

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.

Vegetation Mask Line (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	<input checked="" type="checkbox"/>
Additional perimeter range	<input type="text" value="0"/> <input type="text" value="0"/>
Noise scale	<input type="text" value="5"/>
Remove Plants	<input checked="" type="checkbox"/>
Additional perimeter range	<input type="text" value="0"/> <input type="text" value="0"/>
Noise scale	<input type="text" value="5"/>
Remove Trees	<input checked="" type="checkbox"/>
Additional perimeter range	<input type="text" value="0"/> <input type="text" value="0"/>
Noise scale	<input type="text" value="5"/>
Remove Objects	<input checked="" type="checkbox"/>
Additional perimeter range	<input type="text" value="0"/> <input type="text" value="0"/>
Noise scale	<input type="text" value="5"/>
Remove Large Objects	<input checked="" type="checkbox"/>
Additional perimeter range	<input type="text" value="0"/> <input type="text" value="0"/>
Noise scale	<input type="text" value="5"/>

Localized vegetation placement

Include vegetation

Width

! Set the width of the line segments in meters to define the mask area



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.

Show Area
Show Handles

MASK SETTINGS

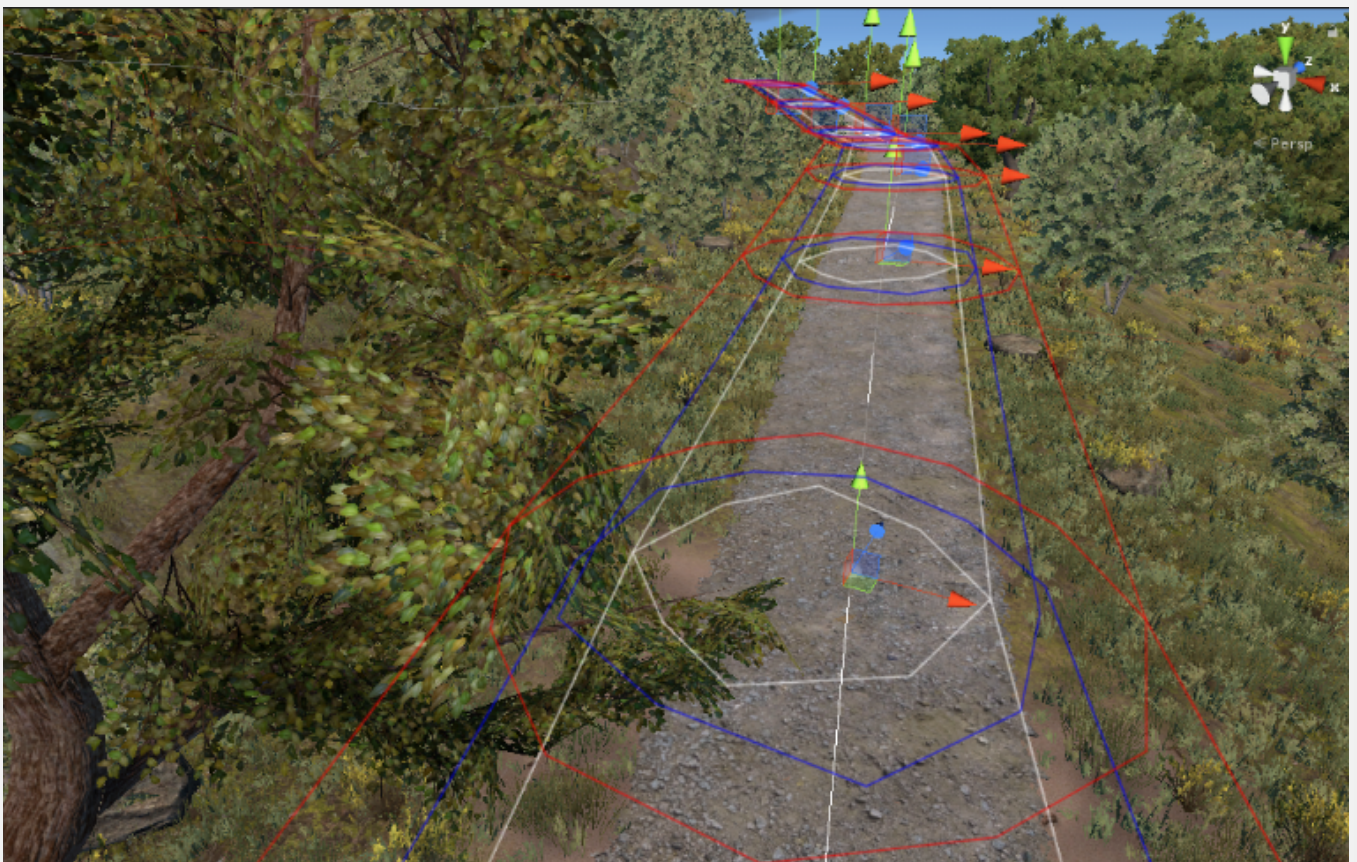
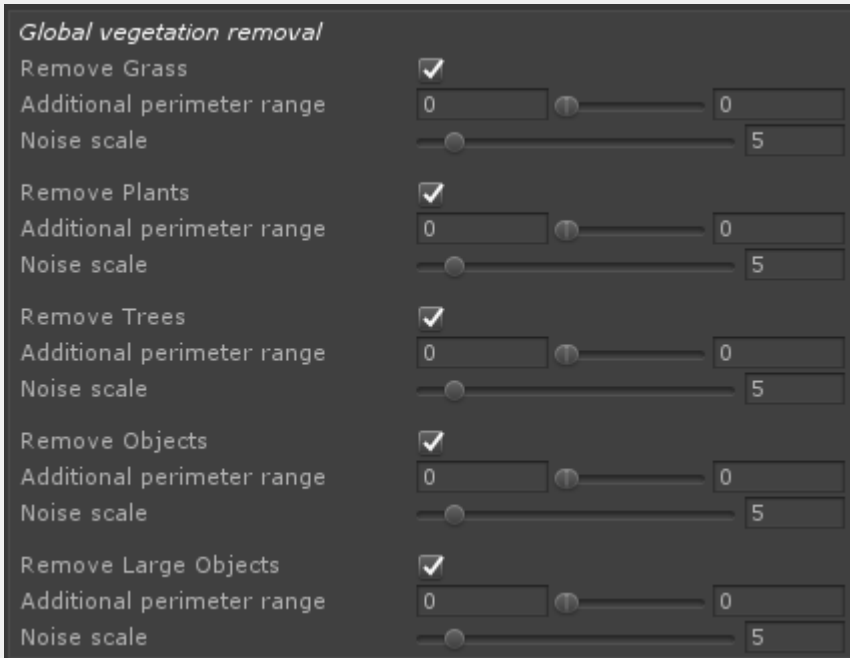
Mask name will show up as a label in the center polygon in sceneview when option is turned on in VegetationStudioManager component.

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



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**)

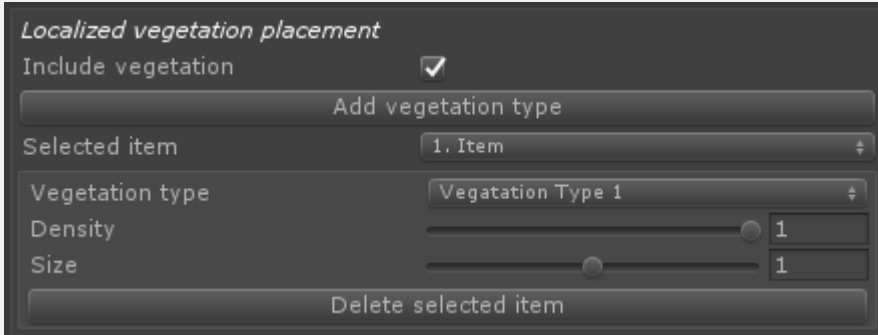
The additional perimeter range has a min/max value. The distance between these is using a perlin noise for the falloff to give a more organic edge to the mask. You can adjust the Noise scale to get a result you like

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.

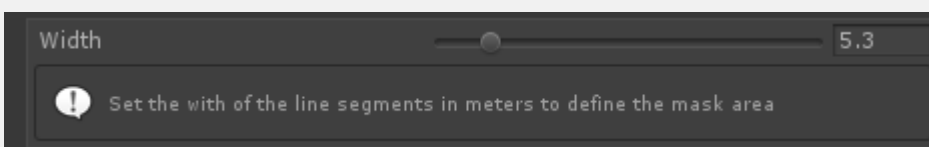
The vegetation item will be spawned inside the polygon with all normal rules set in the Vegetation System inspector.

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:

```
[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"]
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();
[/enlighter]
```

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.


```
[enlighter lang="csharp"]
```

```
List<Vector3> worldSpaceNodeList = GetWorldSpaceNodePositions();  
//Replace GetWorldSpaceNodePositions with your own code to make a list of Vector3 positions in  
worldspace.
```

```
PolygonMaskLine maskLine = new PolygonMaskLine  
{  
    removeGrass = RemoveGrass,  
    removePlants = RemovePlants,  
    removeTrees = RemoveTrees,  
    removeObjects = RemoveObjects,  
    removeLargeObjects = RemoveLargeObjects,  
    additionalGrassWidth = AdditionalGrassPerimeter,  
    additionalPlantWidth = AdditionalPlantPerimeter,  
    additionalTreeWidth = AdditionalTreePerimeter,  
    additionalObjectWidth = AdditionalObjectPerimeter,  
    additionalLargeObjectWidth = AdditionalLargeObjectPerimeter  
};  
maskLine.AddPolygon(worldSpaceNodeList);  
VegetationStudioManager.AddVegetationMask(maskLine);
```

```
[/enlighter]
```

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