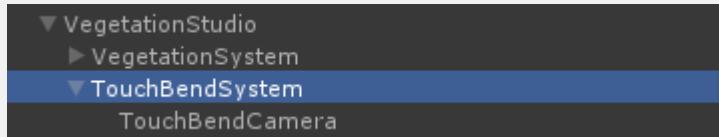
AFFECTED AREA

Affected area is the ortho size of the camera used to render the touch buffer to GPU. a size of 50 will affect 25 meters in radius from the camera.

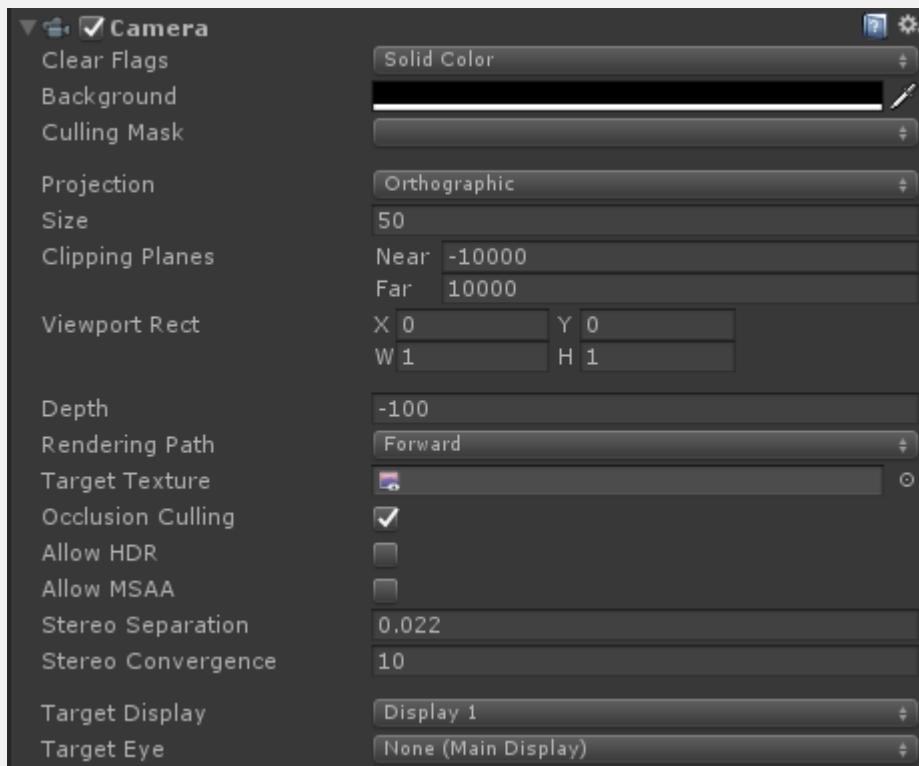
EDITOR TAB



Disable the “Hide TouchReact Camera” checkbox to show the Camera rendering the touch buffer in the inspector hierarchy.



The camera is created and configured by the TouchReactSystem component. There should normally be no need to change anything on the camera.



DEBUG TAB



If you enable the “Show colliders/meshes” checkbox everything that will bend the grass will show up as green in the sceneview.



Touch react debug mode enabled and disabled

COLLIDERS AND MESH

Count of colliders and meshes include all added by normal GameObjects. Vegetation Items rendered instanced to the touch buffer will not show up in the count.

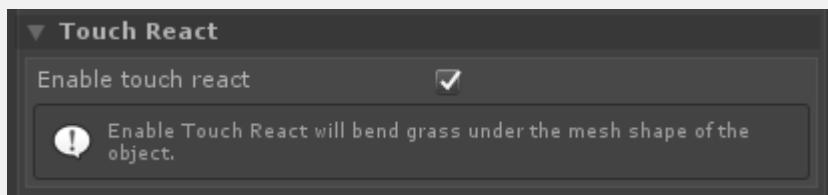
ADDING TOUCH REACT TO YOUR SCENE

VEGETATION ITEMS

Touch react can be enabled on any Object or Large Object added to a VegetationSystem Component. The objects mesh will be drawn instanced direct to the Touch React Buffer. To enable just select the object in the Vegetation System Component.



Enable “Enable touch react” checkbox to enable.

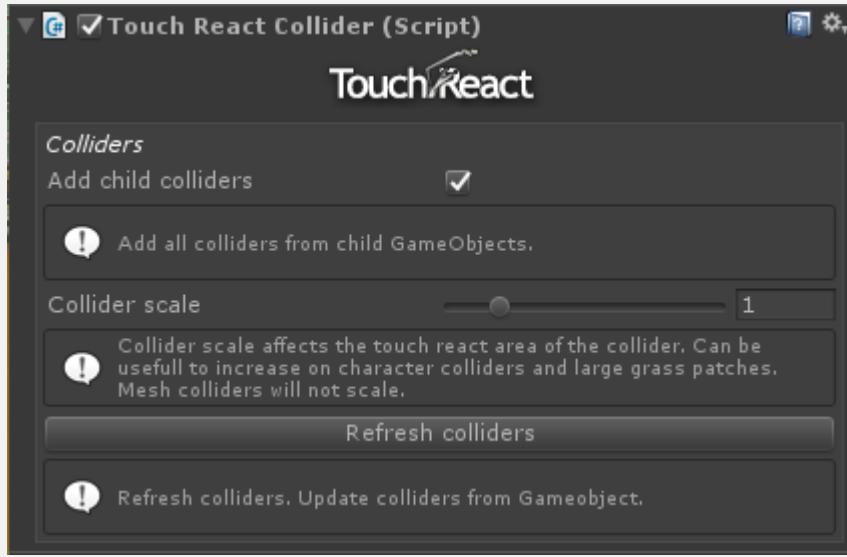


 Rock spawned with vegetation system with and without touch react enabled.

NORMAL GAMEOBJECTS

Any GameObject in the scene with a Collider or a MeshFilter can be used with touch react.

TOUCH REACT COLLIDER COMPONENT



ADD CHILD COLLIDERS

With Add child colliders checked the component will add and render all colliders in the GameObject and children to the Touch React buffer.

When disabled only colliders on the GameObject will be used.

COLLIDER SCALE

It can sometimes be useful to scale up the effect of colliders. In cases with many and small colliders like ragdolls you will get better effect with a bigger scale of the colliders rendered to the buffer. To get head and body overlapping etc. Play with the scale until you get the effect you want.



Effect of touch react enabled on a character with a ragdoll and colliders. The scale is increased a bit in order to properly hit the vertexes of the grass



Same scene with touch react turned off. You can almost not see the character in the grass.

REFRESH COLLIDERS

Press refresh colliders if you change or update the colliders run-time.

```
//get a reference to the TouchReactCollider and call  
touchReactCollider.RefreshColliders()  
//to do the refresh in script.
```

TOUCH REACT MESH COMPONENT



Add the TouchReactMesh collider to any GameObject with a MeshFilter. The mesh in the mesh filter will be rendered to the touch react buffer and grass will bend.

REQUIREMENTS

In order for touch react to work with grass or flowers they need to be using Vegetation Studios custom grass shader. You can add Texture 2D grass, plants or Flowers direct to the Vegetation Studio Component or customize them using the **Grass Patch Generator** that makes mesh patch



grass patches with built in LOD.

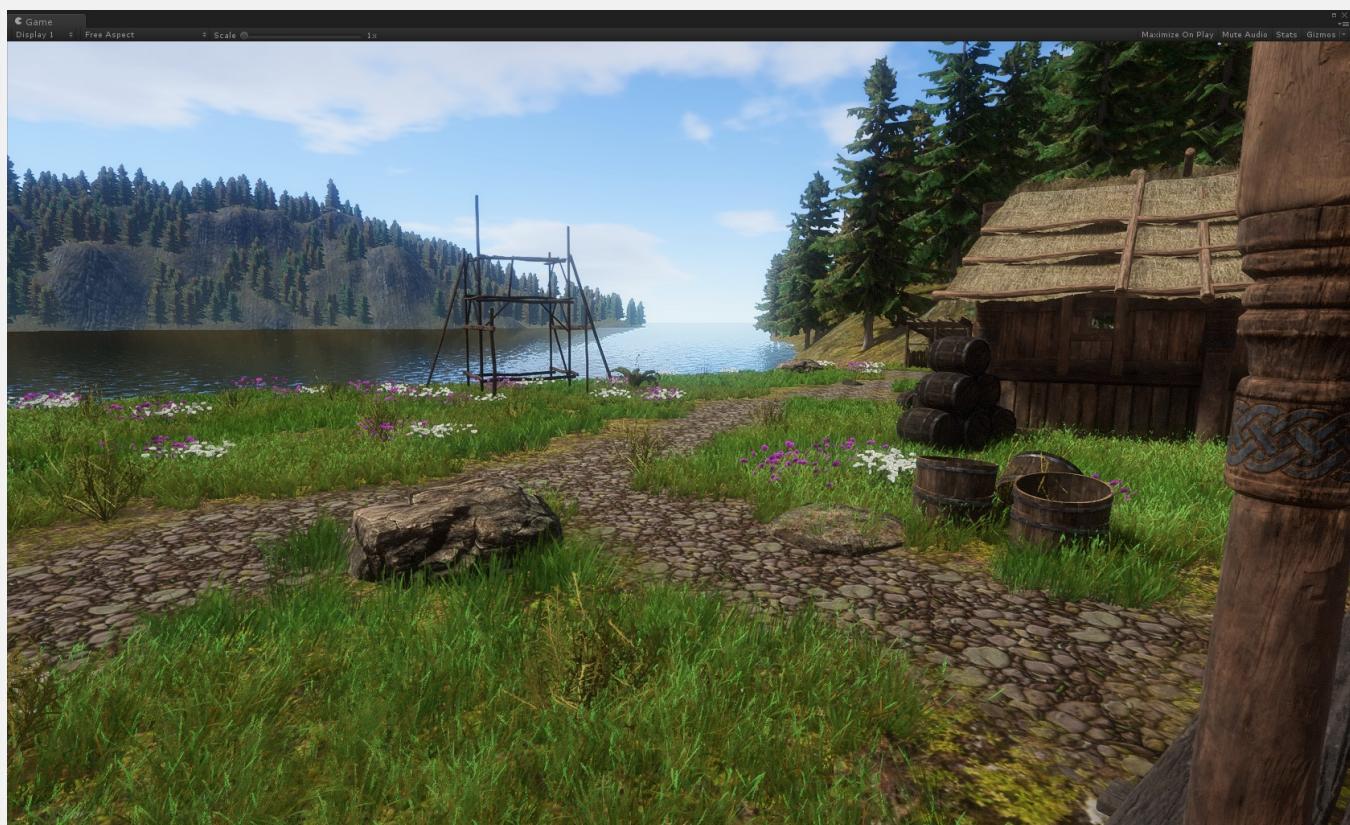
BILLBOARD SYSTEM

Vegetation Studio includes a billboard system that can create a billboard on any tree model added to the VegetationSystem Component. These are used for long range when displaying trees in the terrain. Billboard creation is automatic when adding a tree prefab in editor mode.

In order to use the billboard system add a BillboardSystem Component to the same GameObject as the VegetationSystem Component. This is done by default when adding Vegetation Studio to a scene.

If billboards are not needed you can disable or remove this component.

Billboards are highly optimized and can render up to 10 000 billboards per drawcall.



Billboards in distance.

Vegetation Studio



Billboards on a ship simulator scene. 7km visible tree range.

▼ Vegetation Global Settings

Vegetation distance: (meter)

Vegetation distance	<input type="range"/>	100
Additional mesh tree distance	<input type="range"/>	200
Additional Billboard tree distance	<input type="range"/>	20000

Total vegetation distance: 20,300 meter

Billboard types

Surface Shader

Normals

VR Support

Settings Tab

Editor Tab

Debug Tab

Adding billboards to Vegetation Item trees

Changing lighting model Linear/Gamma

BILLBOARD TYPES

2D

There are 2 types of billboards used in Vegetation Studio. The basic billboard will render 8 images of the tree in different angles. Based on the cameras view of the tree the correct image will be selected from the Atlas.



3D

The 2nd version will render images of the tree from various angles including from above.

3D billboards are still in an experimental stage and are not advised for production. The end functionality is still not decided. An even higher sample resolution might be required for a better result. Images from below is also in test to show better billboards when looking up at billboards on high mountains etc.



SURFACE SHADER

The billboard shader is created as a Unity Surface Shader and should work in most light conditions.

Vegetation Studio



Vegetation Studio



NORMALS

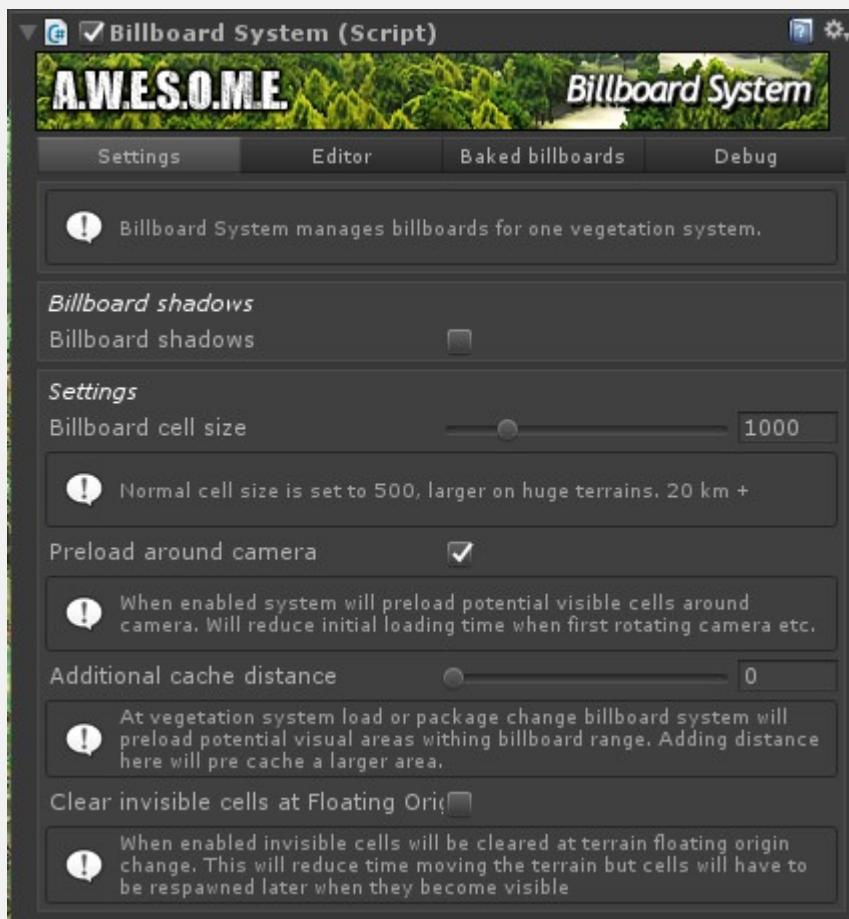
In addition to a normal image a normal map atlas is created to light the billboard better.



VR SUPPORT

Vegetation Studio billboards support single pass VR and does not have the strange rotation and bend behavior of some other billboards.

SETTINGS TAB



CUTOFF

The cutoff parameter will adjust cutoff level for the alpha channel of the generated billboards. On scenes with a huge tree visibility it could be useful to increase this value a bit to remove darker

outlines in the distance.



BILLBOARD CELL SIZE

Billboards are grouped in cells for batched rendering and culling on larger terrains. On huge terrains with a lower tree density it could be useful to experiment with a larger cell size.

PRELOAD AROUND CAMERA

with preload around camera the BillboardSystem will spawn all billboard cells within billboard visible range in all directions. This will eliminate loading time the first time you turn the camera.

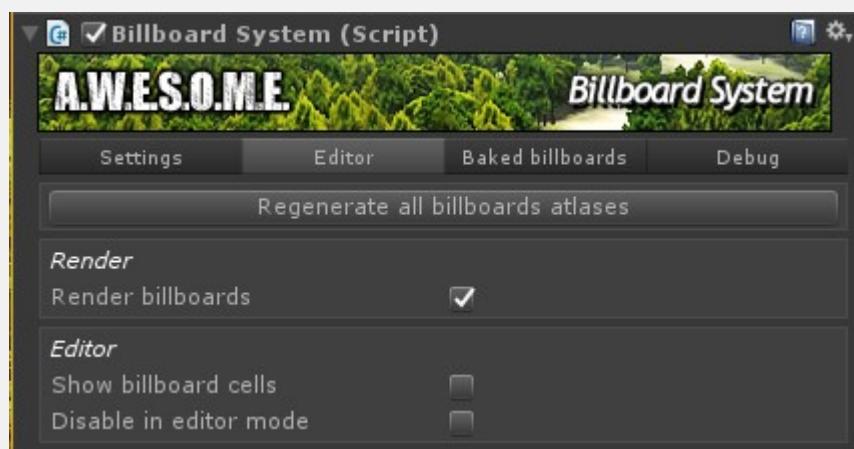
ADDITIONAL CACHE DISTANCE

Additional distance will pre cache billboards at a longer distance in when initializing the billboards.

CLEAR INVISIBLE CELLS AT FLOATING ORIGIN CHANGE

When system is used with a floating origin system and terrain is moved during gameplay it has to recalculate billboard positions. On larger terrains it will speed up the process to only translate the currently visible billboard cells and clear the cache of all other. The other cells will spawn normally when visible later.

EDITOR TAB

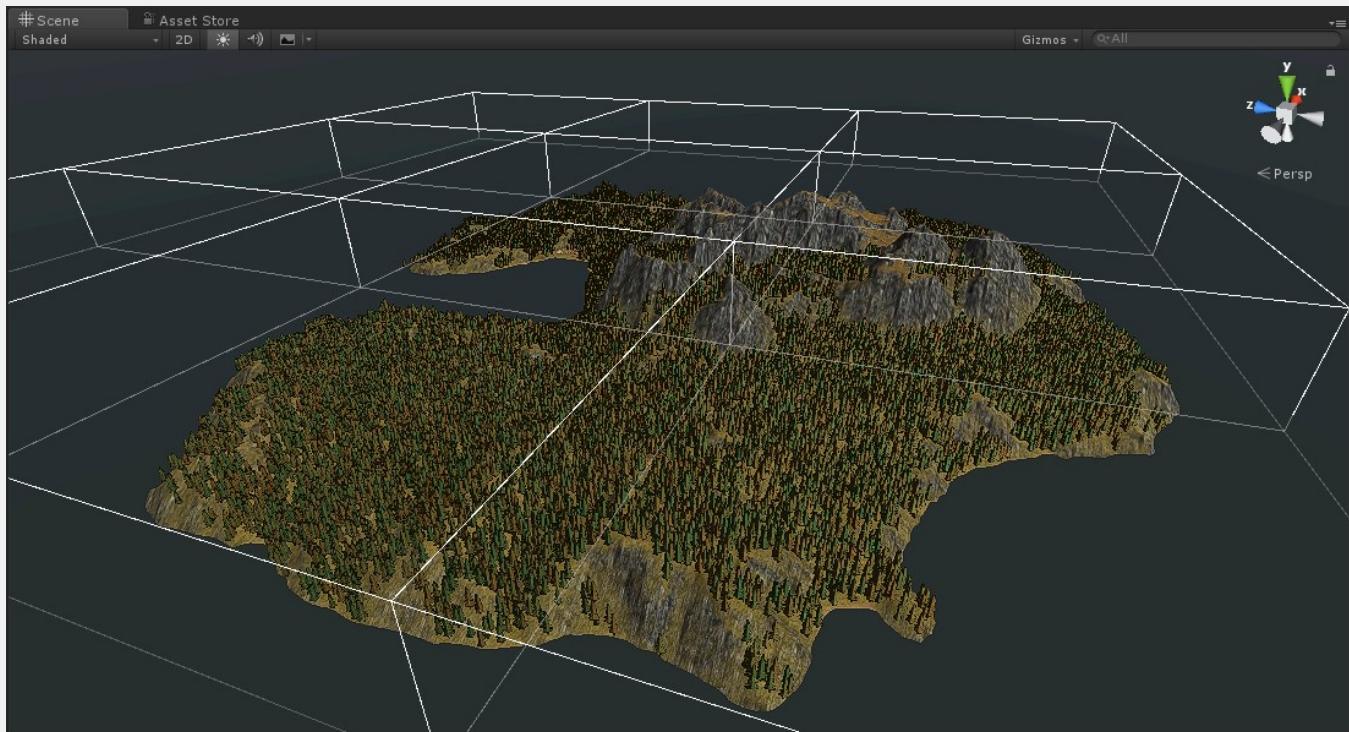


Render billboards will turn on and off billboard rendering in the scene. They are still created and

loaded as you move the camera. Only rendering loop is disabled.



For use on large terrains billboards are culled in cells as well as camera distance. This is done to optimize loading on larger scenes. We have used system with scenes with multiple terrains with a total terrain size of 80×80 km.



The disable in editor mode allows you to disable billboard loading and rendering while in editor mode. When entering playmode everything will work as normal.

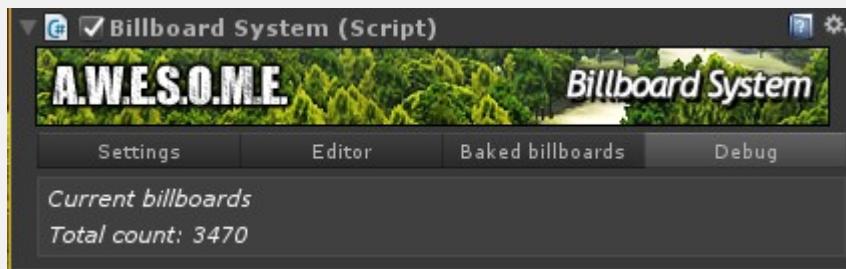
BAKED BILLBOARDS TAB

There is an option to bake out all billboards to unity meshes. Depending on the billboard cell size and tree density up to 10 000 billboards will be saved in each mesh. This will reduce loading time

but the billboards will no longer be dynamic if you change rules, mask out areas etc. Enable save to project to save the meshes and materials to the project. This allows you to save this to asset bundles with the terrain.



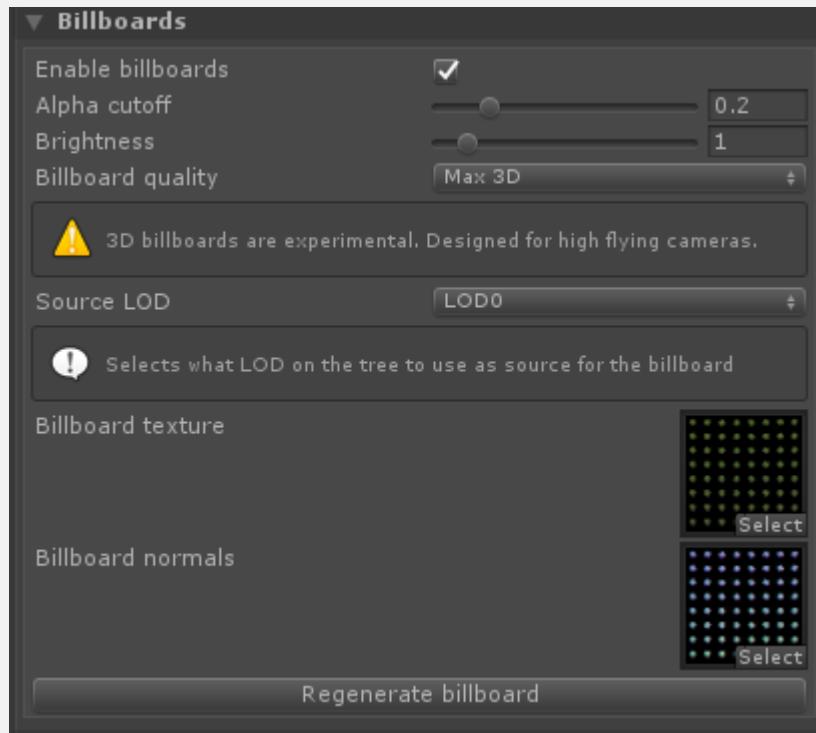
DEBUG TAB



The total count of billboards is the number of currently loaded billboards. Load state is dependent on the billboard visibility range set in the VegetationSystem component.

ADDING BILLBOARDS TO VEGETATION ITEM TREES

Vegetation Studio



You can add billboards to any tree in Vegetation Studio by enabling the “Enable billboards” setting on trees in the VegetationSystem component. This is done by default for new trees.

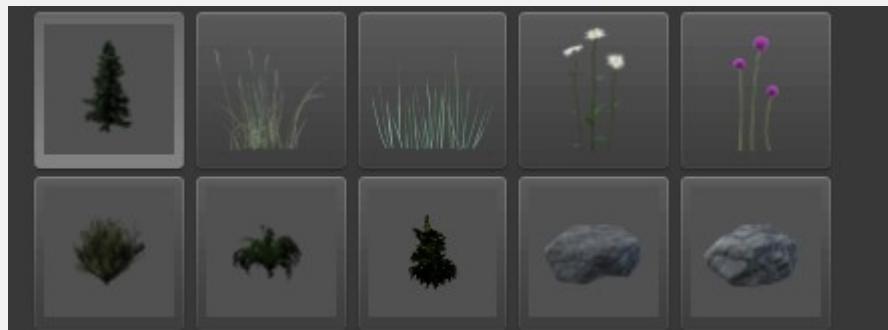
There are currently 6 different quality settings. 3 for 2D and 3 for 3D billboards.

Normal, High and Max. The settings control the size of the Atlas texture and sizes are 1024, 2048 and 4096.

CHANGING LIGHTING MODEL LINEAR/GAMMA

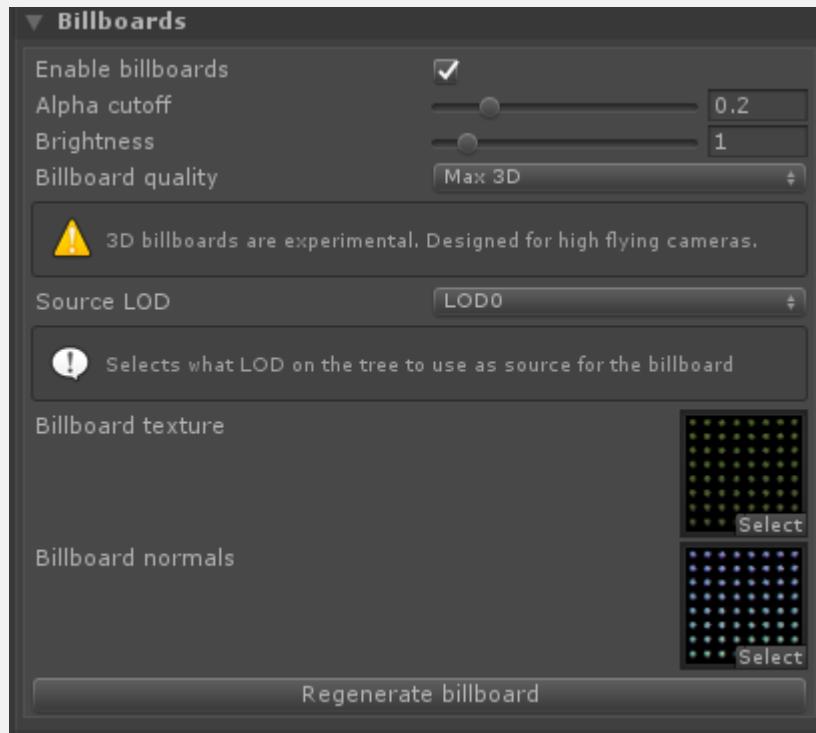
When changing lighting model in Unity you need to manually regenerate the billboards of all trees in the package.

Go to the Vegetation Tab in the VegetationStudio Component, select the trees and press “Regenerate billboards”



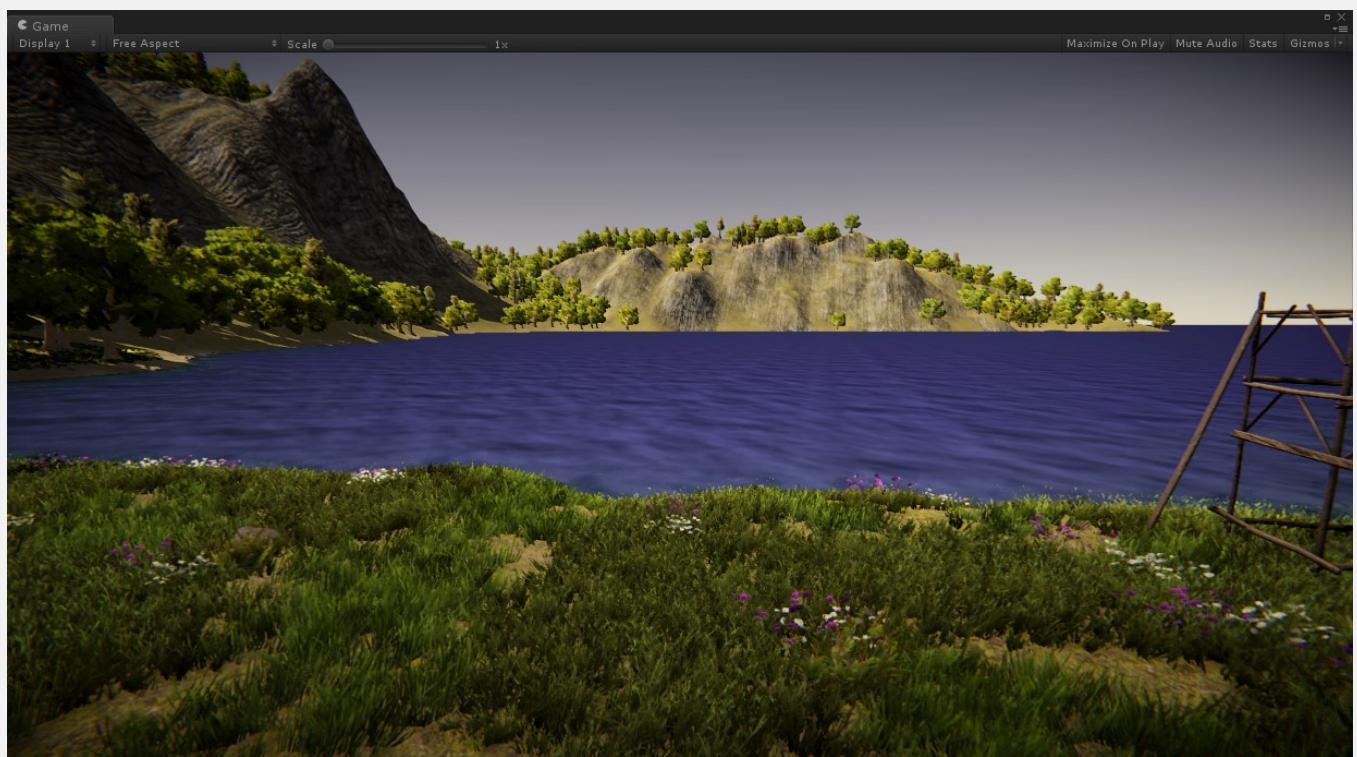
Select the trees in the VegetationSystem Component

Vegetation Studio



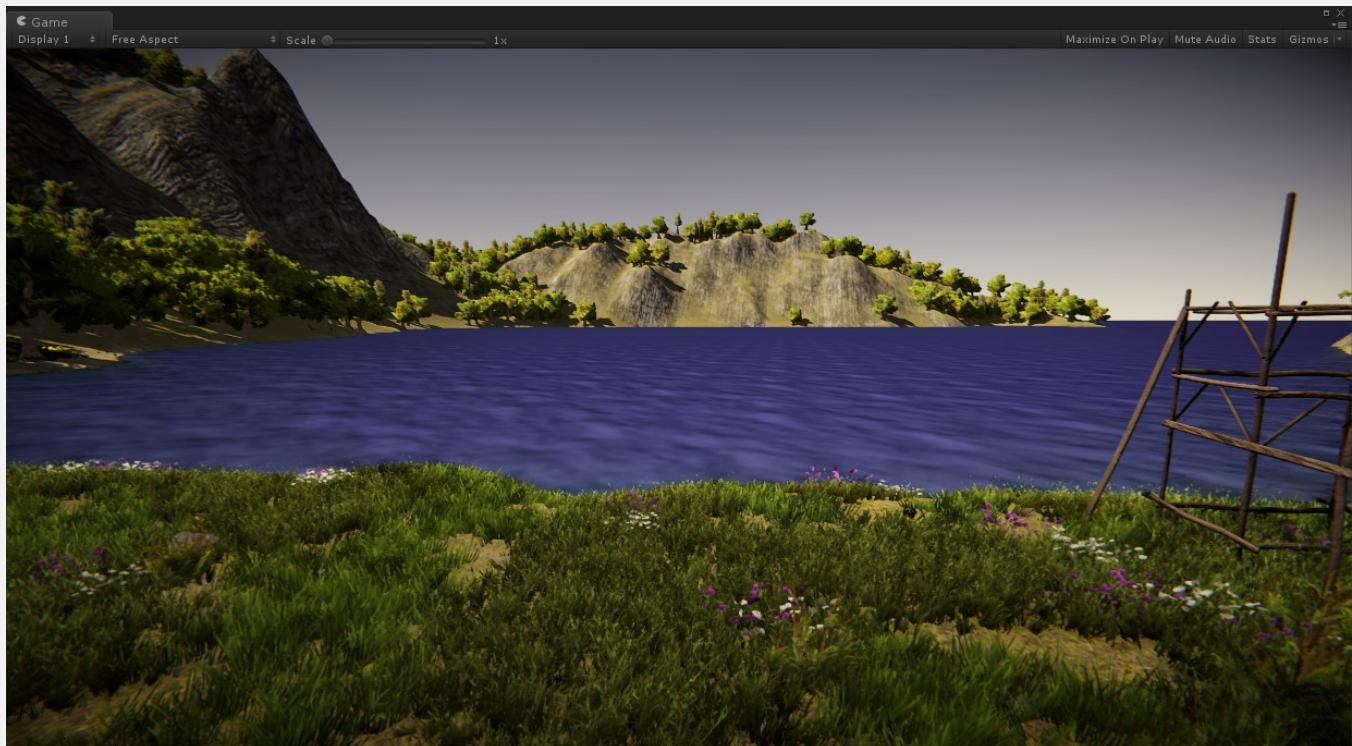
SHADOWS

Billboard shadows is exposed as a setting on BillboardSystem Components



Vegetation Studio

Shadows disabled



Shadows enabled

RUNTIME PREFAB SPAWNER

The Runtime Prefab Spawner component is designed to instantiate prefabs related to the trees or objects in Vegetation Studio. It allows you to assign a prefab to a Vegetation Item. When the item is within range of the camera an instance of the prefab is created at the exact location of the item. This item will be removed automatically when item is out of range again.

The use case of this could be to add effects like falling leaves, insects at flowers, sounds from the old tree, harvesting logic etc. The prefabs can have any scripts attached.

This approach allows you to add special functionality to large amounts of vegetation items with no extra overhead of culling and processing the effect gameobjects in the hierarchy.

You add it to the GameObject with the VegetationSystem component and configure. Multiple components is possible.

Settings

Editor

Debug Info

API

Vegetation Studio

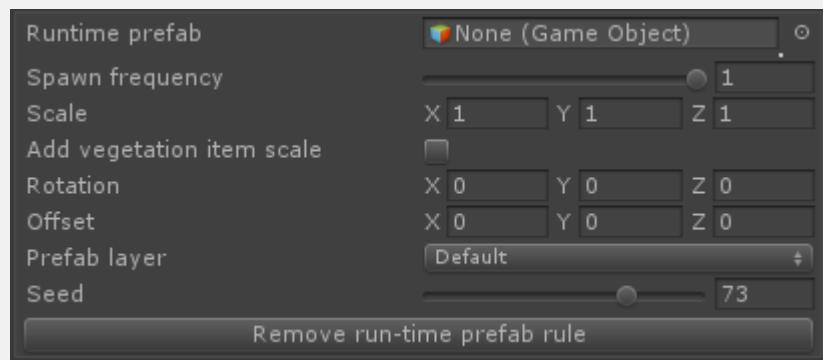


In this example video we added a prefab with a falling leaves particle effect to one of the tree models. As any tree of this type gets within range it will instantiate a copy of the effect with the same position as the tree. When it is out of range it is removed automatic. This will allow for effects on huge amounts of trees or plants with no overhead of culling and gameobject hierarchy handling.



Image showing the falling leaves.

SETTINGS



SELECT VEGETATION ITEM

Select any of the available Vegetation Items from the list. It is possible to add runtime prefabs to any

Plant, Tree, Object or Large Object.

RUNTIME PREFAB

Assign any prefab you want to instantiate. They will be created with the same position, rotation and scale as the VegetationItem

SPAWN FREQUENCY

Spawn frequency is the chance of a Vegetation Item having the effect. A value of 1 will create a object for every Vegetation Item.

SCALE MULTIPLIER

Scale of the instanced object will be multiplied with this scale. Usefull to set particle system size etc.

ROTATION

Instanced object will be rotated with this value.

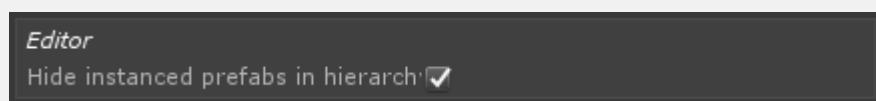
OFFSET

Offset is a local space position offset added to the instanced prefab at run-time.

RANGE

This settings sets the range/distance from camera where you will get the objects created. It set as a factor of the set vegetation distance.

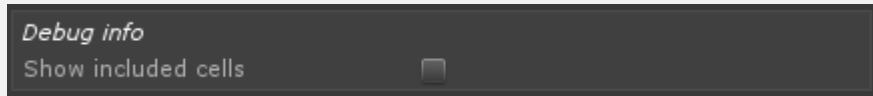
EDITOR



HIDE INSTANCED PREFABS IN HIERARCHY

When enabled the instanced objects are hidden in the scene hierarchy.

DEBUG INFO



SHOW INCLUDED CELLS

Enable “Show included cells” to visualize the cells used for selecting what Vegetation Items we will spawn prefabs for.



API

There are 2 events you can connect to to get informed when a run-time prefab is created. Use them to get a reference and process the instanced objects with your gamelogic

```
public delegate void MultiCreateRuntimePrefabDelegate(GameObject go);
public MultiCreateRuntimePrefabDelegate OnCreateRuntimePrefabDelegate;

public delegate void MultiBeforeDestroyRuntimePrefabDelegate(GameObject go);
public MultiBeforeDestroyRuntimePrefabDelegate OnBeforeDestroyRuntimePrefabDelegate;
```

To connect to the events.

```
void Start()
{
    runtimePrefabSpawner.OnCreateRuntimePrefabDelegate+= OnCreateRuntimePrefab;
    runtimePrefabSpawner.OnBeforeDestroyRuntimePrefabDelegate+=
    OnBeforeDestroyRuntimePrefab;
}

void OnCreateRuntimePrefab(GameObject go)
{
    //Your code here.
}

void OnBeforeDestroyRuntimePrefab(GameObject go)
{
    //Your code here.
}
```

PERSISTENT VEGETATION STORAGE

The persistent vegetation storage component is designed to handle and store persistent vegetation instances for a terrain. It uses a **Persistent Vegetation Storage Package** to store this information. The package is a scriptable object that serializes to the project.

The storage has included painting tools, a bake system that can bake run-time spawned rules to the storage.

In addition to this there is a importer system that imports from the terrain and scene and a API for 3rd party tools to integrate.



Due to the amount of features in this component the documentation is divided on several pages.

Settings Tab

Stored Vegetation Tab

Bake Vegetation Tab

Edit Vegetation Tab

Paint Vegetation Tab**Precision Painting Tab****Import Tab****Custom Importers**

The importers are made using an interface. It is possible to create new that register in the UI.

Terrain Tree Importer**Terrain Detail Importer****Scene Vegetation Importer**

PERSISTENT VEGETATION STORAGE - SETTINGS

In order to work the Persistent Vegetation Storage component needs a **PersistentVegetationStoragePackage**. Create a new package and drag and drop it to the Storage slot in the inspector. The first time a package is added or if the package is initialized for another terrain it will ask you to initialize it.

When initialized it will work for only this terrain with the current vegetation cell size. If you change the cell size you need to initialize the storage again. This will clear all data in the storage.



PERSISTENT VEGETATION STORAGE - STORED VEGETATION

On the stored vegetation tab you will see what is stored in the package.

You can select each vegetation item and see the count and the source. The painting tools, baked vegetation and external API all has its own VegetationSourceID and this will show up on the UI. You can clear the storage per source also. Allowing you to rebake from rules while keeping all manually painted vegetation.

Loading vegetation from a persistent storage is faster than the run-time spawning but it does not allow you to change vegetation spawn rules run-time.

You can also edit the vegetation in the storage with the included tools and API.



In this example you can see there is 1.4 million instances of vegetation. The package is 59mbyte.

PERSISTENT VEGETATION STORAGE - BAKE VEGETATION

In order to bake the results of the run-time spawning rules to the storage package select the vegetation item you want to bake and press the "Bake vegetation from ruleset button". This will create all instances for the terrain and store it in the package.

Use the "Bake ALL" button to bake all vegetation items to the package. This will also disable the "run-time spawn" on each vegetation item. After bake it is loaded form the storage.



PERSISTENT VEGETATION STORAGE - EDIT VEGETATION

This allows you to manually add, remove and edit any tree, object or large object in the persistent storage.

If you edit a items position, scale or rotation the VegetationSourceID will be set to manual edit.

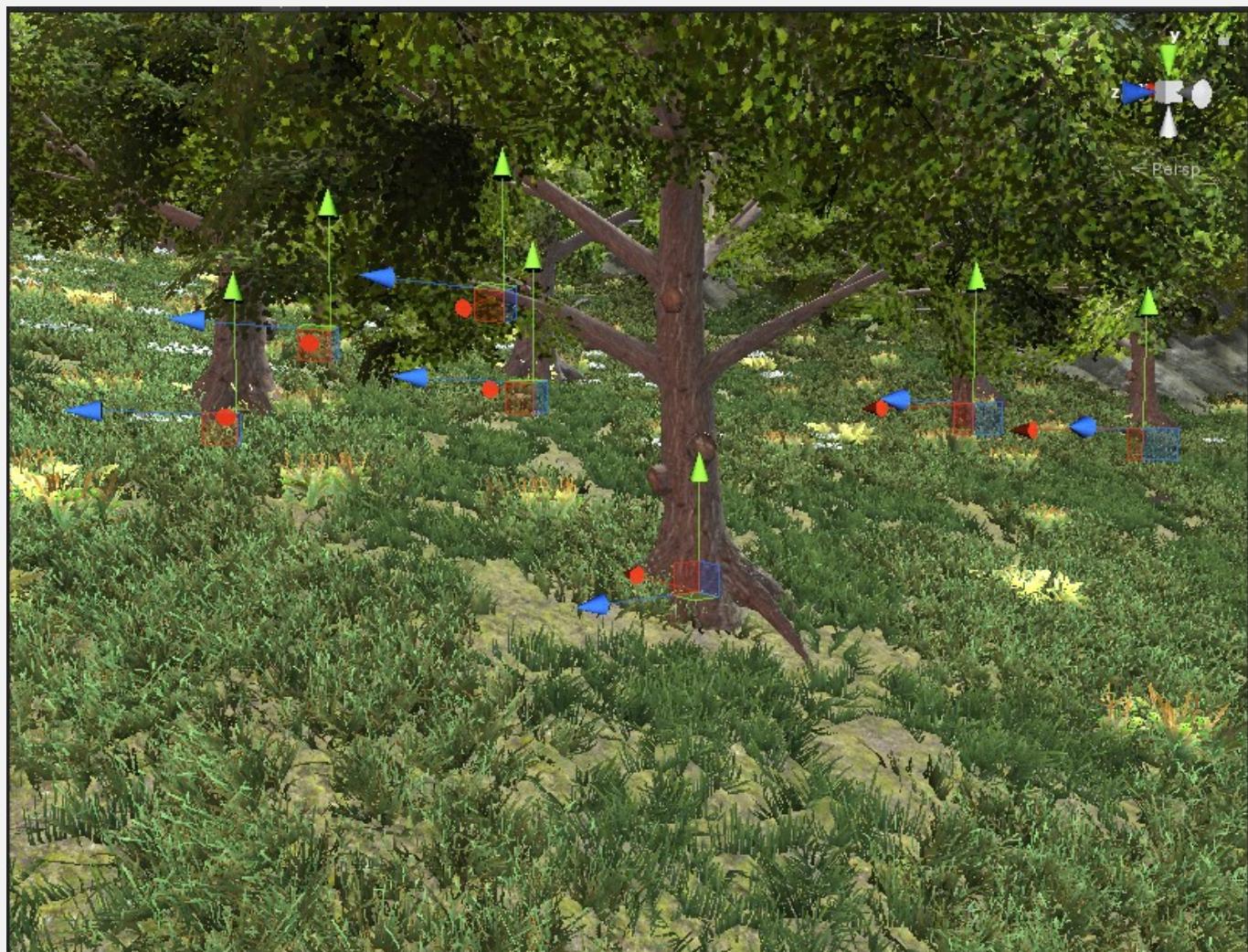


Add new items with Ctrl-click in the terrain. Remove with Ctrl-Shift-Click



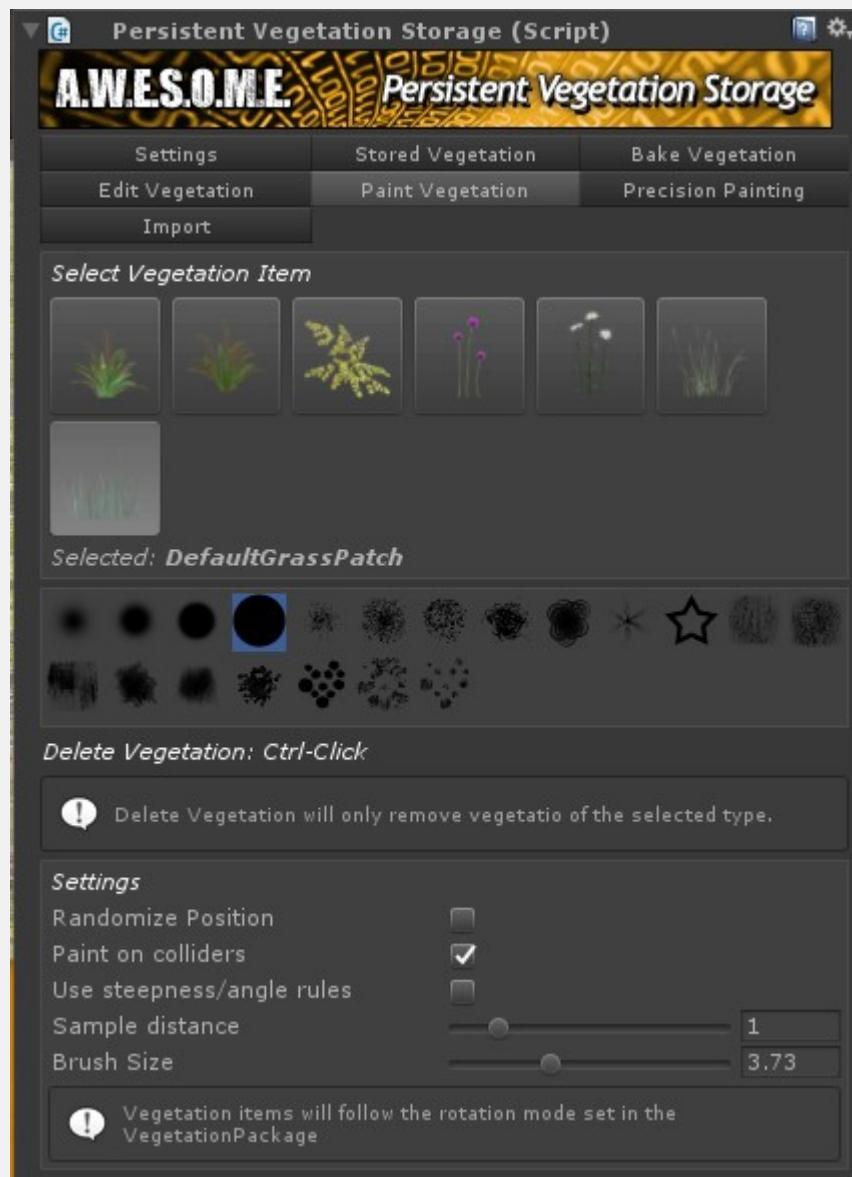
All items within range will get normal unity move, rotate and scale handles. Mode is selected with the normal unity buttons.

Vegetation Studio



PERSISTENT VEGETATION STORAGE - PAINT VEGETATION

The persistent storage has a painting tool for painting grass and plants. It works in a similar way unity terrain painting tool does but with some additional functionality.

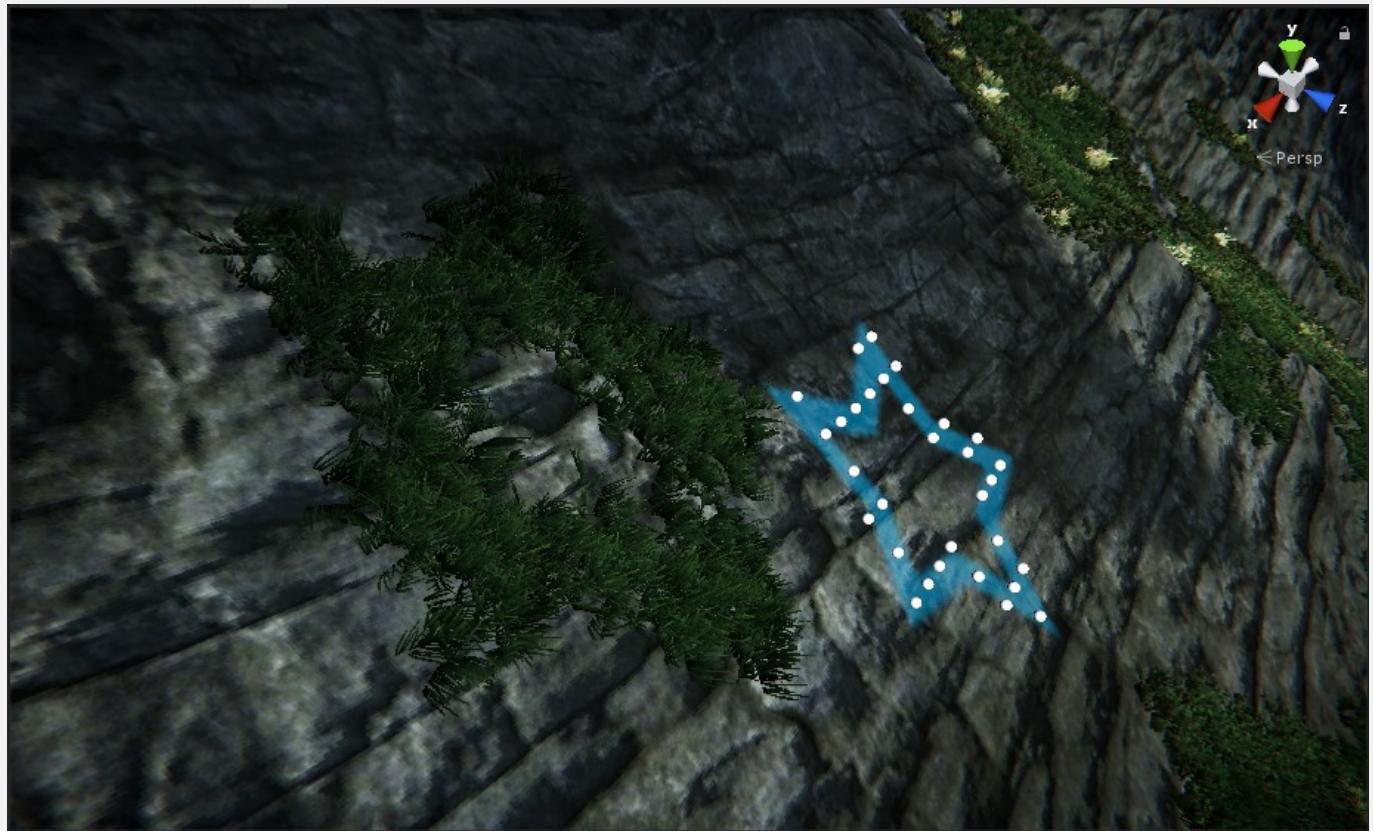


PAINTING

You paint with left click in the map, remove instanced with ctrl-click.

SETTINGS

- Randomize position
will add randomness to the sample positions on top of the point grid you see in the terrain
- Paint on colliders
With this enabled you will be able to paint on any collider in the scene as well as the terrain
- Use steepness/angle rules
With this enabled each vegetation items steepness rules (set in VegetationSystem component) will be applied before painting
- Sample distance
This is the density of the painting
- Brush size
The size of the brush



Vegetation Studio



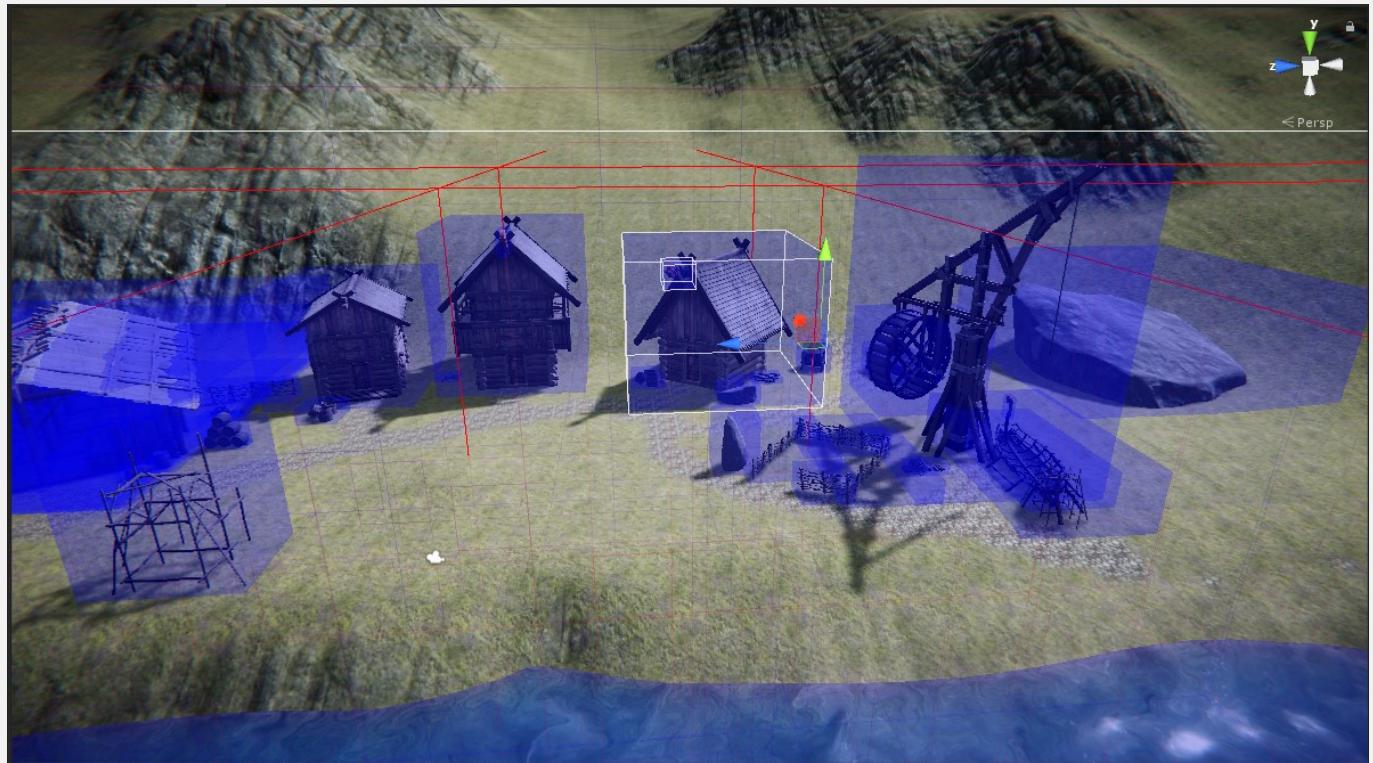
PERSISTENT VEGETATION STORAGE - PRECISION PAINTING

The precision painting tool is similar to the normal painting tools but works a bit different. While the normal painting tools projects a brush from above and down on the terrain and colliders the precision painting tool will use a ray from the camera perspective. The normal of the hit point will be used as up direction for the placed vegetation. This allows for fine tuning the positioning.

The precision tool will allow you to paint on any mesh, even without a collider.



Vegetation Studio



Internally the painting tool is building up an octree of all gameobject with meshes in the scene and manually raycasting these to find the mesh intersection.



PERSISTENT VEGETATION STORAGE - IMPORT

The Persistent Vegetation storage currently have 3 importers. Fore more detail see the page for each importer.



Terrain Tree Importer

Terrain Detail Importer

Scene Vegetation Importer

TERRAIN TREE IMPORTER

The terrain tree importer will show you all trees in the terrain assigned to the VegetationSystem component. You can also see tree count.

Press the “Import trees” button on each of the available trees to import. The trees will be added as VegetationItems and the instances imported to the Persistent Storage.



TERRAIN DETAIL IMPORTER

The terrain detail importer is designed to help you set up a terrain with grass and plants that uses the detail distribution of the original terrain.

It will not import actual instances of grass from the terrain but use the terrain detail info in the terrain as a density mask.

By pressing import all the grass/plant textures will be added as separate VegetationItems in the vegetation package. Each of them with a DetailDensityMask rule set to the index of the Texture in the terrain.

You can now adjust sample distance, rotation, scale until you get the look you are looking for.



SCENE VEGETATION IMPORTER

The scene vegetation importer will help you import existing vegetation and rocks from the scene. It will search the scene and match objects based on filename, tag and layer. Then add the instances to the Persistent Storage.

You create a importer profile with right clicking in any project folder. Choose “Create/Awesome Technologies/Vegetation Importers/Scene vegetation importer settings”. Give the settings a the name you want and assign it to the importer.

The profile can hold many import rules and allows you to re-use this between scenes/projects.

Vegetation Studio

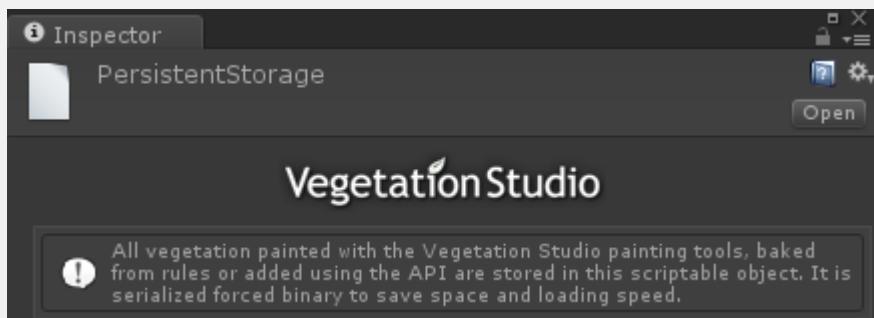


PERSISTENT VEGETATION STORAGE PACKAGE

The persistent vegetation storage package is a scriptable object designed to store vegetation instances for a single terrain. You create the package with right clicking in any project folder. Choose “Create/Awesome Technologies/persistent vegetation/Persistent vegetation storage package”. Give the package a the name you want and assign it to the PersistentVegetationStorage component.

The package is configured to serialize binary. A text serialization will be slow and use way more space and memory. You might have to configure your version control software to store this file binary.

See the **PersistentVegetationStorage** component for more info.



GRASS PATCH GENERATOR

Grass Patch Generator is a tool to produce mesh grass and plant patches with level of detail (LOD) from standard Texture2D images with alpha.

Size, resolution and a range of settings will be saved with the prefab. The resulting prefab can be used with Vegetation Studio as instanced rendered grass with Touch Bend support. Or used as normal GameObjects in Unity with a LOD Group.

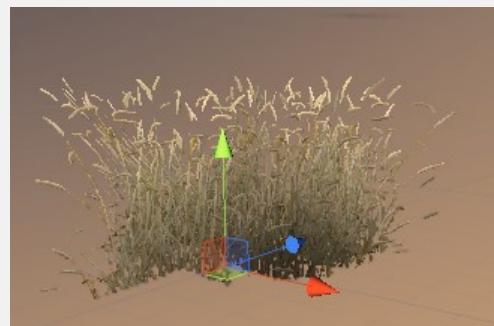
To create a new generator select “Window/AwesomeTechnologies/Add Grass Patch Generator” from the menu in Unity. This will make a new GameObject with a generator component. There is a default grass texture loaded. Add your texture. Set settings as described below in this page. When finished choose one of the 2 options(with and without LOD) to save the prefab. Prefab, mesh asset and Material will be saved in the folder you choose.

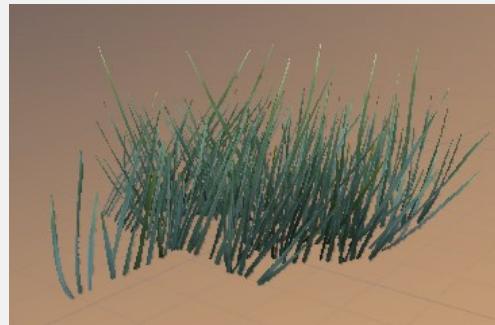
The settings in the box “Shader settings” can be changed runtime in the Vegetation System inspector when used later.

When the prefab is saved the GameObject with the generator is no longer needed and can be removed from the scene.

If you want to save the settings for later use, make a prefab of the GrassPatchGenerator object.

See Guide – **Create a new grass patch** for an example.





Patch Settings

Resolution

Bending

Mesh

Grass texture

Shader Settings

Ambient Occlusion

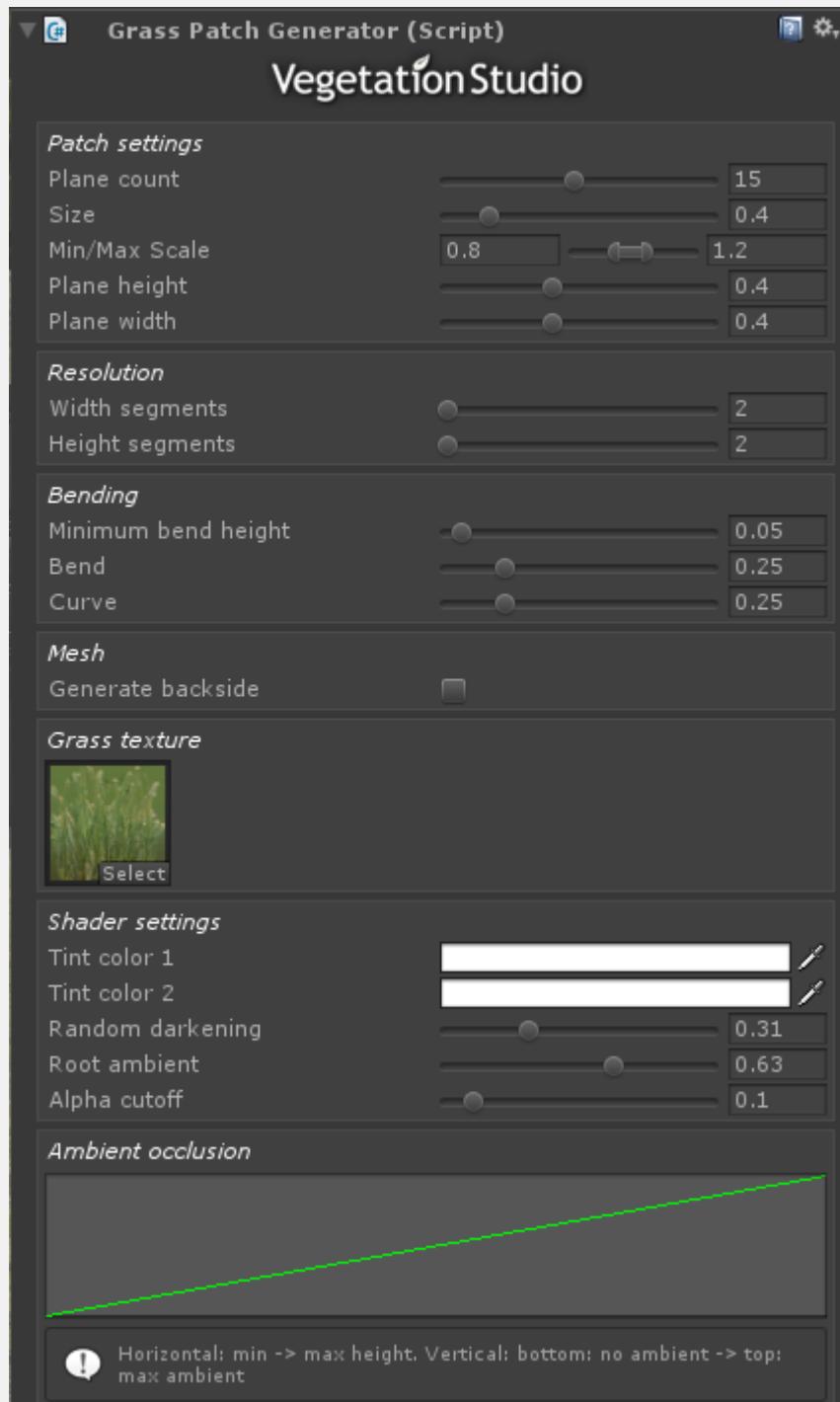
Wind Bending

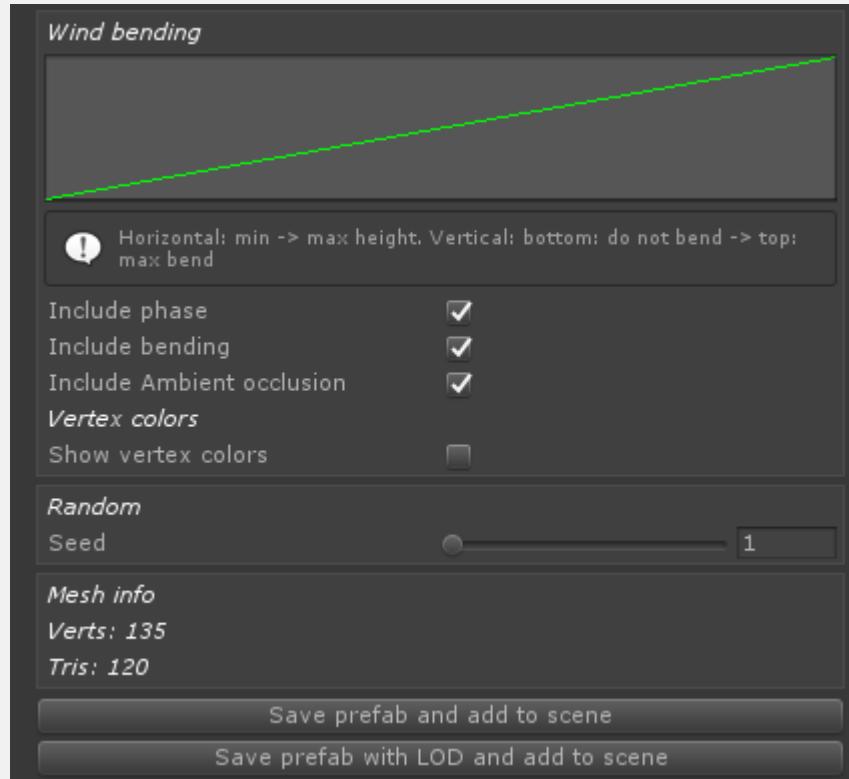
Random

Mesh Info

Generate

Vegetation Studio





PATCH SETTINGS

- Plane count

Number of planes used for the grass patch. Higher plane count gives more grass but also a higher polygon mesh.

- Size

The size of the mesh. This adjust the randomness of the plane center position.

- Min/max scale

Min/max scale of each individual plane. Add a bigger range for more randomness

- Plane Height

Base height of each plane. Make sure the height/width aspect fit the texture you are using.

- Plane width

Base width of each plane. Make sure the height/width aspect fit the texture you are using.



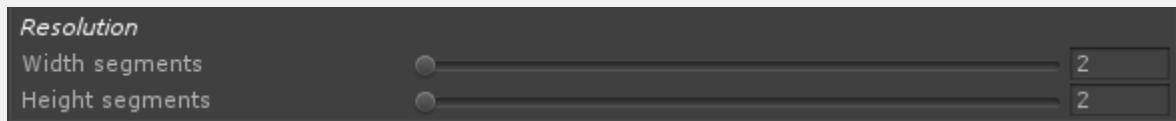
RESOLUTION

- width segments

Number of width segments for each plane. If you want the planes to curve you need a higher count than 2

- height segments

Number of width segments for each plane. If you want the planes to bend you need a higher count than 2



BENDING

- Minimum bend height

This setting will set the minimum height in meters where the planes are allowed to start bending.

- Bend

Bend amount above the minimum height setting

- Curve

Curve amount of each plane.



MESH

Enable this setting to generate a backside for the planes in the grass patch. This is not needed for use with the VegetationStudio grass shader.

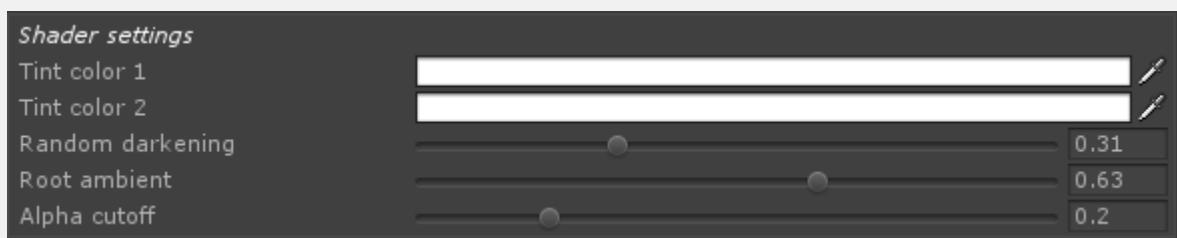


GRASS TEXTURE



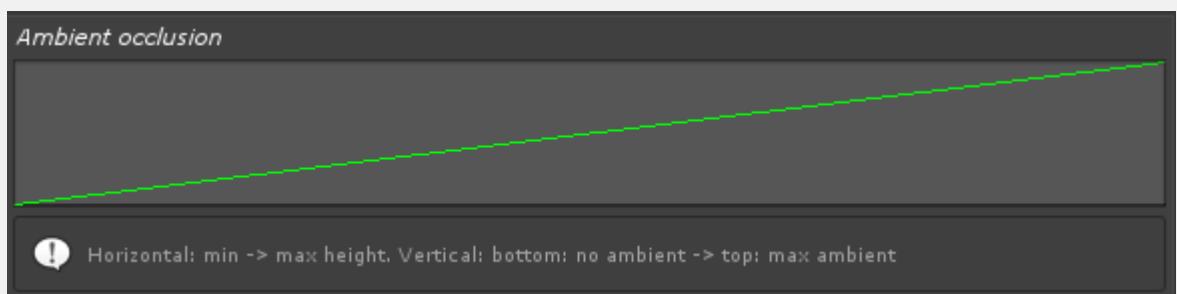
Here you assign the grass/plant texture you want to use with the grass patch.

SHADER SETTINGS

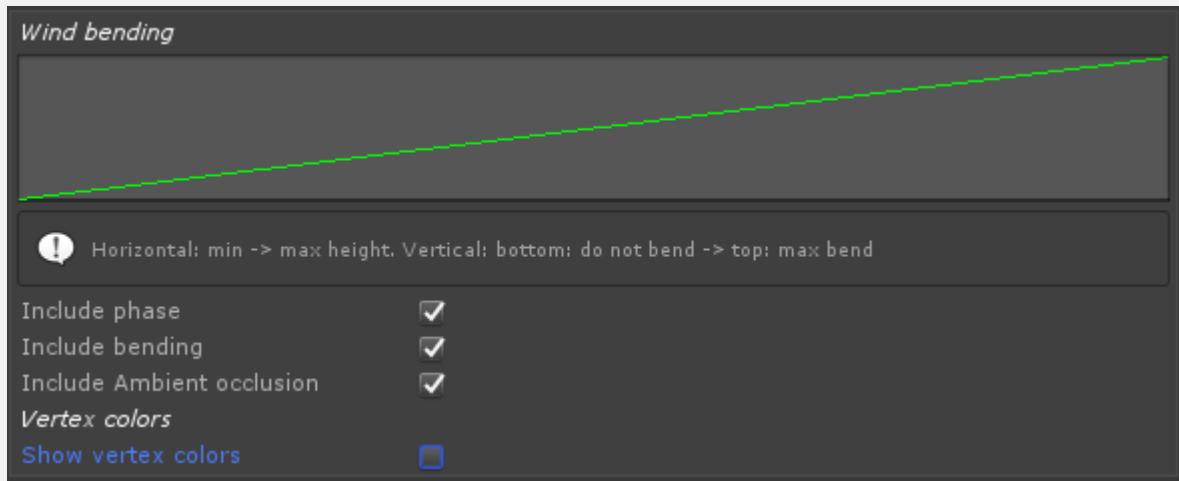


These shader settings are the initial settings for the grass patch. They can be adjusted in the VegetationSystem inspector when you add a grass patch to Vegetation Studio.

AMBIENT OCCLUSION

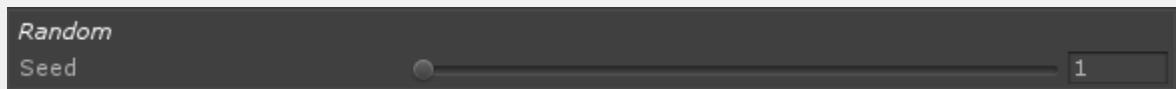


WIND BENDING



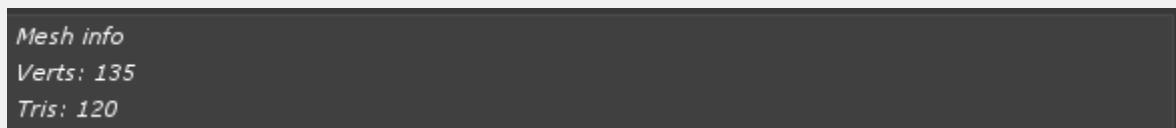
RANDOM

Sets the random seed used to generate all the positions, rotation and size of each plane.



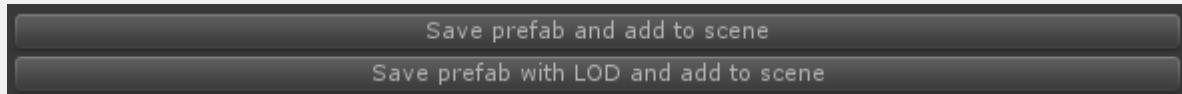
MESH INFO

Size info on the current generated mesh



GENERATE

When you are happy with the patch you can save it to the scene and project. If you select to save with LOD it will automatic generate a LOD for the grass patch. The LOD is additive and will keep adding more planes on the higher quality LODs up to the max plane count.



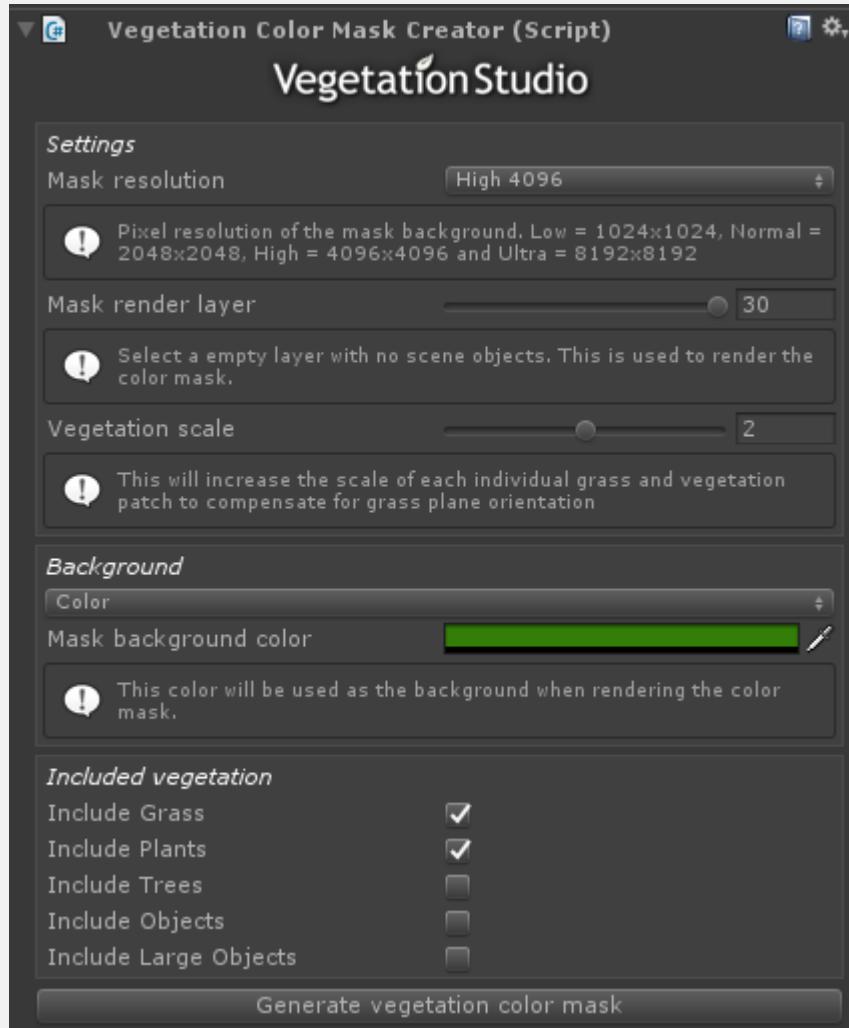
VEGETATION COLOR MASK CREATOR

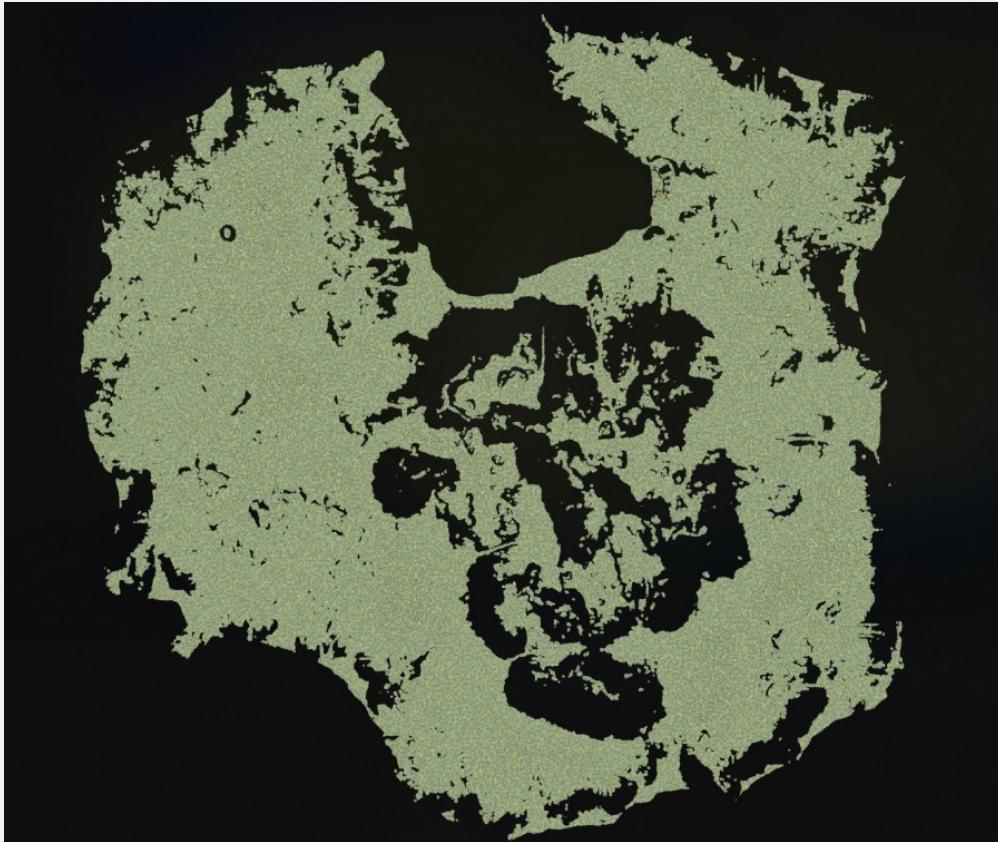
The VegetationColorMaskCreator component is a utility component designed to render the vegetation on a terrain to a ortho projected mask covering the entire terrain. The resulting image can be used to blend in vegetation coverage with color on terrains. This will be blended on to the terrain at the distance where normal grass stops giving a much better transition and visible grass distance.

Microsplat terrain shader has a built in module for the Vegetation Studio vegetation color mask. To see how to set it up follow this **guide**.

The component have to be added to the same GameObject as the VegetationSystem Component. It can be removed after generation.

Vegetation Studio will automatic configure fade in distance for the mask based on the current vegetation distance.





MASK RESOLUTION

Here you set the output resolution for the created mask. Available resolutions are 1024 to 8192. Rendering large masks will use a bit of memory while rendering.

MASK RENDER LAYER

The mask render layer must be set to an empty layer. This is need to make sure no other object in the scene will be rendered to the mask. Default layer is set to 30.

VEGETATION SCALE

Vegetation scale will increase the scale of individual vegetation items that is rendered to the mask. Since most grass patch meshes are designed with a better coverage from the side than the top the default setting is a 2x scale of each grass patch to give a better coverage. Experiment to see what looks best for you.

BACKGROUND

There are 3 settings for background. Setting a background is important to get good coloring on the partial transparent areas of the mask.

The options are.

- Color

Fixed background color. A color that fits the grass/plants are recommended.

- Image

Option to add a custom background image of the terrain. Some terrain shaders can produce this.

- Microsplat Terrain

When with microsplat installed you will get this option activated. It will use microsplat to export a background image with the current shader settings and use this in the generation.

INCLUDED VEGETATION

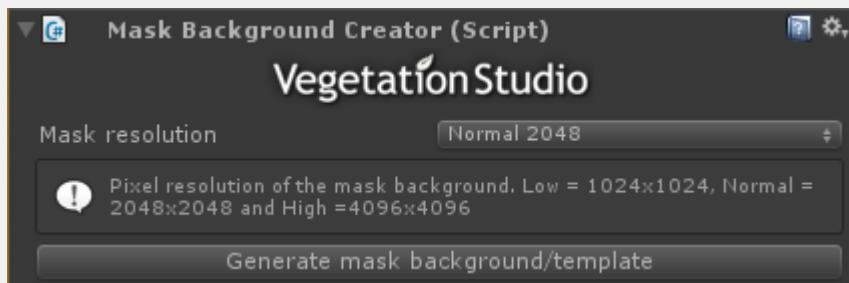


In the distance you can see the vegetation color mask blended with the terrain

MASK BACKGROUND CREATOR

The mask background creator component is a utility component design to help people paint manual masks. It will export color ortho image of the terrain as seen from the top. This image has a 1:1 scale of the terrain and can be used as a background in your painting application when you do manual mask edits. It helps you find where to paint.

Add the component to the GameObject with the VegetationSystem component. Select the mask resolution and press the “Generate mask background” button. Then select where in the project to save the image.



Example output of the demo scene terrain.



SCENE VEGETATION BAKER

The SceneVegetationBaker component is a utility component designed to allow users to bake the content of a persistent storage and run-time spawn rules to gameobjects in the scene. Add the component to the GameObject that has the VegetationSystem component. Select the VegetationItem you want to bake and press the button. There is an option to export this as static objects. Enabled by default.

The exported objects are all placed under a common root in the scene.

This could be used if you want some Vegetation Studio controlled objects to temporary be used for baking lighting. Then removing after.



Vegetation Studio



OBSTACLE MASK CREATOR

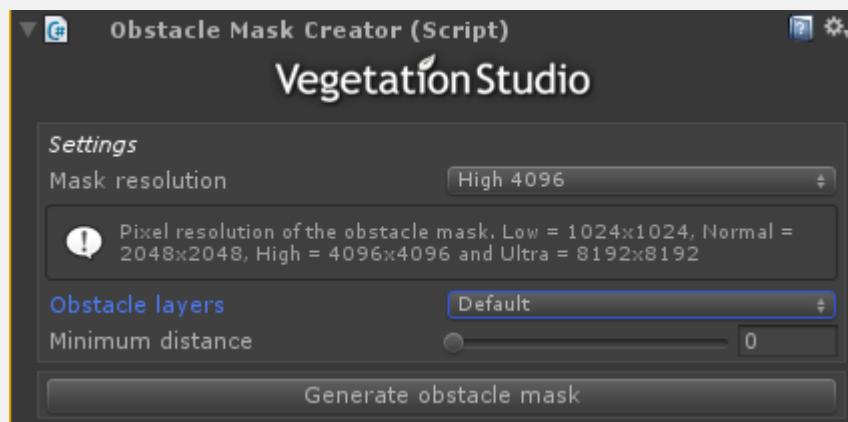
The obstacle mask creator component is a utility component designed to help creating exclusion masks for vegetation. To use the component add it to the GameObject with the VegetationSystem component.

Select the resolution you want on the output mask 1024 to 8196.

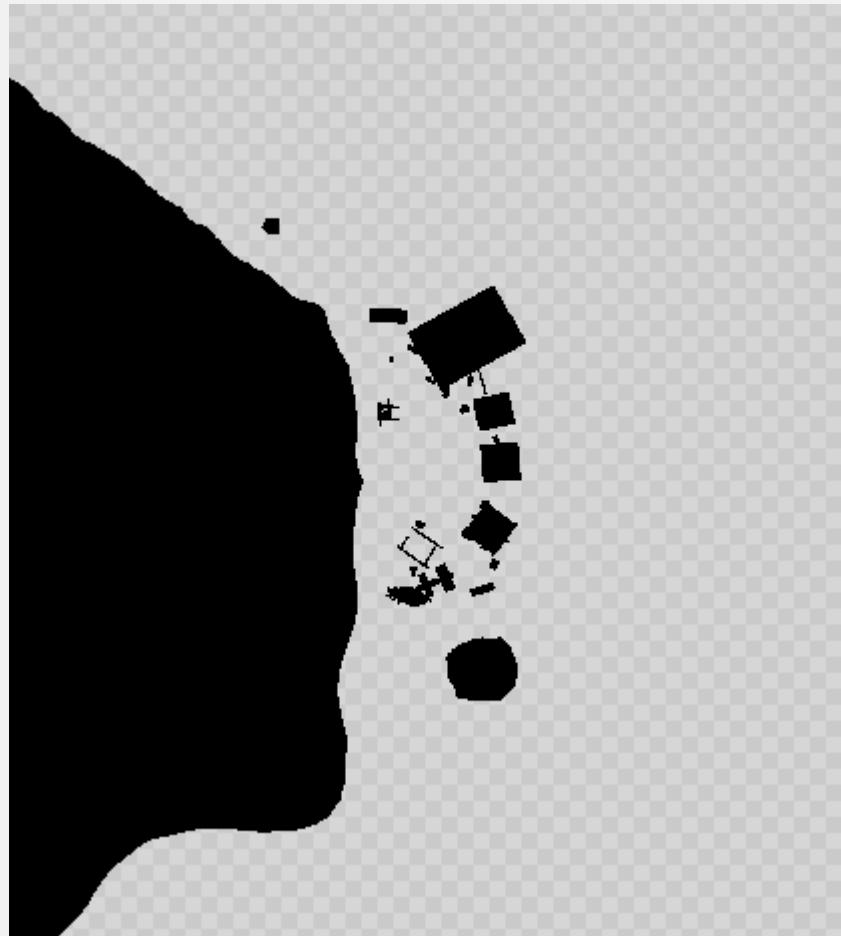
The component will spherecast against all colliders on the selected obstacle layers and add these to the mask. There is a minimum distance setting you can adjust. This will inflate all collider areas and make sure vegetation is not spawned closer than this.

It will generate a mask in 1:1 scale of the terrain assigned to the VegetationSystem component.

Press the “generate obstacle mask” button and choose a location in the project to save the mask. The generated mask can then be used as any other mask texture with the VegetationSystem component.



The example mask below is a cutout of the mask showing the village area in the demo scene.

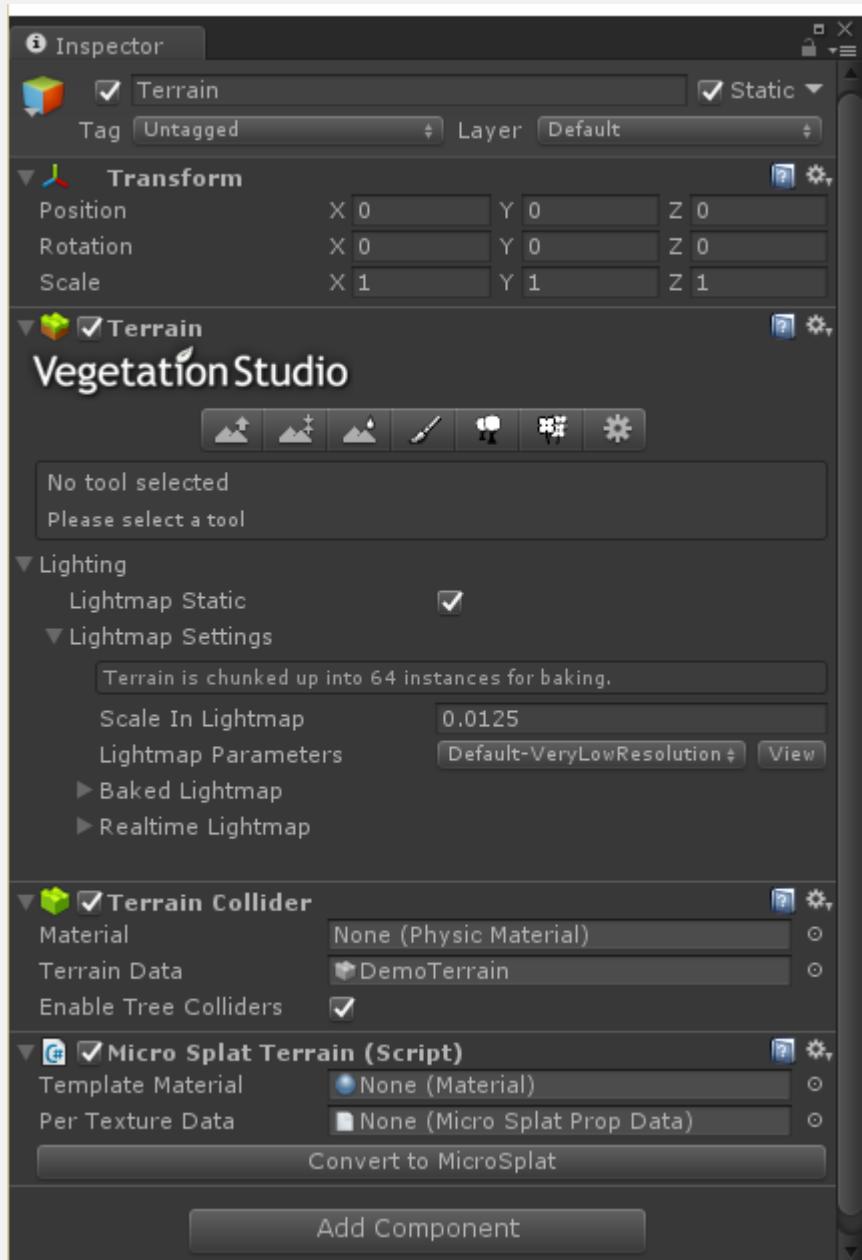


VEGETATION COLOR MASK SETUP WITH MICROSPLAT

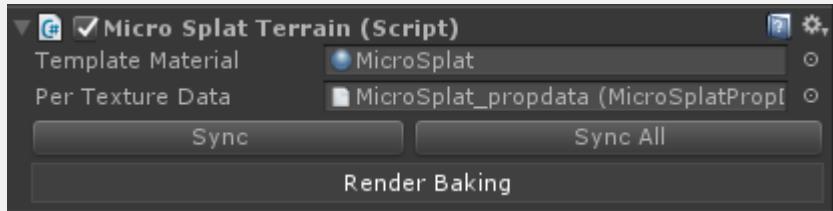
In this guide I will explain how to create and set up a vegetation color mask with the **Microsplat** terrain shader. Microsplat has a built in module for the vegetation color mask. As an example I will generate a mask from the included demo scene and set up Microsplat with the mask on the terrain. The module is included with the free Microsplat asset on the **asset store**. The mask is created using the **VegetationColorMaskCreator** component.

First we load the demo scene included in Vegetation studio. Find the terrain and add the Microsplat Terrain Component.

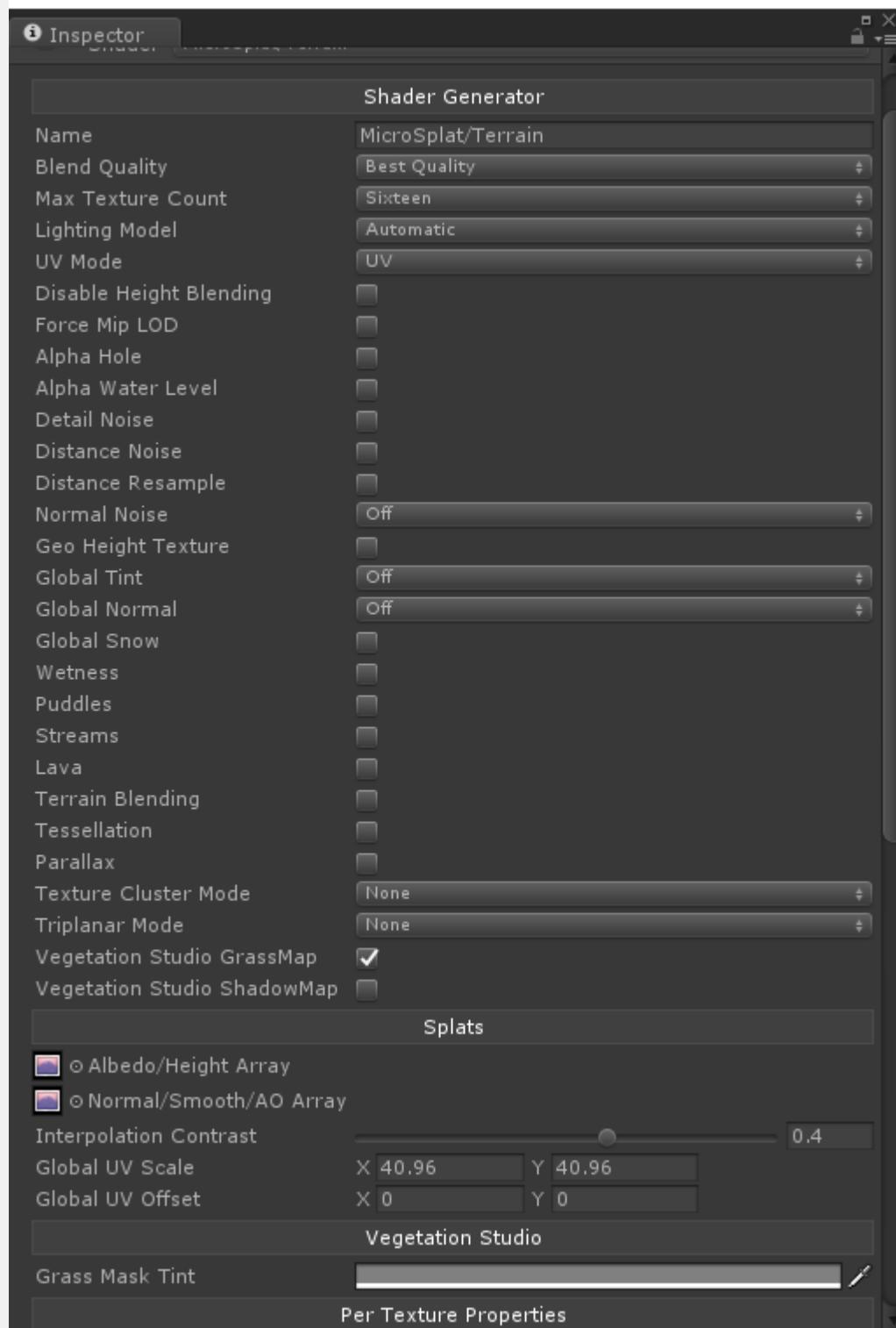
Vegetation Studio



With the component added we need to update the terrain and create all the microsplat materials and texture arrays. Press the Convert to Microsplat button. After some seconds you will get this.

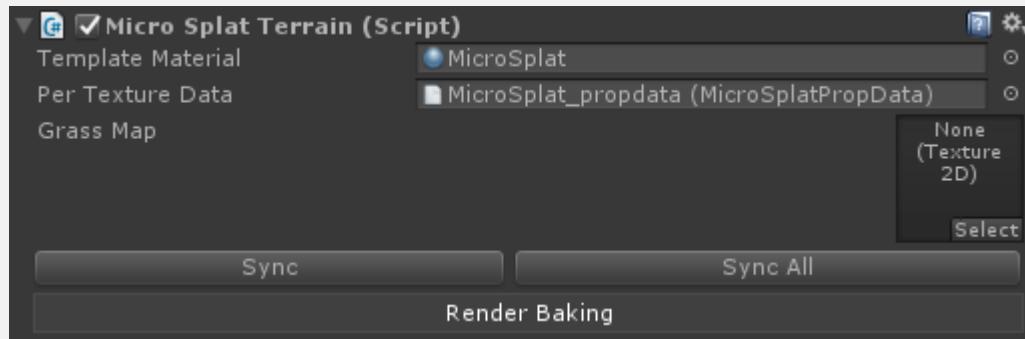


Microsplat is now set up on the terrain and is ready for use. Next step is to enable the Vegetation Studio grass module. Microsplat detects that Vegetation Studio is installed and enables 2 new modules. Click to enable the Vegetation Studio GrassMap



If we go back to the Microsplat Terrain component we will see that a GrassMap texture slot has

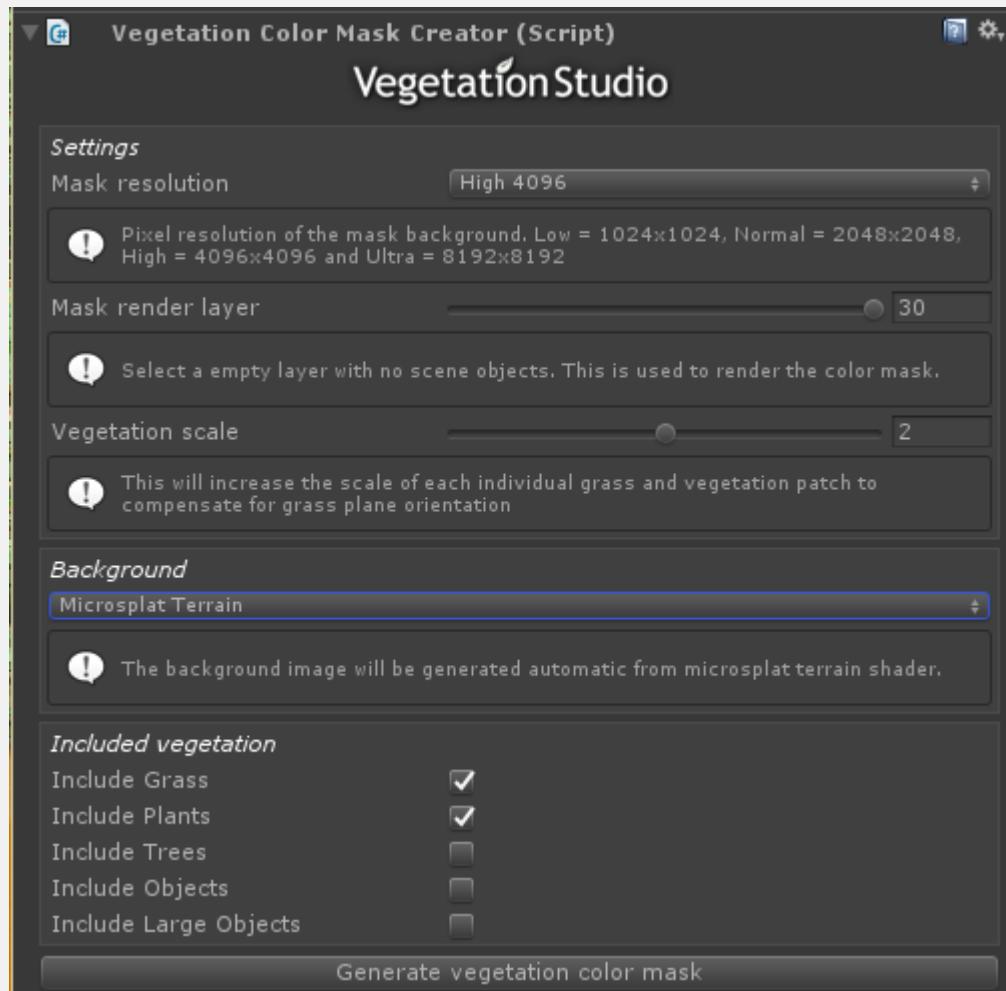
shown up. Now to generate the mask itself.



Find the VegetationSystem GameObject in the scene and add a VegetationColorMaskCreator component.

The only setting we need to change in this case is the background. Select microsplat terrain. The component will then use microsplat to generate the background used for the mask. This gives a better color match on the partial transparent edges of grass/plant areas as it has the correct color to use in the alpha blend.

Press the “Generate vegetation Color mask” button. choose a save location in the project and wait. This could take a minute to generate. There is a progress bar

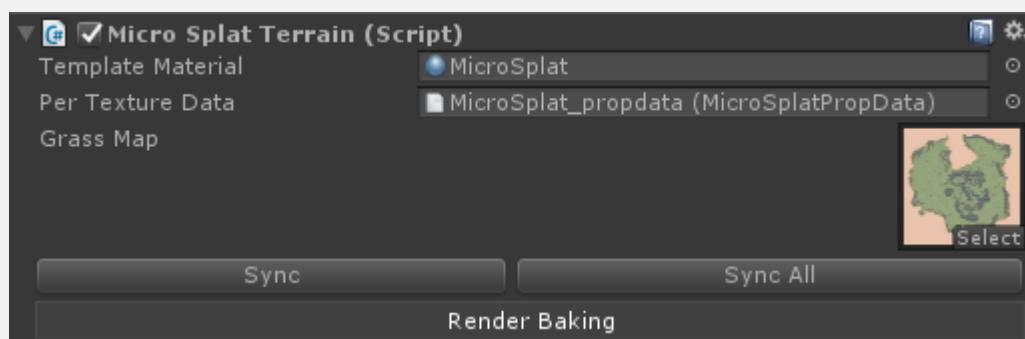


The resulting mask will look like this. It has an alpha for the non grass areas but this does not show in the unity preview window.

Vegetation Studio



Now assign the mask to the empty Grass Map slot on the MicroSplatTerrain component and it is working.



Vegetation Studio

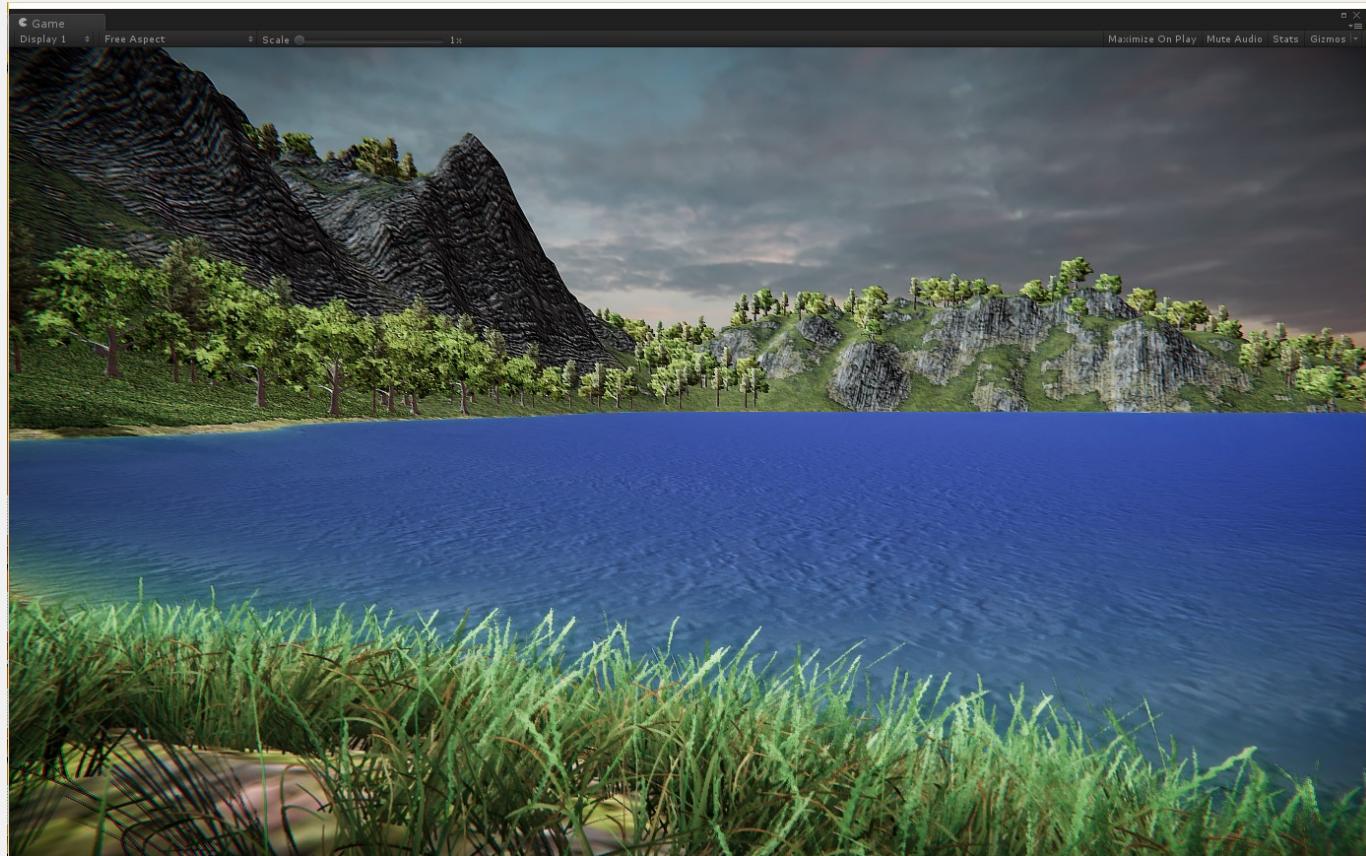
For fine tuning you can adjust the mask tint on the new property that is added on the Microsplat material



Here you see the result of the generated grass mask before and after.



Vegetation Studio



CUSTOM GRASS/PLANT MESHES FOR VEGETATION STUDIO

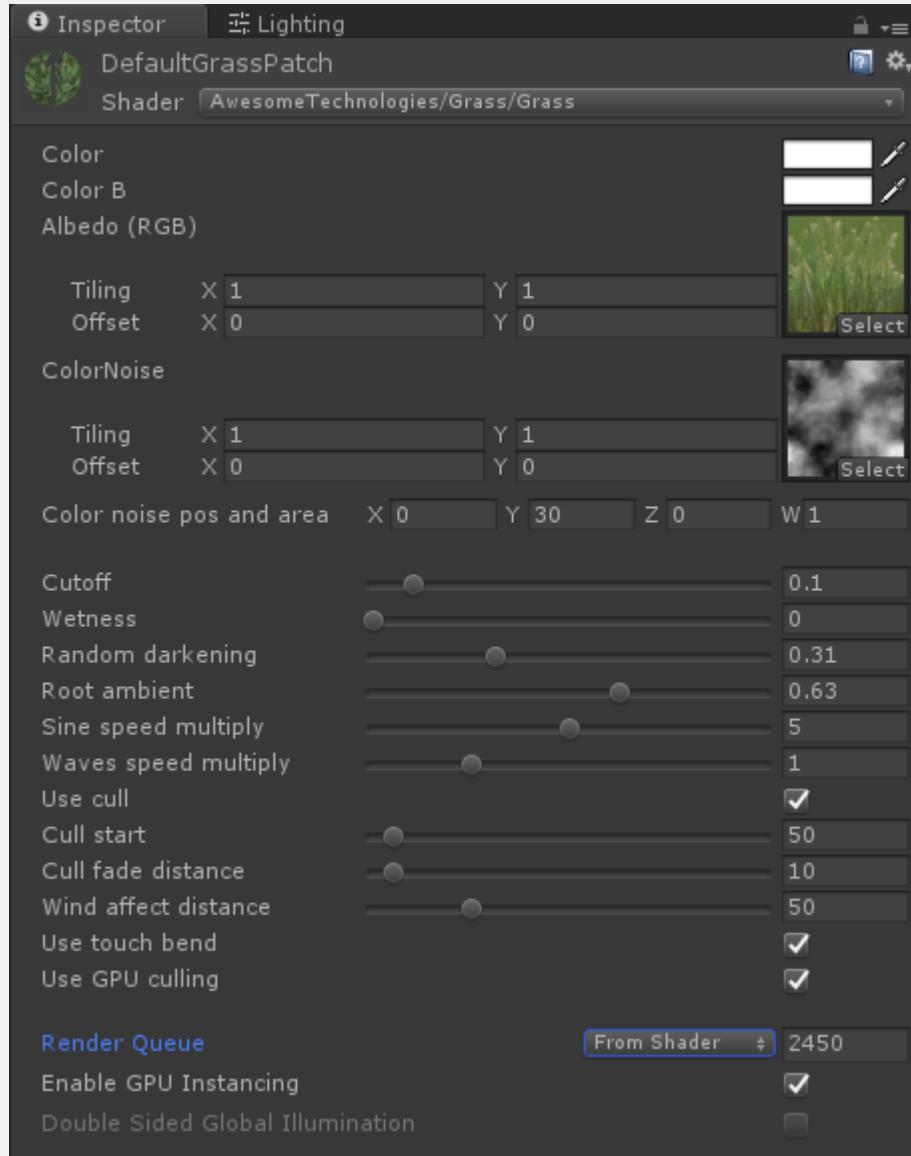
In this small guide I will explain how to set up your custom grass and plant meshes with the Vegetation System grass Shader. This will allow you render your grass meshes using the instanced indirect implementation in Vegetation Studio and also use the TouchBend features.

You can also convert existing shaders to be used with instanced indirect, but that is another guide.

MATERIAL

In order to get started create a new material or change shader on your existing. The shader to select is AwesomeTechnologies/grass/grass. This shader can be used for plants and flowers also.

Vegetation Studio



- Assign the your albedo color texture to the material.
- Assign a noise texture to the ColorNoise texture field. There is an included noise texture called PerlinSeamless in the VegetationStudio package. This noise is used to generate a color distribution on the terrain lerping from Color to Color B tint.
- You can leave the rest of the settings to default as Vegetation Studio will configure most of them for you automatic.
- The settings below are exposed to be configured in the UI of vegetation studio, but you can still set them on the material if you want a different default setting.



WIND AND ROOT DARKENING

In order to get wind and root darkening on your vegetation meshes we need to have a bit of info in the mesh. This is done using the vertex colors on the mesh itself. This can be added in any mesh modeling tool that supports vertex painting or in code if you are generating the mesh

VERTEXCOLORS

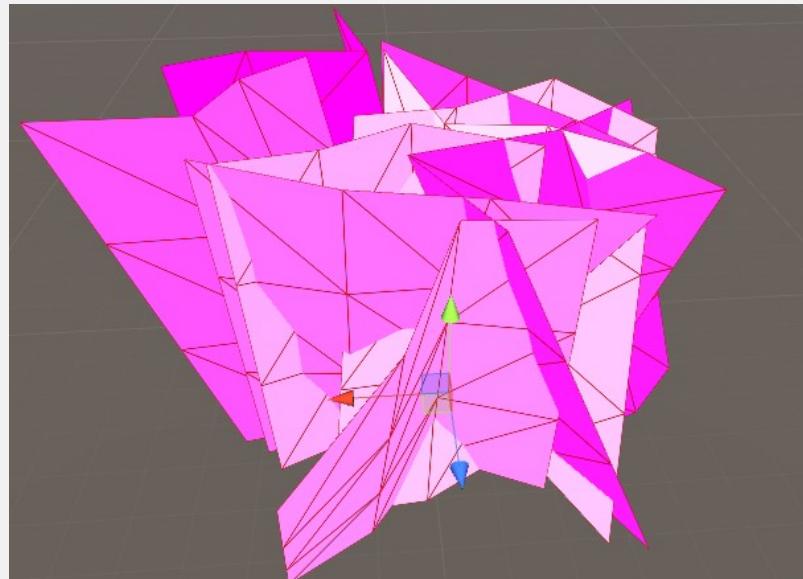
There is different info coded in each channel of the mesh.

R CHANNEL/ROOT DARKENING

In the Red channel you can add a root darkening amount. This is used to create a vertex based ambient occlusion to the root of the grass. Value goes from 0-1. This is multiplied with the root ambient value on the shader for the final effect.

G CHANNEL/PHASE

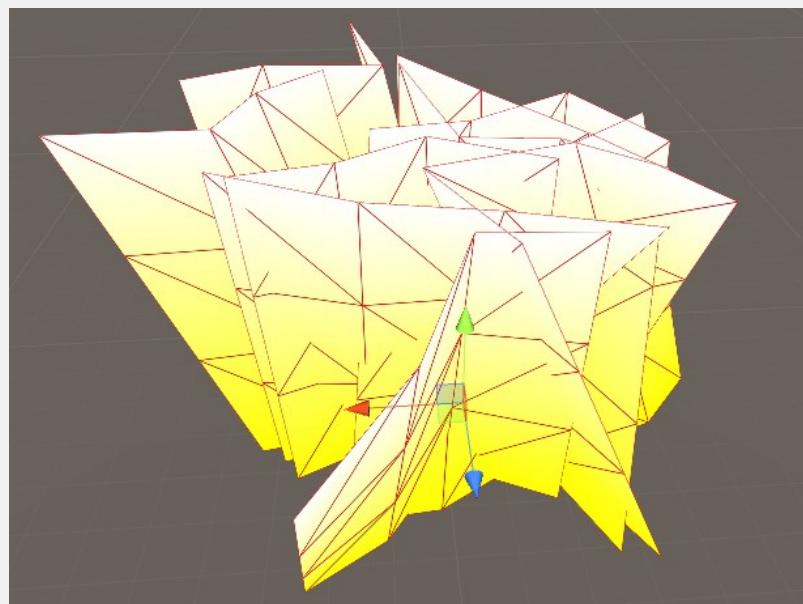
In order to get better looking wind it is nice to have a phase value painted on the mesh. In the example image below each plane in the mesh has a different value (0-1) in the Green channel. This will make the individual grass strands move in different phase with the sinus based wind. It looks much better if all of the mesh does not move together.



The shown colors of the example image does not reflect the channel color you add info to

B CHANNEL/BEND AMOUNT

The B channel of the vertex colors are the bend amount. for grass this is usually based on a curve or linear from bottom to top with the root vertices set to 0. This gives no movement. A higher value would make the root move on the ground and not look good.



The shown colors of the example image does not reflect the channel color you
add info to

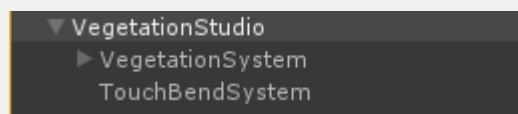
Add the finished prefab to Vegetation Studio as a Grass or Plant prefab and play. □

VEGETATION STUDIO WITH MAP MAGIC

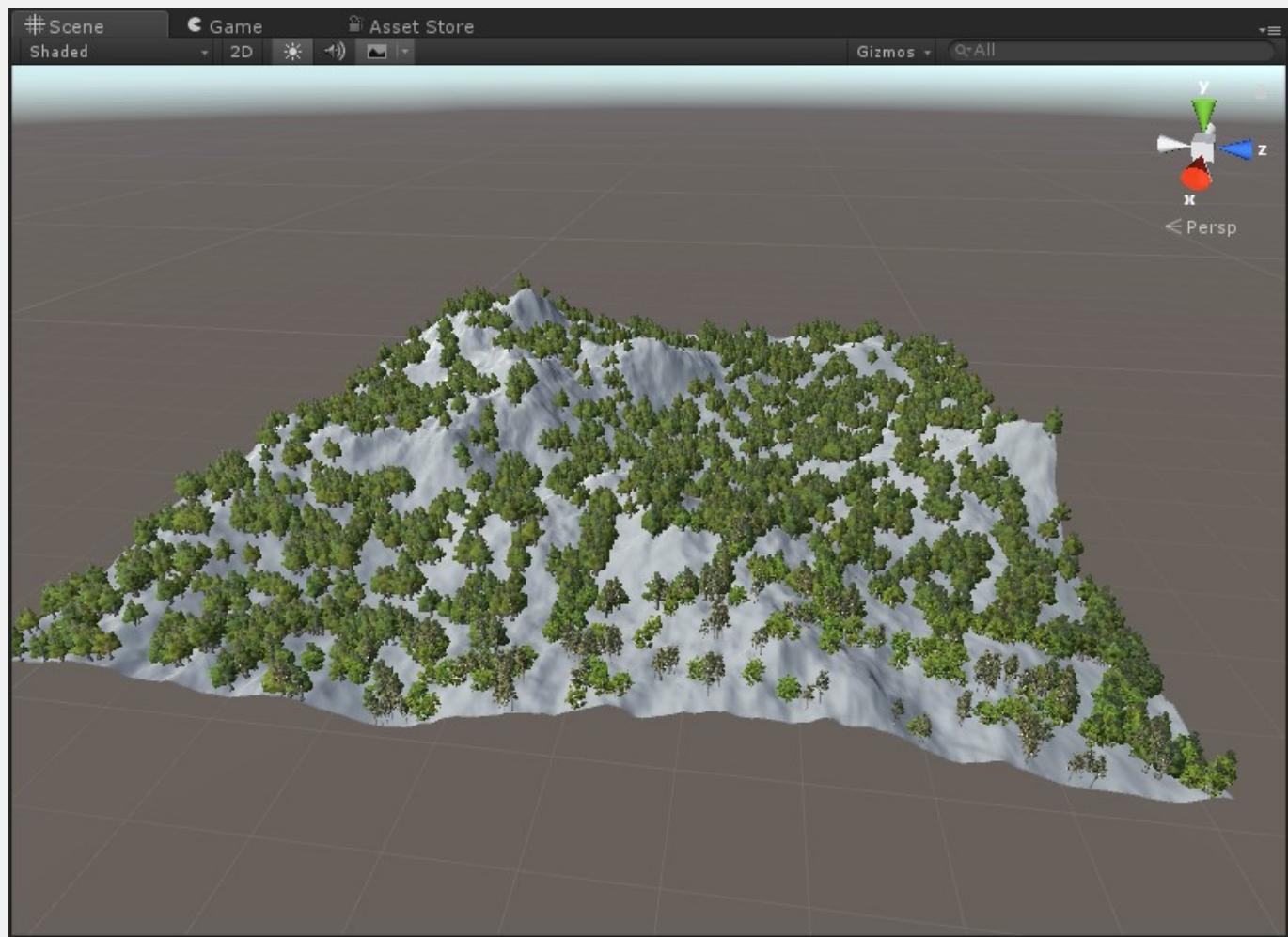
This is a guide for how to set up Vegetation Studio with the Infinite terrain system from Map Magic. It will show you how to create vegetation on the run-time created terrains.

Included with Vegetation Studio is a component called **MapMagicInfiniteTerrain** that will connect to Map Magic and get events when a new terrain is generated.

To get started make sure you have a Vegetation Studio GameObject in your scene.

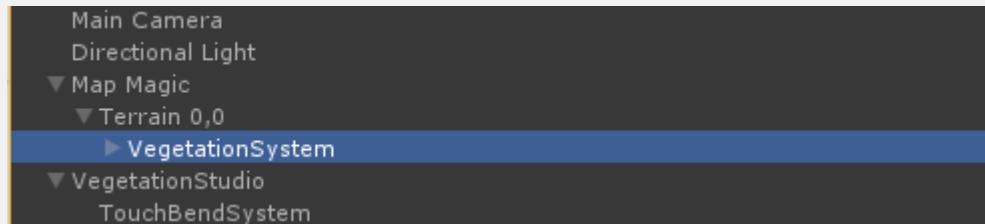


In my test scene it is set up with these trees for the main terrain.



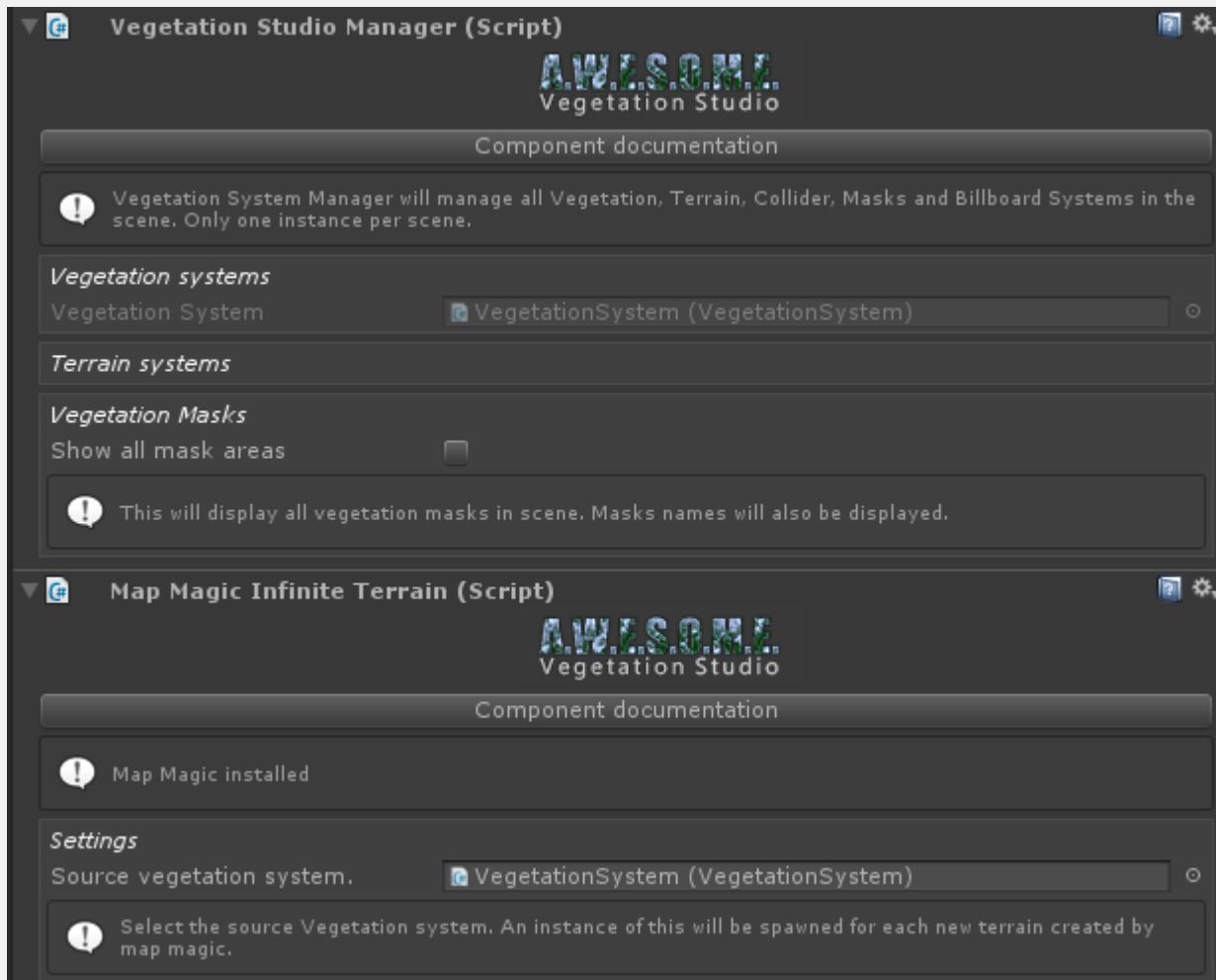


Make sure "auto select terrain" is turned off and your main terrain in the scene is selected.



Move the VegetationSystem component as a child object of your main terrain. This is to make sure vegetation on that terrain is disabled if your main terrain gets out of range during playmode. Only do this when you are finished making your Map Magic terrain. If you rebuild it with the VegetationSystem component as a child it will be destroyed.

Vegetation Studio

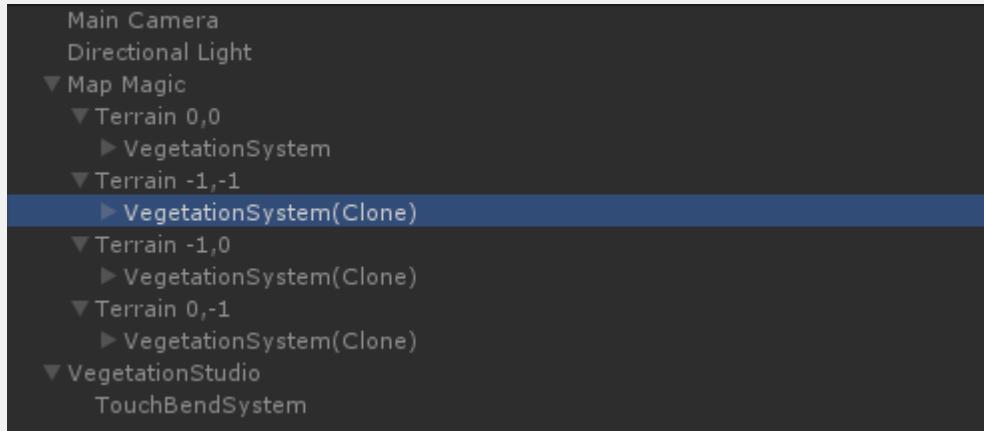


Add a MapMagicInfiniteTerrain component to the VegetationStudio root GameObject.

Drag and drop the VegetationSystem component you moved as a child to the center terrain. This will be the source component for vegetation. All new terrains will get a clone of this configured with the correct terrain.

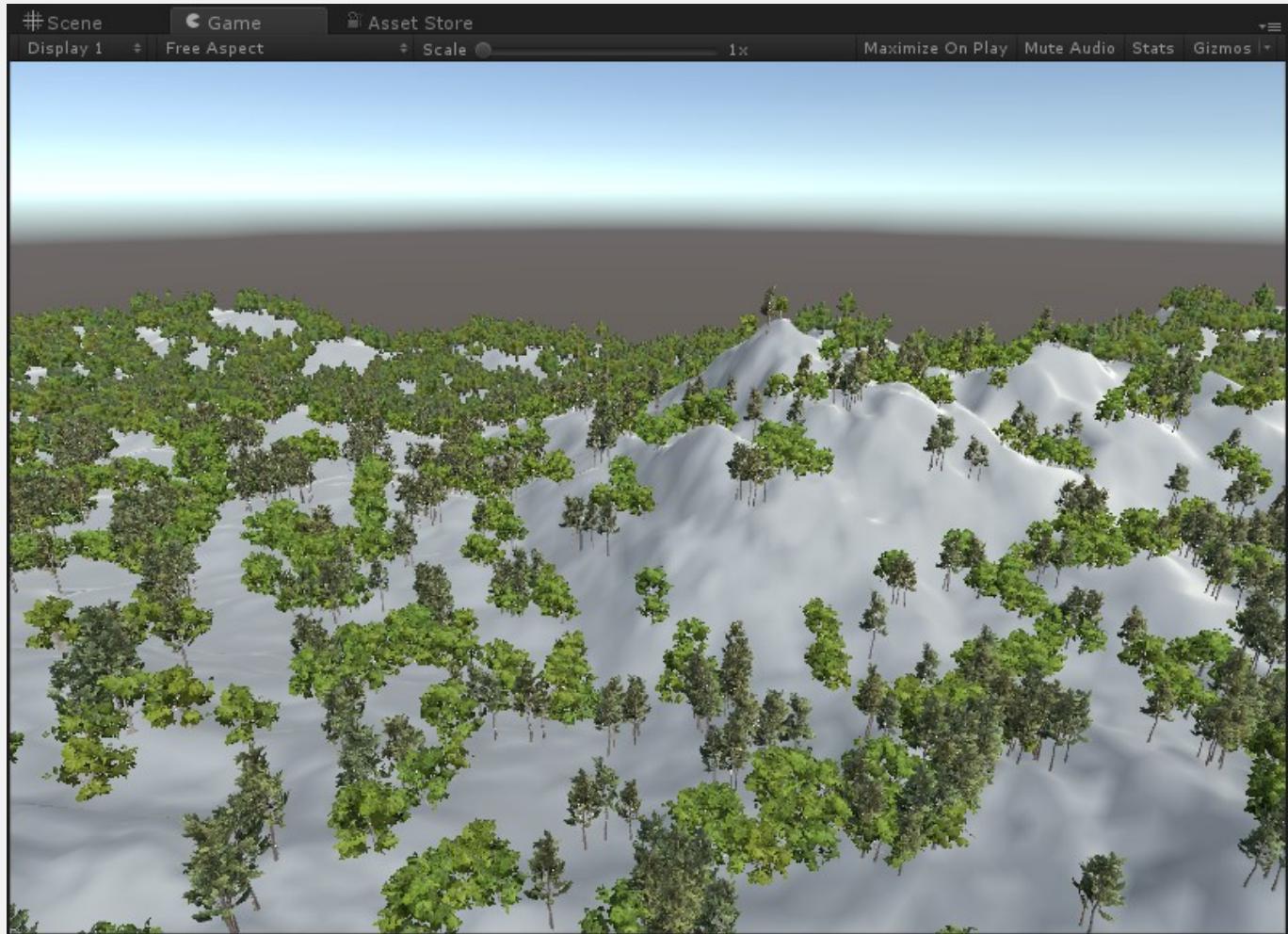
All needed settings is now done.

Try to enter playmode.



Depending on your Map Magic settings it will now spawn new terrains and a clone of the VegetationSystem component is added to the new terrains. As it is a child object of the terrain it will be enabled/disabled/destroyed with the terrain. All controlled by Map Magic.

Vegetation Studio



This last image shows the 4 terrains loaded with the same vegetation.

Good luck testing.