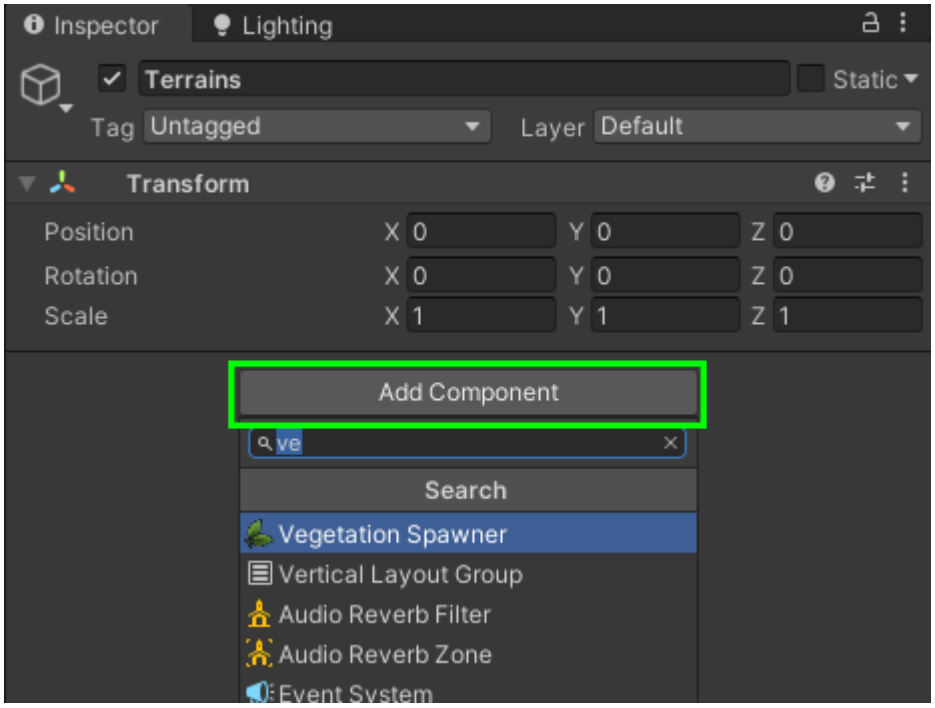
CONTENTS

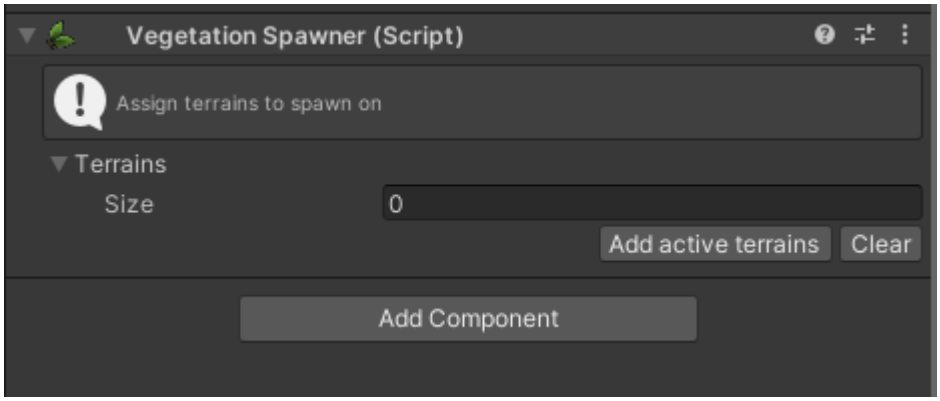
If you're familiar with the asset, please consider leaving a review!

FIRST STEPS

1: Create an empty GameObject and add the "Vegetation Spawner" component



2: Add your terrain objects to the list, or choose "Add active terrains" to add any enabled terrains.

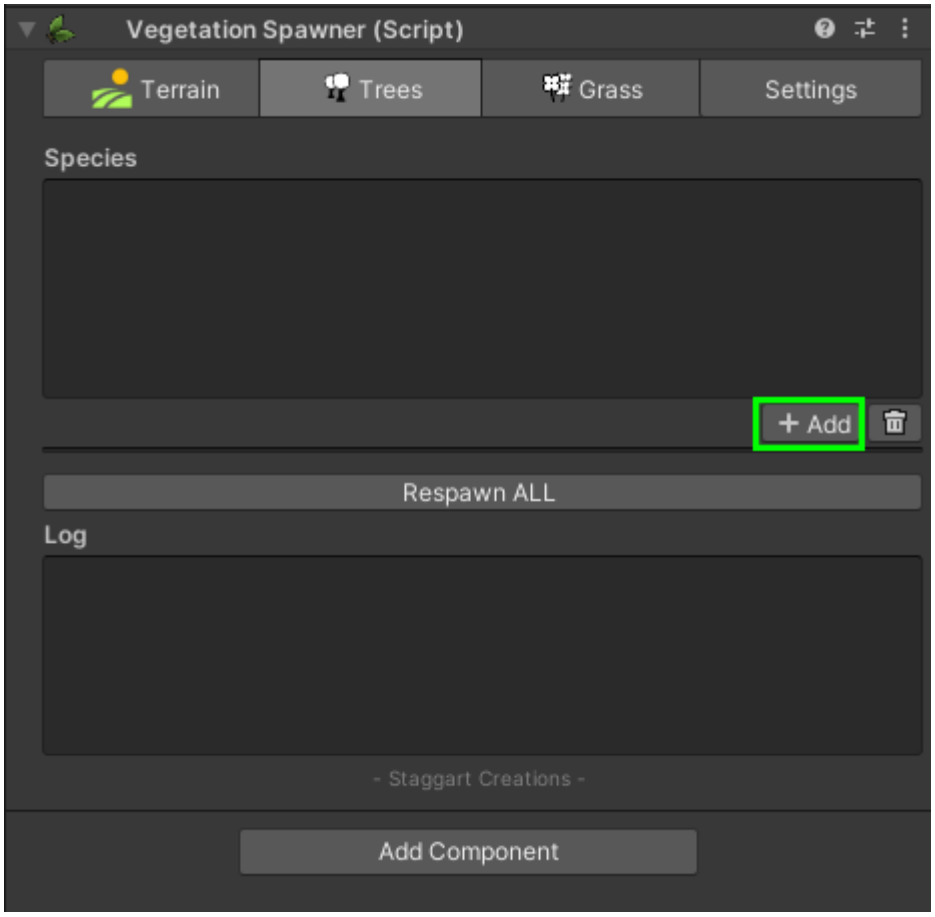


Note that the spawner should now be used to control vegetation. This means that any vegetation currently painted on the terrain is going to be cleared! This is also the case for any vegetation manually painted after running the vegetation spawner. There is unfortunately no layered vegetation system in Unity, that separates manually painted vegetation from procedural.

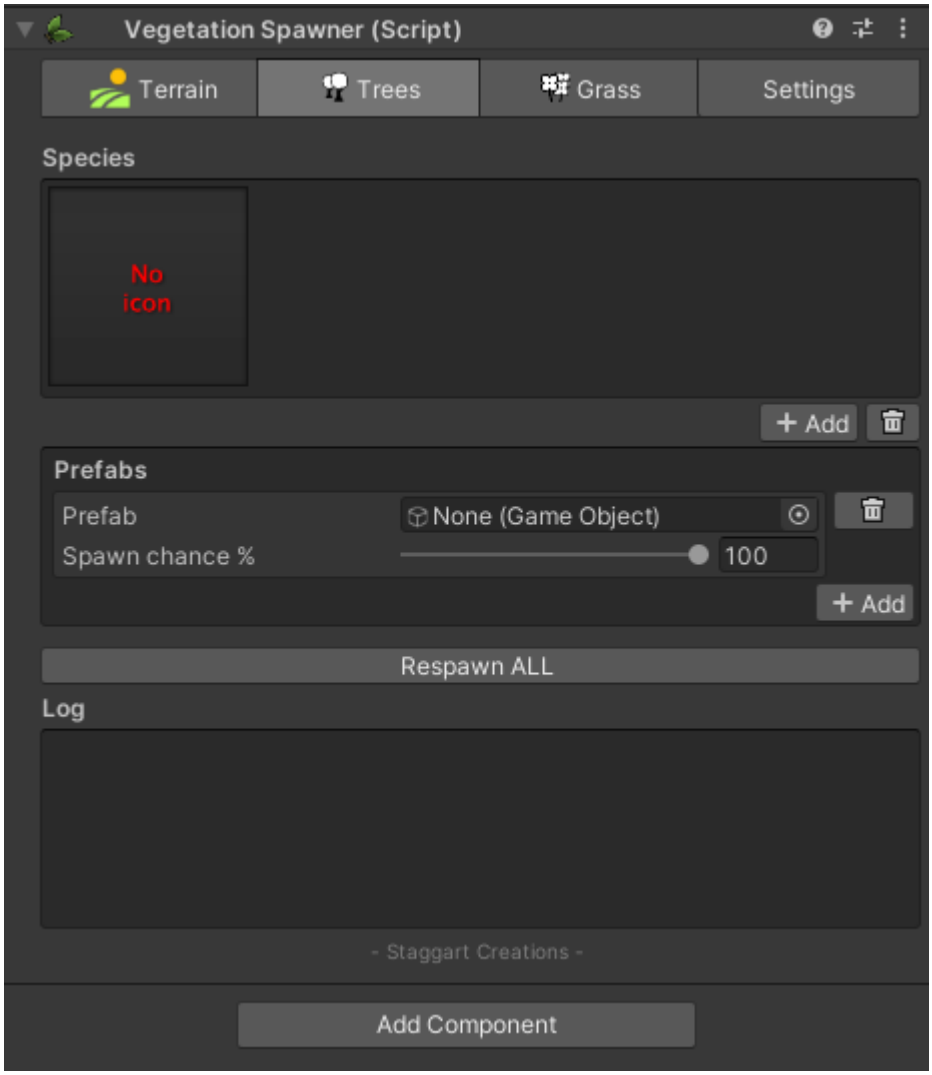
The component can safely be removed, without affecting the vegetation.

SPAWNING TREES

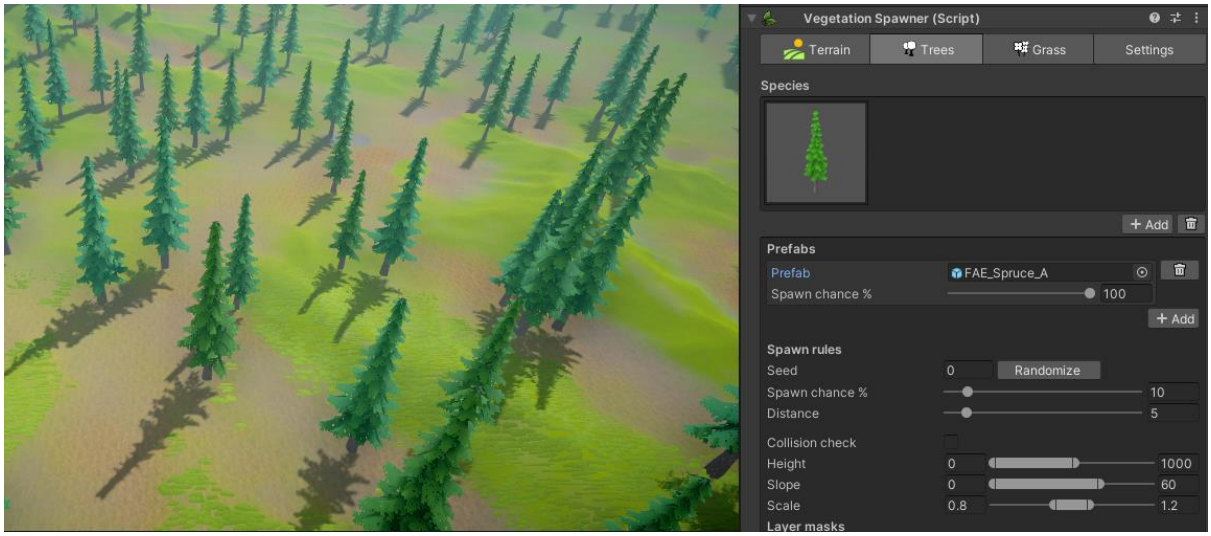
In the “Trees” tab you’ll be able to add tree prefabs to the spawning list. These are specified as species (eg. Spruce, birch, pine, oak, etc)



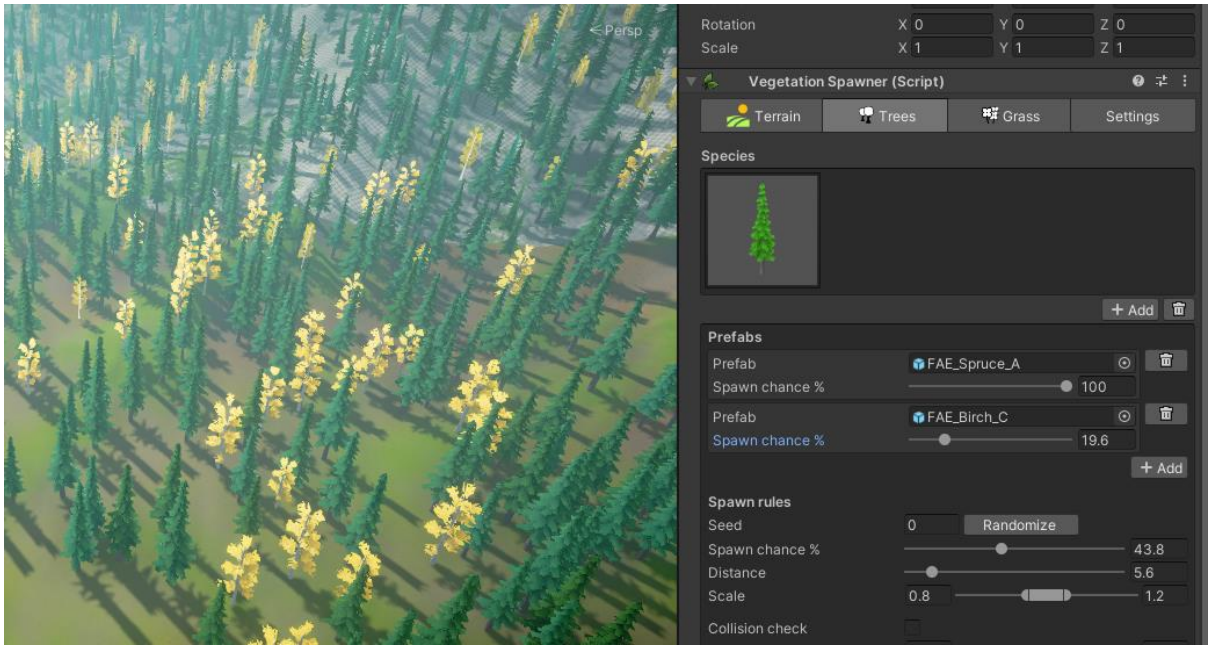
Click the “Add” button to add a first item, and assign a prefab. Always ensure “GPU Instancing” is enabled on the materials used by your tree, otherwise each individual tree requires a drawcall, which dramatically affects performance.



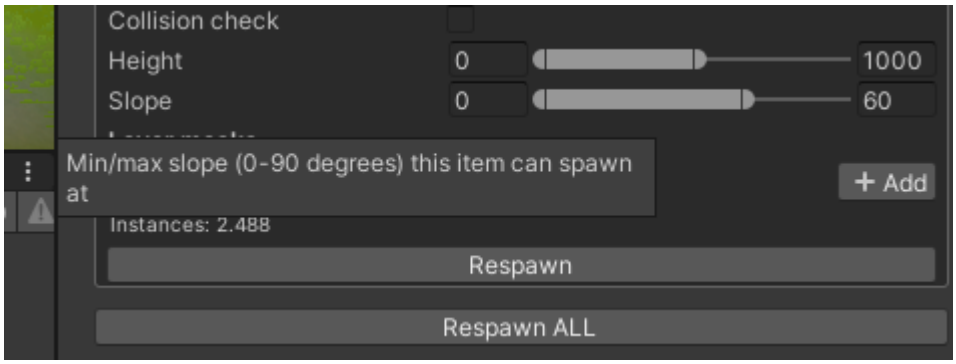
You'll notice it will immediately be spawned using the default settings.



You can add an additional prefab to the species, and specify a spawn chance for it. This way it's possible to spawn certain variants with a lower frequency (eg. dead or broken trees). Spawn point for a species will never overlap!

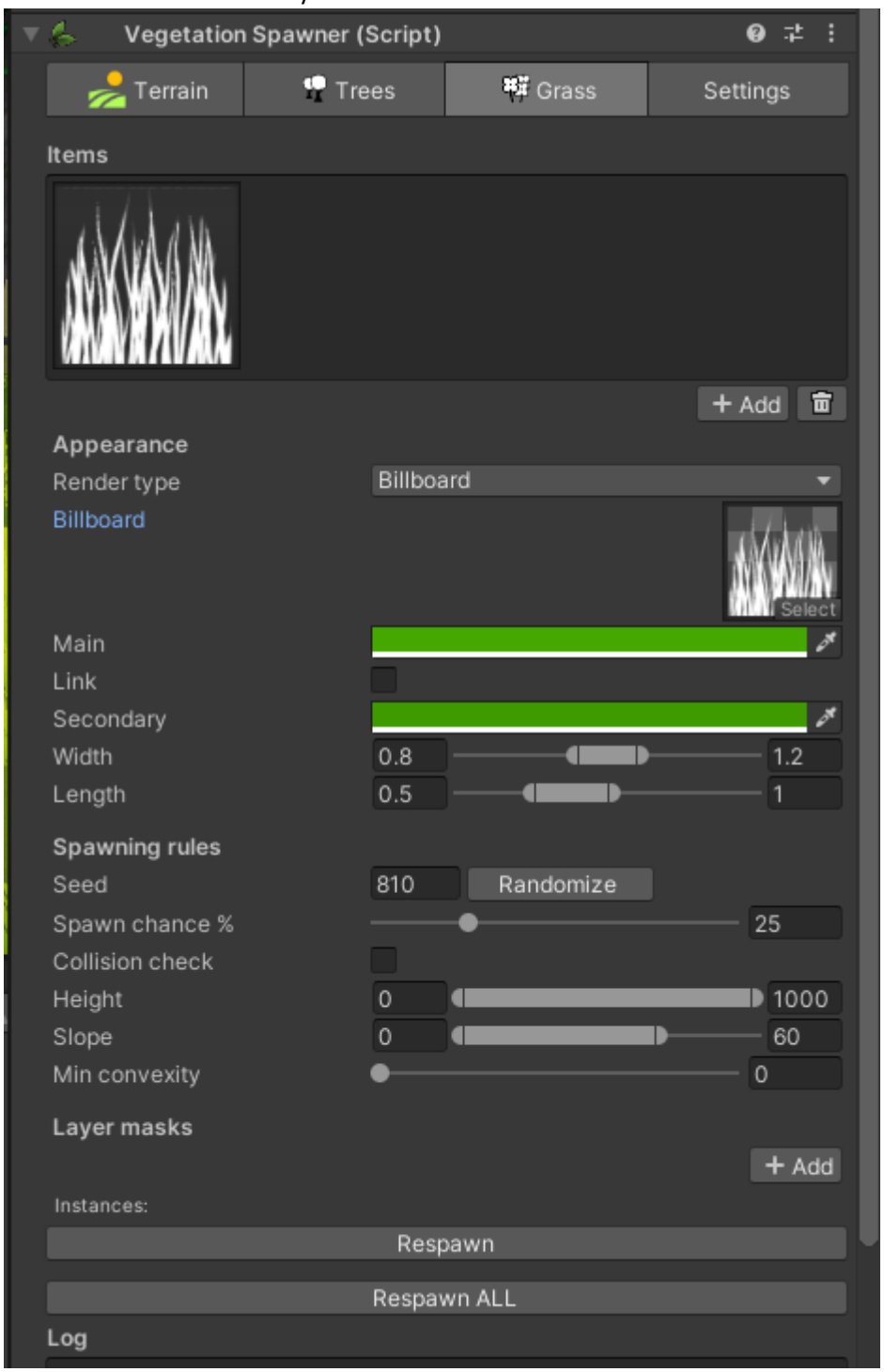


Changing most spawn rule parameters triggers an immediate respawn for the tree, this usually takes less than 0,2 seconds to process. Each parameter (when not obvious) has a tooltip description, mouse over to view it.



SPAWNING GRASS/PLANTS

This works in the same way as it does for trees



Note that spawning grass takes a lot longer than it does for trees (example: 6 seconds for 1 million instances on a 1x1km terrain made up of 4 tiles). Because of this, changing spawning rules isn't immediately reflected and requires to hit the "Respawn" button.

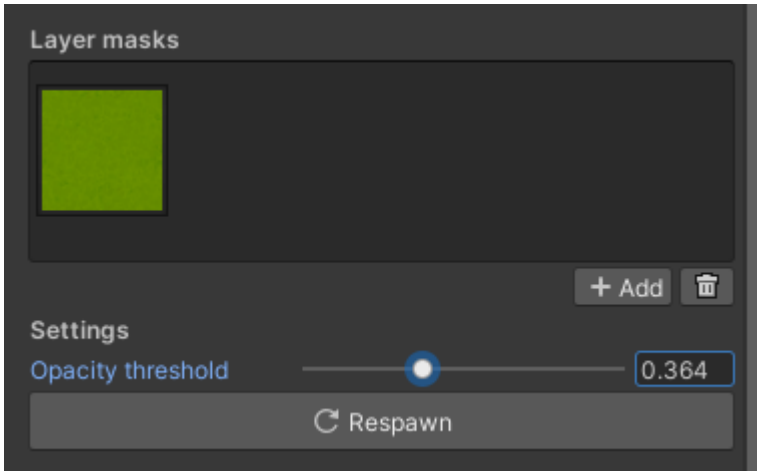
The terrain system forces you to use the built-in grass shader. If you want to use a prefab with a custom shader, add it as a tree instead. Unfortunately, the tree rendering isn't optimized for dense

spawning, so game performance will likely tank if you spawn items too close to each other. Using Nature Renderer is highly recommended for modern performance levels!

TERRAIN LAYER MASKS

This is an option available for both trees and grass, and specifies on which terrain material the item should spawn.

You can add a mask at the bottom of the inspector:



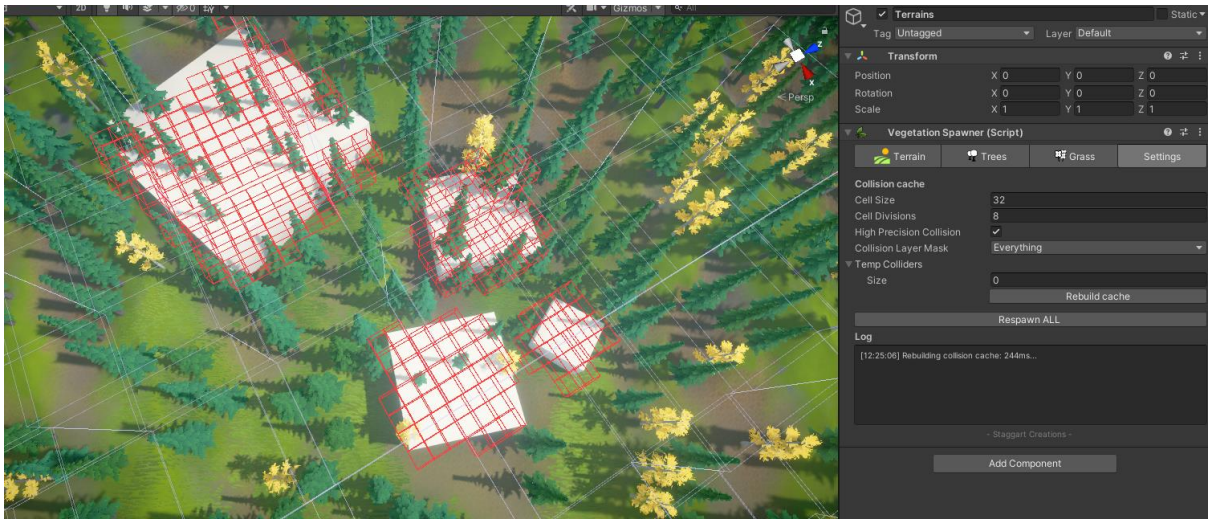
An item will only spawn on the materials added to this list, so is automatically excluded from other materials.

COLLISION DETECTION

In the Settings tab, you'll find the settings related to the "collision cache". This functionality checks the terrains for any colliders that are placed on it, and marks cell as occupied (red). If the "Collision check" option is enabled under an item's spawning rules, it will never be spawned inside a red/occupied cell.

This system is in place because its orders of magnitude faster than checking for collision on a per-item basis.

Increasing the amount cell divisions will make this more accurate, but it will take longer to rebuild the cache.



Note that this information cannot be serialized, this means the cache is rebuild if empty and called for. If you're looking to spawn vegetation at runtime/build, take into account that the first spawning job may take longer. You can also manually do this through the StaggartVegetationSpawner .VegetationSpawner.RebuildCollisionCache() function.

SCRIPTING

There are callbacks which trigger when a grass or tree item respawns. Custom scripts can subscribe to them to perform custom post-processing actions (eg recalculating navmeshes, spawning prefabs, etc).

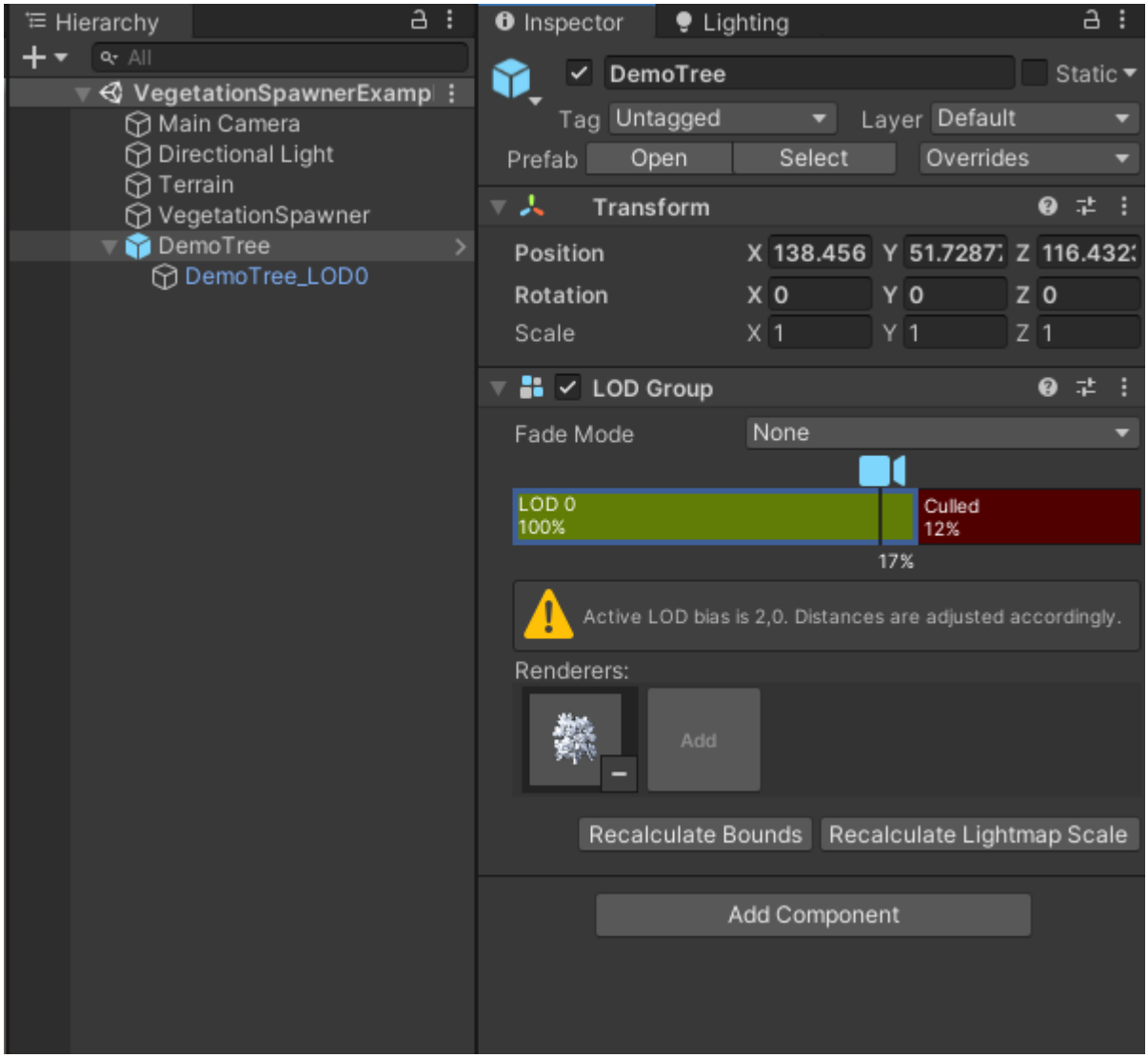
Example:

```
using UnityEngine;
using Staggart.VegetationSpawner;

[ExecuteInEditMode]
public class EventTest : MonoBehaviour
{
    private void OnEnable()
    {
        VegetationSpawner.onTreeRespawn += OnTreeRespawn;
        VegetationSpawner.onGrassRespawn += OnGrassRespawn;
    }

    private void OnDisable()
    {
        VegetationSpawner.onTreeRespawn -= OnTreeRespawn;
        VegetationSpawner.onGrassRespawn -= OnGrassRespawn;
    }

    public void OnTreeRespawn(SpawnerBase.TreePrefab item)
    {
        Debug.Log(item.prefab + " Tree respawned");
    }

    private void OnGrassRespawn(SpawnerBase.GrassPrefab item)
    {
        if(item.type == SpawnerBase.GrassType.Mesh) Debug.Log(item.prefab.name + "
grass respawned");
        if(item.type == SpawnerBase.GrassType.Billboard)
Debug.Log(item.billboard.name + " grass billboard respawned");
    }
}
```

KNOWN ISSUES

Console warning: *The tree DemoTree must use the Nature/Soft Occlusion shader. Otherwise billboarding/lighting will not work correctly.*

Join the club! Unity's vegetation system was designed for the dinosaur that is the Tree Creator tool. Instead, create a prefab with a LOD group:



This is also required for the random rotation of trees to work.