Terra Canvas Documentation

Introduction

Terra Canvas is a modular terrain-generation toolkit developed by Cody Dream Studio. It uses a Shader Graph–driven heightmap pipeline, giving developers complete freedom to shape terrain logic without noise presets or vendor-locked custom nodes. The system reads a Render Texture generated by your Shader Graph, converting that data into a Unity Terrain with fast iteration and predictable results.

The workflow is partially destructive, but controlled. Users can define protected regions, and anything inside those zones is left untouched. Our protection system currently preserves:

- **Terrain height**
- **Terrain paint (texture layers)**

Only unprotected areas are altered, and blending ensures they transition smoothly into protected zones.

To support safe iteration, Terra Canvas includes a simple heightmap backup system. It stores only height data. It does not save trees, grass, terrain details, or paint. We intentionally avoid auto-restoring backups during generation to keep iteration fast; users can restore manually when needed.

Current early-release limitations include:
• Tree placement and grass/details are overwritten during generation.
• These will be addressed in future updates, with proper rule-based handling and respecting existing content.

For height generation, Shader Graph provides full creative freedom. For object, tree, and detail placement, Shader Graph would be impractical and overly complex, so Terra Canvas uses rule-based logic instead. This keeps the system predictable and manageable while still allowing extensibility.

Terra Canvas supports runtime one-off terrain generation and is equally useful for creating large base terrains inside the editor before further manual work. It can be used to kick-start massive worlds quickly, then refined by hand.

Quick Start

After importing Terra Canvas into your project, there is one additional step before opening the demo scene. The demo relies on our optional graphical dependency: **Low Poly Stylized Nature**. This asset provides the visual elements used in the example environment. Terra Canvas itself does not depend on it, but the demo scene does.

Link to the asset:

https://assetstore.unity.com/packages/3d/environments/low-poly-stylized-nature-281338
Asset Name: **Low Poly Stylized Nature**

We strongly recommend importing this pack, especially for beginners, because it allows you to see the demo scene functioning exactly as intended.

Once both Terra Canvas and the optional pack are imported, proceed with the following steps:

1. **Open the provided demo scene included in Terra Canvas.**
2. **Press Play.**
3. **The terrain generation will start automatically.**

After a few moments, the terrain will finish building. You will see a handcrafted campsite area and a small bridge near a lake. These areas are untouched by generation—no floating objects, no objects buried into the ground. This demonstrates the protected-region system working correctly.

At this point, you can explore the **Terrain** prefab in the scene hierarchy. Its child objects contain each component that controls the individual generation passes. By tweaking these components, you can customize how the terrain is sampled, blended, protected, and applied.

This quick start gives you a complete, working example of the full pipeline, and from here you can begin experimenting with your own Shader Graphs and terrain configurations.

Quick Knowledge Guide

This section covers important technical details and behaviors that new users may not be familiar with when working with Terra Canvas. These points help you avoid common mistakes and understand how the system behaves under the hood.

---

1. The Material and Shader Graph Setup

Inside the Terra Canvas asset folder, under **Data**, you will find the Render Texture and the Material used for heightmap generation.
This Material references a Shader Graph. Terra Canvas includes two example Shader Graphs:
- **Base Terrain Generator**
- **Island Generator**

You can swap the shader in the Material to test different algorithms or assign your own custom Shader Graph for experimentation.

---

2. Keep a Buffer Around Protected Regions

In the Protected Region Manager, leave a generous boundary around important handcrafted areas.
If the boundary is too tight, you may see abrupt transitions where procedural logic tries to place terrain features (like mountains or cliffs) directly against villages, lakes, or buildings.
A buffer maintains natural-looking transitions.

---

3. Shader Graph Offset Precision Issues

If you use offsets in your Shader Graph—whether for randomness or large-scale terrain shifting—keep the values small.
Prefer offsets **below 5000**.
Very large offsets will cause floating-point precision issues, which lead to blocky or broken terrain results.

---

4. Seed Behavior and Runtime Randomization

The randomizer is deterministic by default.
If you provide a seed, you will always get the exact same terrain, trees, and placements—similar to how games like Minecraft behave.

If you need a new random seed each runtime generation, set the seed to **–1**.
This forces Terra Canvas to generate a fresh seed every time.

---

5. Shader Graph Flexibility and Stamping Subgraph

Shader Graph is powerful, but still limited compared to fully written shaders.
To help with this, Terra Canvas includes a **heightmap stamping subgraph**.
You can use this subgraph inside your custom Shader Graph to stamp heightmaps or shape patterns into your generation logic.

The base terrain generator already uses this approach, which is why your terrain often features a distinctive mountain shape by default.
Users creating new algorithms can reuse the stamping subgraph for more realistic or structured generation.

---

6. Pass Ordering and Custom Pipelines

Under the Terra Canvas asset folder, you will find **Passes** as ScriptableObjects.
These define how and when each generation pass runs.
Users can edit the order or remove/add passes to customize the workflow.

Example:
You could remove the “Save Backup” pass from the stack and instead always start by loading a stored backup.
This ensures every generation begins from the same original terrain instead of regenerating from previously modified data.

We do not use this flow by default because our priority is maximum iteration speed, but users are free to configure their own pipeline logic.

Acknowledged Limitations & Known Issues

While Terra Canvas provides a flexible and modular terrain generation system, there are some current limitations and known issues that users should be aware of:

---

1. Protection System Limitations

- The **protection system currently only preserves terrain height and painted terrain layers**.
- It does **not yet protect trees or terrain details**.
- Support for protecting trees and details is planned for future updates but is not implemented in the current early release.

---

2. Sphere Exclusion in Protected Regions

- While **path-based exclusions** include sphere-like shapes, box or custom region exclusions **do not yet support sphere exclusion**.
- This feature is recognized as useful and will be added in future releases.

---

3. Blending Artifacts at Handcrafted Boundaries

- In some cases, blending between protected hand-crafted areas and procedural terrain may produce **noticeable box-like or unnatural transitions**.
- To reduce this effect in the current version:
  • Make protected regions **larger**
  • Leave generous **buffer zones** between hand-crafted and procedural areas
  • Use terrain heightmap strategies to avoid abrupt height differences
- A proper fix for smoother transitions is planned for future upgrades.

---

4. Other Notes

- Procedural generation currently overrides trees, grass, and terrain details.
- Large offsets in Shader Graph may cause **floating-point precision issues**, producing blocky terrain features.
- Backup system stores **height data only**. Trees, grass, or splatmaps are not included.

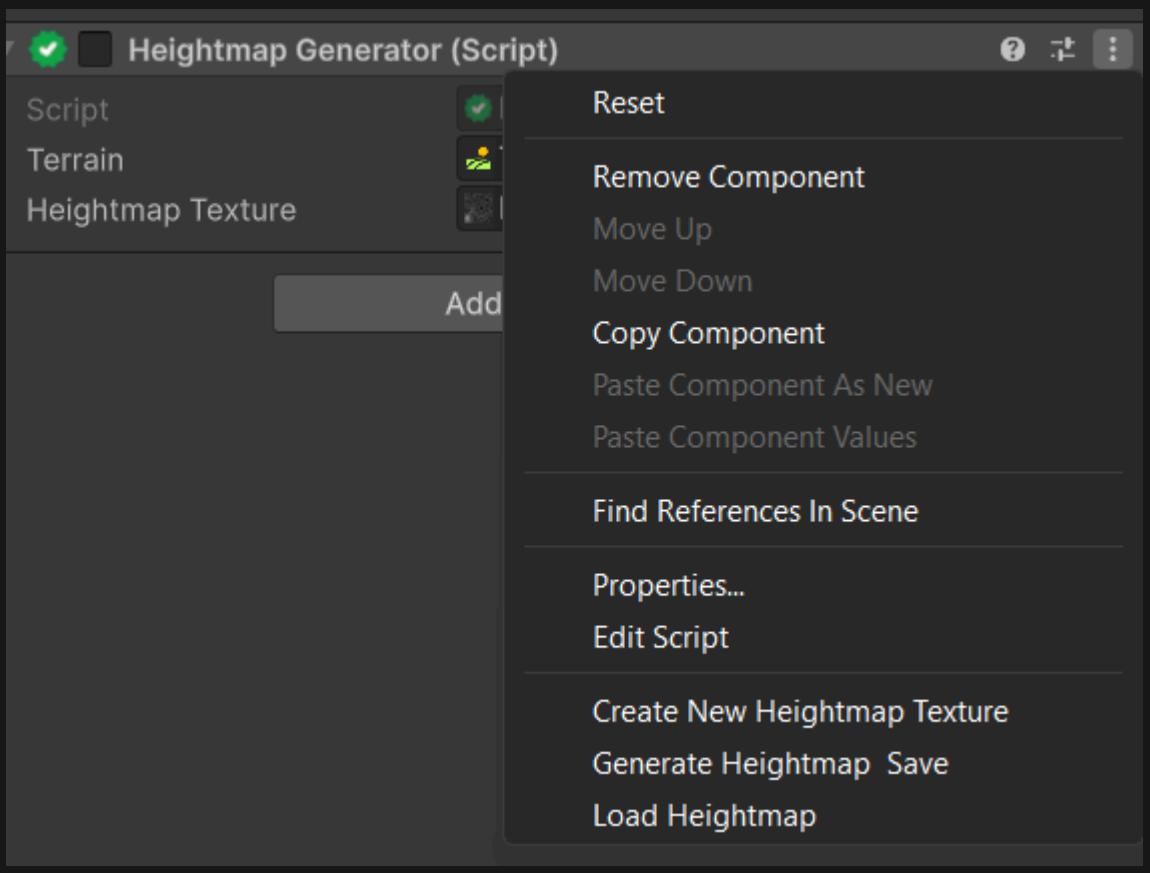
Final Notes & Caveats

Terra Canvas is designed to be **easy to use, self-explanatory, and modular**. While we do not provide fully comprehensive, per-component documentation, most of the system can be mastered through the demo scene, inspection of components, and experimenting with Shader Graphs.

---

1. Backup System

- The backup system **only works in the editor**.
- Users can create a new backup texture via the menu item



- Backup stores **height data only**; trees, details, and terrain paint layers are not saved.

---

2. Noise System

- The “CNoise” node/system refers to **Perlin noise** provided by the Unity Mathematics library.
- It is used as a building block for procedural terrain generation.

---

3. Rule Systems

- Rule systems govern **terrain painting**, **tree generation**, and **detail generation**.
- They are designed to be self-explanatory, enabling users to define areas, conditions, and exclusions.

Detail Generation Special Note:

- Detail rules allow **multi-type placement** within a single rule area.
- Rule areas can **exclude other areas**, allowing for overlapping rules with proper avoidance (e.g., grass avoids flower field areas).
- This gives advanced control over distribution while keeping the workflow manageable.

---

4. Scene Gizmos

- Demo scenes include necessary gizmos for visual guidance.
- Some gizmos are computationally expensive; configurable variables (size, display options) are provided to control performance.

---

5. Terrain Painting

- “Super Sample” mode enables **anti-aliasing** to reduce blockiness in terrain painting.

---

6. Tree Generation

- Tree generation settings, particularly **resolution**, affect both visual placement and **density**.
- Be careful when adjusting these values; higher resolutions increase density but may impact performance.

Thank You

Thank you for using **Terra Canvas**! Projects like this take a lot of effort, and we genuinely appreciate your support.

We want to emphasize that **Terra Canvas is completely free**—you can use it, experiment with it, and integrate it into your projects without any cost. If you enjoy using the asset, please consider leaving a review on the Unity Asset Store—it really helps us continue developing and improving tools like this.

If you have any questions, feedback, or run into issues, feel free to let us know by filling out this form:

https://docs.google.com/forms/d/1p3U8KN0PyXiGl9O2F11p_BsKiSETVnAlRsI5bMrPKnM/edit