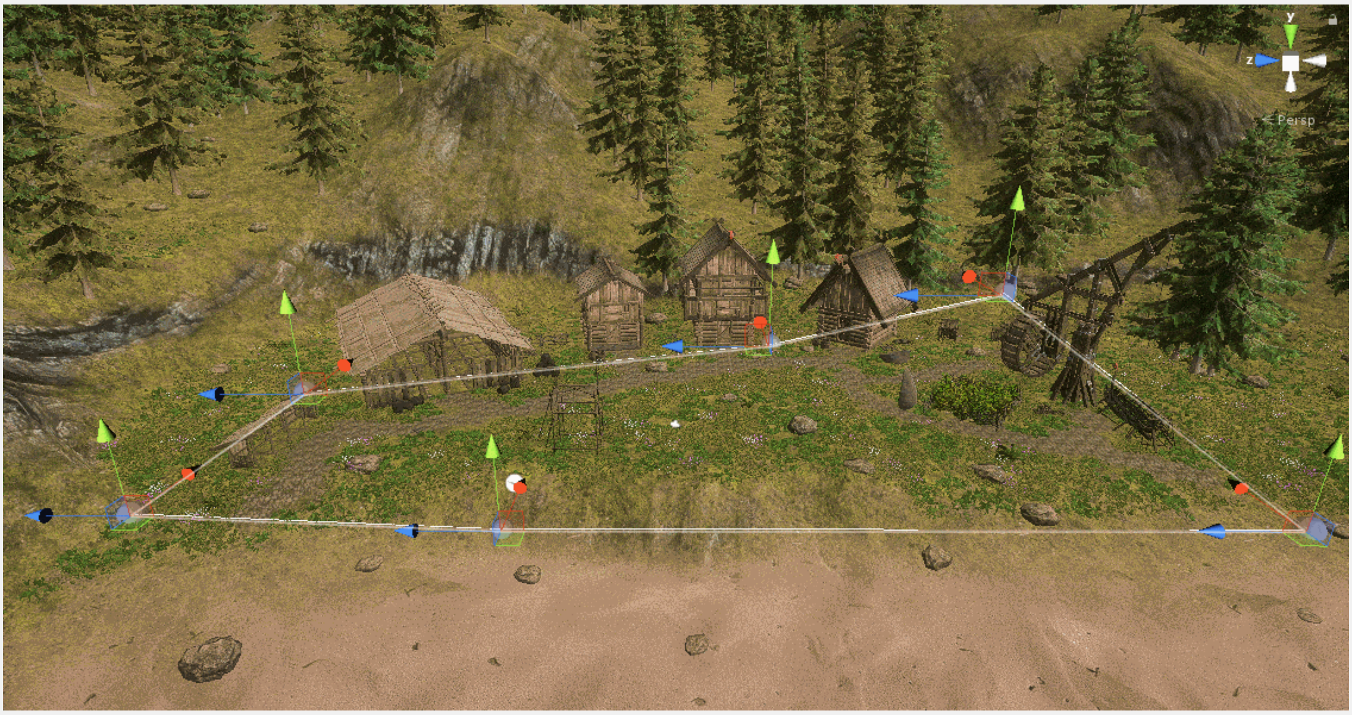
VEGETATION MASK AREA

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.

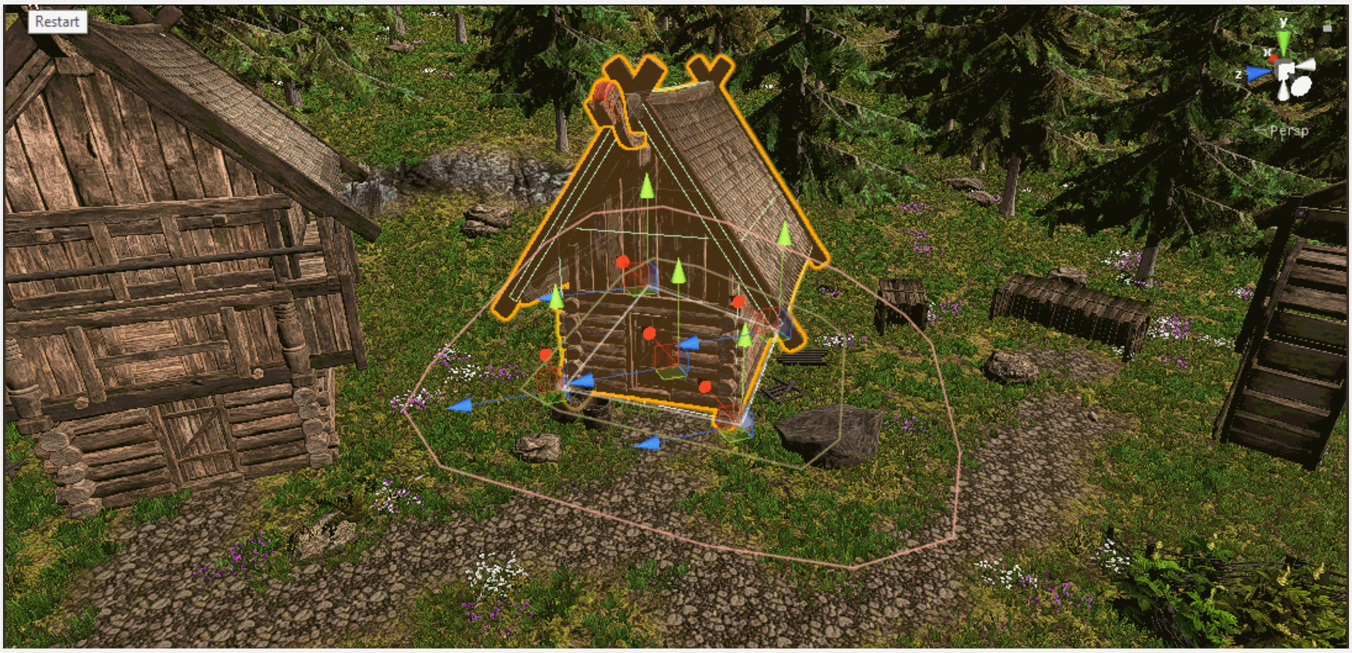
In addition to the area defined by the nodes in the component, each vegetation type, Grass, Plants, Trees, Objects and Large Objects can be filtered and can have an additional range outside of the polygon mask area.

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.

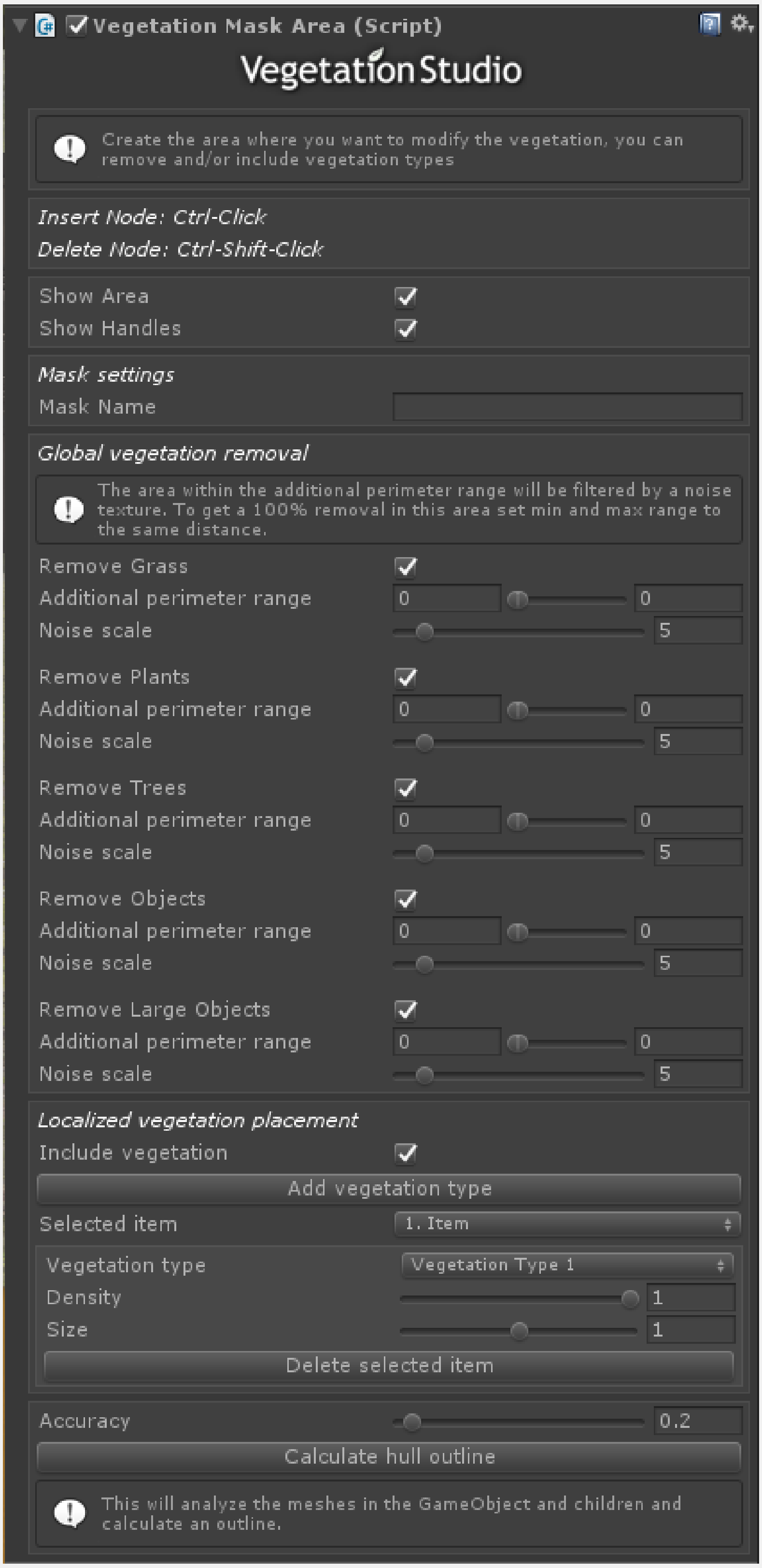
Node editing
Handles
Mask settings
Global vegetation removal
Localized vegetation placement
Script access



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 Mask Area (Script)

Vegetation Studio

Create the area where you want to modify the vegetation, you can remove and/or include vegetation types

Insert Node: Ctrl-Click

Delete Node: Ctrl-Shift-Click

Show Area ☑

Show Handles ☑

Mask settings

Mask Name

Global vegetation removal

The area within the additional perimeter range will be filtered by a noise texture. To get a 100% removal in this area set min and max range to the same distance.

Remove Grass ☑
Additional perimeter range 0 0
Noise scale 5

Remove Plants ☑
Additional perimeter range 0 0
Noise scale 5

Remove Trees ☑
Additional perimeter range 0 0
Noise scale 5

Remove Objects ☑
Additional perimeter range 0 0
Noise scale 5

Remove Large Objects ☑
Additional perimeter range 0 0
Noise scale 5

Localized vegetation placement

Include vegetation ☑

Selected item 1. Item

Vegetation type Vegetation Type 1
Density 1
Size 1

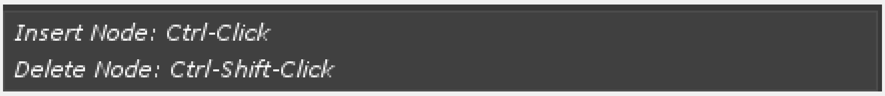
Accuracy 0.2

This will analyze the meshes in the GameObject and children and calculate an outline.

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.

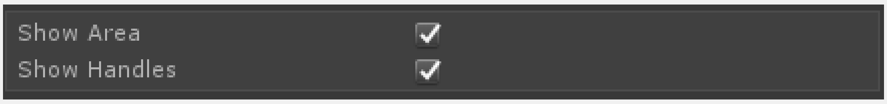


Insert Node: Ctrl-Click
Delete Node: Ctrl-Shift-Click

HANDLES

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

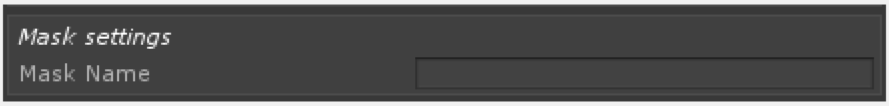
Show handles will add movement handles to the scene view. Use them to move nodes. With high node count polygons (100+) handles at a distance will not show.



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.

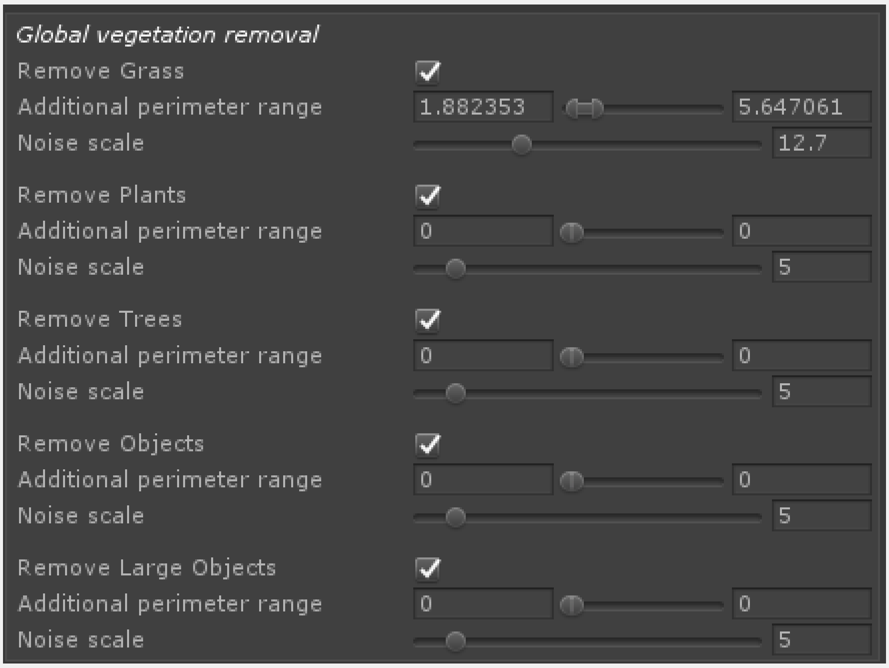


Mask settings
Mask Name

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.



Global vegetation removal
Remove Grass ☑
Additional perimeter range 1.882353 5.647061
Noise scale 12.7

Remove Plants ☑
Additional perimeter range 0 0
Noise scale 5

Remove Trees ☑
Additional perimeter range 0 0
Noise scale 5

Remove Objects ☑
Additional perimeter range 0 0
Noise scale 5

Remove Large Objects ☑
Additional perimeter range 0 0
Noise scale 5

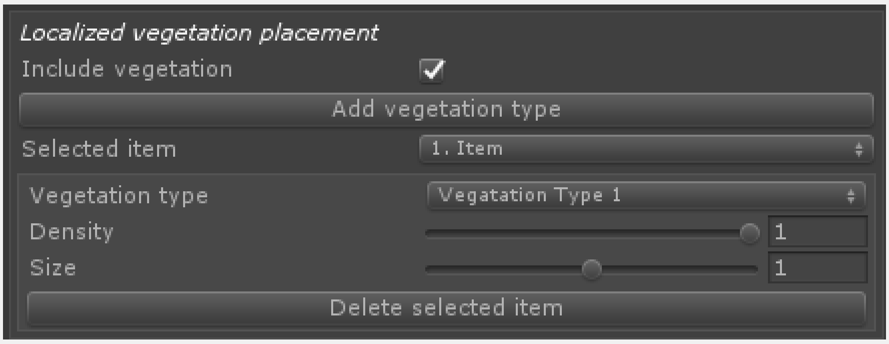
The additional perimeter range has a min/max value. The distance between these is using a perlin noise for the falloff to give a more organic edge to the mask. You can adjust the Noice scale to get a result you like

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.



Localized vegetation placement

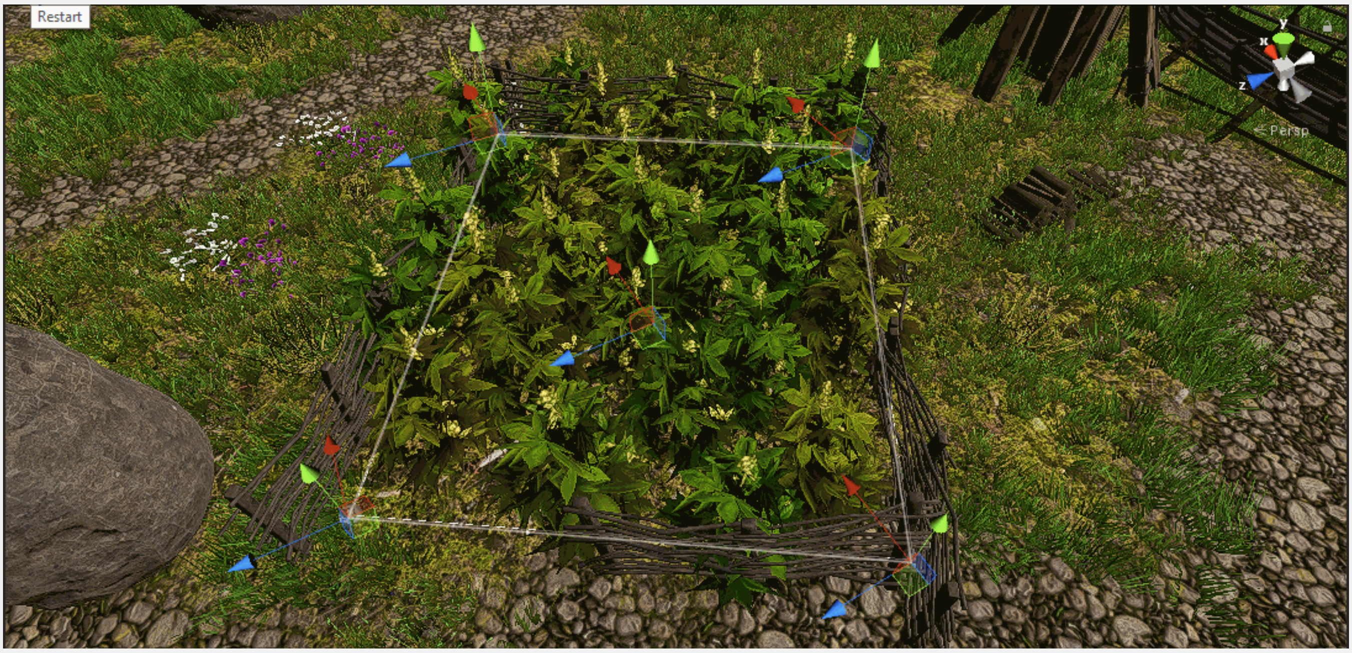
Include vegetation ☑

Selected item: 1. Item

Vegetation type: Vegatation Type 1

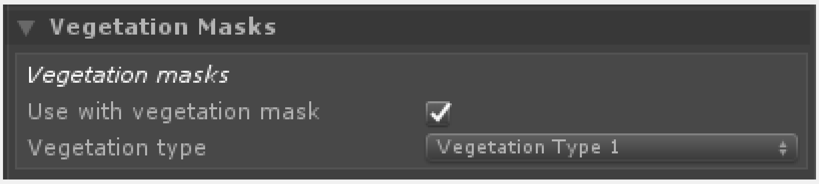
Density: 1

Size: 1



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.



Vegetation Masks

Vegetation masks

Use with vegetation mask ☑

Vegetation type: Vegetation Type 1

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:
[enlighter lang=”csharp”]

```
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;
```

[/enlighter]
Adding a new mask to a gameobject:

[enlighter lang=”csharp”]

```
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();
```

[/enlighter]
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);

[enlighter lang=”csharp”]

```
List<Vector3> worldSpaceNodeList = GetWorldSpaceNodePositions();
```

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

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