World Builder

Documentation

Version 1.0

Contact: superiorworlds@protonmail.com

Introduction

A complement to the Unity Terrain system for placing objects inspired by natural patterns. Tailored for Environmental Artists, powered by Gaea.

World Builder populates objects on the terrain using the Unity Terrain system via heat masks. It does this with a flexible concept called 'Layers' that scatter the objects using a single parameter of Minimum Distance via Poisson Disk Sampling algorithm. It is designed to augment a heightmap and texturing workflow provided by Quadspinner's Gaea, a node-based sculpting tool. This Mask-Based/Procedural hybrid permits unparalleled placement on a global scale while retaining pinpoint accuracy with the default Terrain brushes.

Capabilities

Layers are distinct from a popular concept of Biomes, because layers can overlap and be discontinuous (splotchy). A layer is not a bounded construct within Unity, but rather a sampling of a mask authored in Gaea. Technically, in nature, an ecosystem is a system within a Biome and is more defined by the biodiverse interaction between plant groups. Where groups clash becomes a new group and this can all be represented with thresholding data masks and their overlaps.

Whatever data map you export from Gaea to represent the target zone can then be sampled from for scattering of objects as a layer, or different layers sample sections of the same mask. So, if planned well, it can cover multiple systems or just a single ecosystem. For example, in your world, the riparian zone and forest edge zone might be combined to reduce the number of

objects loaded on the terrain, yet still give a specific environmental diversity and look.

The interface is clean and simple with only needed parameters

The layer system works independent of the terrain textures, so that they can be synchronized or can be managed independently. So, the quality of your result directly depends on the quality of the data maps provided and the composition of your layers.

This workflow is designed to be an iterative process even if you are a skilled Gaea user. Areas are sampled within Gaea, so it is all subject to the degree of accuracy cultivated there.

This simple and abstract concept allows for complex capabilities such as:

- Forest Edge, a transition from forest to non-forest
- Riparian, a transition from water to mainland

- Riverbed/Lakebed/Underwater ecosystems
- Place Trees and Details over multiple ecosystems at once
- Overlapping, common, and scattered positions/ecosystems
- No runtime processing overhead as items are stored instanced in the Unity Terrain
- Fully preserves the hand painting of Unity's tree and detail painting tools

Limitations and Workarounds

Detail Scatter Mode
Unity introduced a terrain setting in 2022 called `Detail Scatter Mode` that can break the placement of details. World Builder sets `Detail Scatter Mode` to `Instance Count Mode` when enabled. `Coverage Mode` is not supported and will not render details. If you are not able to print details check this setting in the `Terrain Settings`.

Terrain Size
As of Version 1, the recommended world size is 2km x 2km or smaller. Larger sizes are possible but issues may occur. This manifests as exponential slowdown in productivity as more samples are calculated, and other symptoms such as imprecise scattering as the mask samples are stretched over more area.

To work around this, a terrain can be split up into tiles but the heatmaps will also need to be sliced in an image editor program. This should only be attempted once the World Builder ruleset is perfected for a smaller version of the master terrain, so that when split occurs, all trees and detail elements will transfer over to the tiles. Then, copy/paste of the World Builder component can quickly sync on each tile terrain to allow population. A better solution for this may be coming in the near/distant future.

Multiple Terrains
The World Builder Editor can work with multiple terrains, but each terrain will need to have its own World Builder component to track the rules for that terrain. The editor will automatically prepare a terrain when the editor is loaded, but other terrains will need to be prepared manually. To change

terrains in the editor in order to change the values, just drag the object containing the Terrain component to the terrain slot at the top of the World Builder Editor and it will load those rules.

Must Use One Object Per Layer

We depend on the way Unity stores the trees and details, so currently trees or details cannot be used in multiple layers or they will overwrite each other. Instead, you need to create a composite heatmap layer where you want certain objects to be populated over the entire terrain (e.g. Flowers). Another way is to duplicate the prefab to use in another layer and they can be populated independently. **Generally, fewer items per layer will reduce population processing time as opposed to few, more populous layers.**

Unity inhibits the reuse of a tree object more than once. To work around this, you may simply duplicate the tree prefab in the project directory and re-add to the terrain, or you may add the same detail prefab multiple times. **World Builder has validation safeguards to alert of duplicates via the Unity Console and it will block population if the parameters are invalid.**

Hand Painting Trees and Details

The suggested workflow is to populate the world with World Builder, then touch up with the Terrain tools for painting more or erasing individual elements in that order. This is because World Builder will overwrite any already assigned Trees or Detail objects before placing them in the layer. A work-around is to add a hand painted alternative that is named with a 'flag' so it isn't chosen in a layer configuration by mistake. This might look like `HAND-PineTree01` so it shows up in an obvious way and is only used for hand placement. This reserved prefab slot will be isolated from the placement logic.

Organizing Many Objects

If you wish to populate many elements, the Unity terrain interface may make things difficult to locate. It is recommended to adopt a naming convention in a convenient format such as `lyrabbr-objname`. This could look like `Rip-Hedge1` or `EvgFor-Spruce1` to indicate Riparian zone or Evergreen Forest zone respectively. Keeping it under 10 characters will help it remain visible due to truncation in the Unity UI.

1024 Texture Resolution Limit

Designed around the Community Version of Gaea and 1024 resolution for layer data maps. Higher resolutions have not been tested but they may work. The Demo scene terrain is a standard Gaea project sculpted to 5km x 5km x 2.6km area at 1024 resolution. This was imported only as 2km x 2km x 1.3km heightmap at 4096 resolution. An In depth tutorial covering this is on YouTube at the following link: Youtube Playlist

Extras

Details Converter

Included in the package is a `Details Converter` script. The Details converter takes up to 10 prefabs with LOD groups and converts them into simple prefabs with the Mesh Renderer as the parent object. This allows use as a Terrain Detail. It requires the convention that the first child is named LOD0. If the prefabs you use do not use this convention the converter will not work and they will need to be changed manually.

Sink Trees Converter

Also included is a `Sink Trees Converter` script. This takes up to 10 prefabs with LODs (or child objects) and adjusts the Transform Y downward by the amount specified. A value of 1 will lower the objects' children by 1. This is so that trees, or other objects regarded as trees, can be placed lower in the terrain relative to their original state. This is to overcome the shortcoming in the Unity terrain that y position must be passed as 0 since the terrain component will determine the y and this cannot be offset during placement.

Camera Fly

Also included is a Camera Fly script. The Camera Fly script simply attaches to the main camera and allows fly controls in play mode as a quick way to explore the world at runtime.

Demo Assets

Also included is a Demo scene with a prebuilt terrain with 6 sample texture layers. This, with the included data maps, are there to help you get started with some Gaea exports and demonstrates the sophistication of the system.

An In depth tutorial covering this is on YouTube at the following link: Youtube Playlist

Roadmap

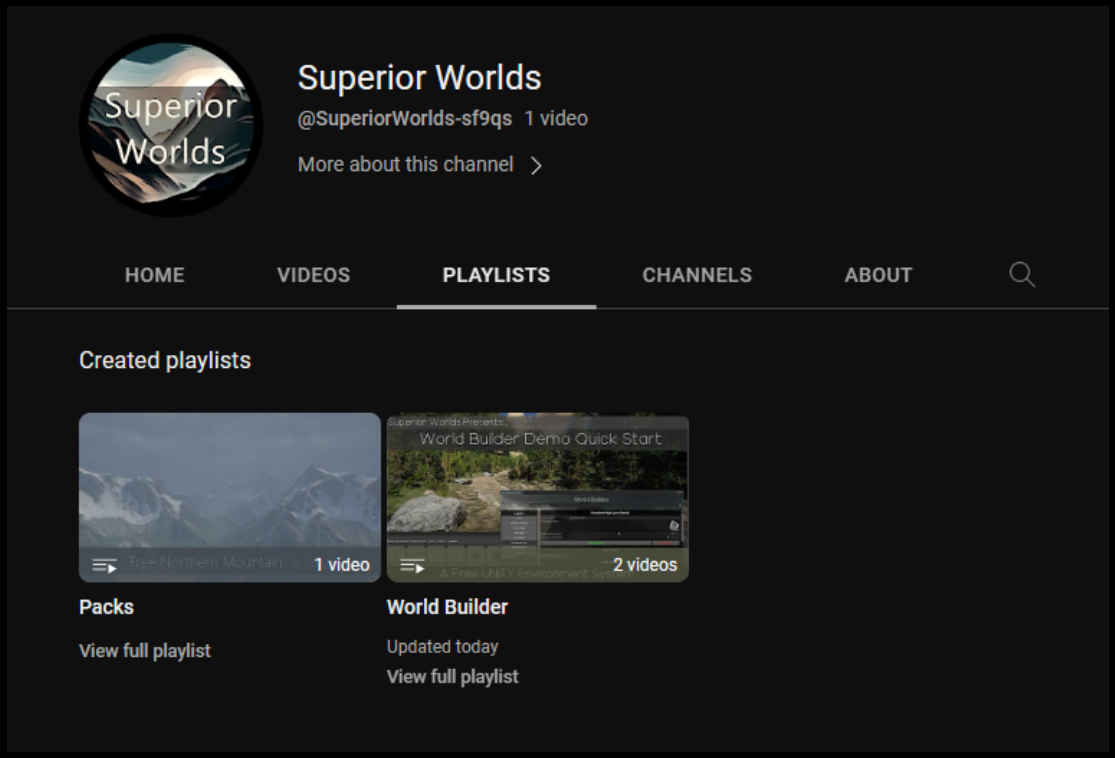
Bugs:
- Please report occurrences to superiorworlds@protonmail.com

- Performance: Populate speed up and efficiency
- Docs: More documentation and supportive materials to fully on board Gaea to Unity workflow
- Code: Refactor code to be more modular and extensible
- Code: Improve consistency of population between small and very large terrains
- Code:More automated tests to ensure stability against regressions
- Code: Support terrains greater than 2km x 2km
- UI: Reconfigure layer details pane to have sub tabs instead of a single stacked list

More specialized capabilities will be coming in paid addons.

Demo QuickStart

A growing in depth video tutorial playlist is available here:



1. Import your desired environment assets and open the World Builder `Demo` scene.
2. Pick an ecosystem to start with (e.g. hardwood forest) and add the appropriate trees, boulders, and other assets to the terrain object.
3. Open the World Builder Editor from *`Window > World Builder > World Builder Editor`*. Make sure the terrain has been assigned in the Terrain Object slot.
4. Add a new layer by clicking on the '`+ Add`' button on the left column.
5. Rename your layer appropriately (e.g. Hardwood Forest)
6. Select the forest heatmap image from the *`WorldBuilder/Demo/`* directory and drag it into the Heatmap Texture slot
7. Add as many tree and detail elements to the layer as were added to the terrain and choose them respectively.
8. Press the Populate `Layer` button and inspect for scattering imbalances. **If nothing happens,** *look at the console as it will describe the output and any validation errors*.
9. Adjust each tree and detail element's Minimum Distance value to achieve the desired look. For forest trees a good values is 9-12, small plants 3-5, grass 1-2 depending on how many items of similar type are placed together.

10. Add other layers and details as desired.
11. Perfect your world with the Unity hand placement tools.
12. Direct your questions to superiorworlds@protonmail.com

About

Hi, I'm Eric! I have been working with Unity since 1/2022 coming directly from a web application software background with a desire to teach. Since 2004, I have made web applications in Ruby on Rails language framework with a specialty in backend architecture, QoS tools, and automated testing.

World Builder was inspired by a desire to populate environment elements with more flexibility and respect to natural ecosystems than was available. This tool went through several design and implementation stages over the Spring and Summer of 2023. The assistance of ChatGPT and leadership of my Lord, and savior Jesus Christ enabled such a rapid development and release offered as a free asset.

Let's make Superior Worlds!