Lil Leveler (Godot Plugin)

Table of Contents

[Installing and creating your first terrain 1](#_Toc91101828)

[Controls 2](#_Toc91101829)

[Overview of inspector properties 4](#_Toc91101830)

[Overview of the terrain editing toolbar 5](#_Toc91101831)

[Overview of vertex color toolbar 6](#_Toc91101832)

[Resizing the canvas 7](#_Toc91101833)

[Exporting your game 9](#_Toc91101834)

[Frequently asked questions (FAQ): 9](#_Toc91101835)

A picture containing text

Description automatically generated

## Installing and creating your first terrain

Lil Leveler should be installed under addons/ (lowercase addons/, not Addons/). The resulting folder structure should be as followed (right).

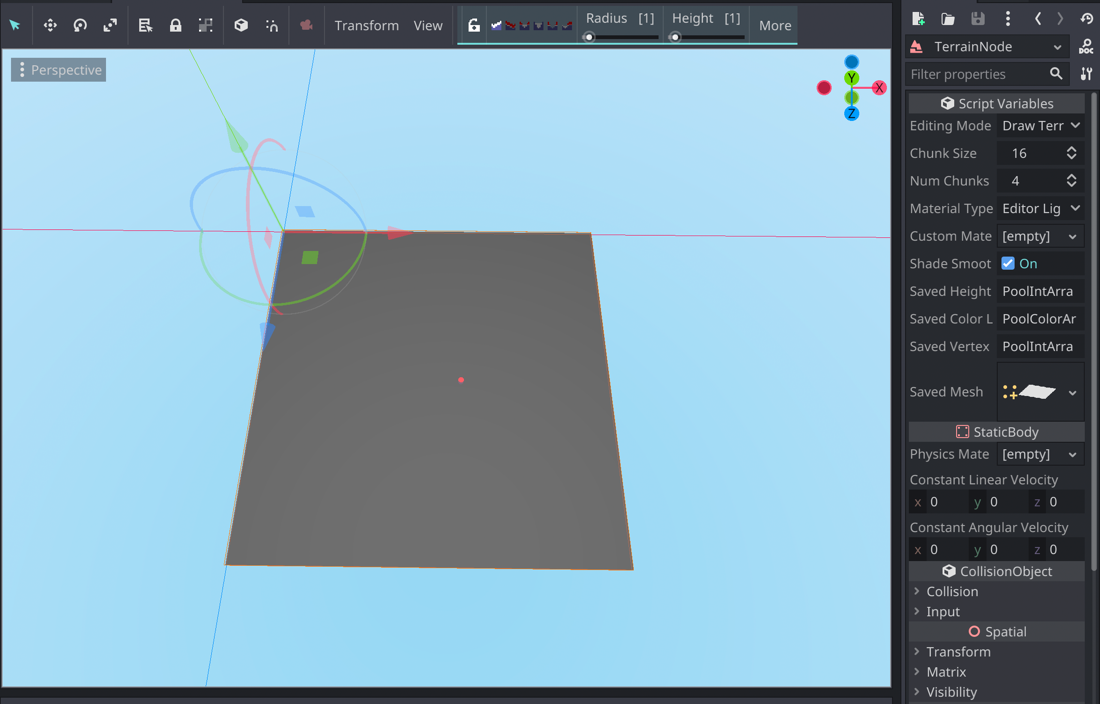
The plugin can now be enabled under Project Settings if it hasn’t been enabled already.

You can create terrain in an empty scene or a pre-existing one by creating a new node (click **+**) and searching/selecting the **TerrainNode** with the mountain icon.

Graphical user interface, text, application, chat or text message

Description automatically generated

If everything is installed correctly, you will see a large gray square, additional options on the top toolbar, and terrain properties in the inspector panel on the right (see screenshot below). At this point, you should be able to start drawing new terrain with your mouse.



## Controls

|  |  |
| --- | --- |
| *Middle mouse button* | Rotates the camera |
| *Middle mouse button + SHIFT* | Pans the camera |
| *Mouse wheel* | Zoom in or out |
| *Left mouse button* | Draw on terrain |
| *Right mouse button* | Quick erase/draw depending on the tool |
| *Mouse wheel + SHIFT -OR- +/-* | Change the brush height |
| *Mouse wheel + CTRL/CMD -OR- [/]* | Change brush radius |
| *Space* | Toggle lock height |
| *X* | Switch colors |
| *S* | Pick vertex color at mouse |

## Overview of inspector properties

The inspector properties are probably the first things you should set up when you create your terrain.

**Editing mode:** There are three editing modes:

* **Draw terrain:** Modify the terrain
* **Vertex colors:** Paint vertex colors
* **Preview/Published:** See the resulting terrain but otherwise has no other functionalities. You potentially reduce loading times in-editor if this mode is set.

**Chunk size / Num Chunks:** This is essentially your terrain size. The terrain is split up into multiple sections called chunks for faster editing. Each chunk possesses the size N x N, where N is the **chunk size**. The terrain is composed of M x M chunks, where M is **num chunks**.

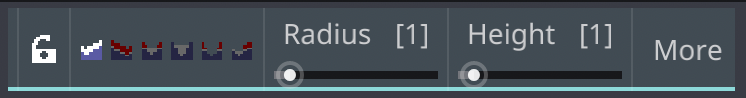
**Material type:** These are material presets for terrain editing or vertex painting. Additionally, you can choose to have a custom material, which you need to provide in the **custom material** space below.

**Custom material**: This is the spatial or shader material for the terrain. Make sure to set **material type** to **custom** to see the custom material applied on the terrain.

**Shade smooth**: Set true if you want smooth shading or false for flat shading. Just keep in mind that flat shading produces meshes with higher vertex counts since vertices are duplicated to make flat faces.

**For scaling terrain:** Just change the scale transform as you normally would for a node. The editor should work fine, but if you run into issues, scale everything to 1 to do your edits before scaling to the original value.

## Overview of the terrain editing toolbar



**Lock (Lock height):** Makes the brush affect only the current height.

**Brush modes:**

* Add: Adds new height
* Erase: Erases height
* Level: Sets height to the brush’s height
* Fill: Sets height if the current height is below the brush’s height
* Shave: Removes all height above the brush.
* Smooth: Smooths terrain to average height in the brush’s radius.

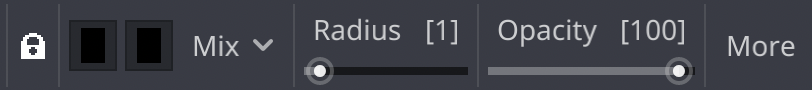
**Radius**: Sets the radius size of the brush.

**Height**: Sets the height of the brush.

**More:** A dropdown menu which includes more tools:

* **Resize canvas:** Adds or subtracts space for more terrain. (See later sections for details).
* **Clear all:** As the name suggests, clears the entire terrain and vertex colors.
* **Export as .OBJ**: Exports the terrain as a .OBJ mesh. Doesn’t include vertex colors in the exported mesh.

## Overview of vertex color toolbar



When you switch the **Editing Mode** to **Vertex Colors**, you will see a slightly different toolbar.

**Color swatches:** Sets the color of the brush. The first swatch is the active color. The second swatch is the back-up color. Switch between them by hitting the **X** key.

**Blend mode:** How color will be applied to the terrain (defaults to **Mix.**).

* **Mix:** Like standard painting program.
* **Add:** Adds RGBA components of the colors together. (Pre-existing + Current = Resulting)
* **Multiply:** Multiplies the RGBA components of the existing color by the RGBA components of the chosen color. (Pre-existing \* Current = Resulting)
* **Subtract:** Subtracts the RGBA components of the existing color by the RGBA components of the chosen color. (Pre-existing – Current = Resulting)

**Radius:** Sets the radius size of the brush.

**Opacity:** Sets strength of the blend mode.

**More:** The same menu as before. See above.

## Resizing the canvas

While you can adjust the terrain canvas size from the inspector, for finer control, you’d want to use **Resize canvas** under **More**in the toolbar when the editing mode is not on preview/publish. Upon selecting this option, you will see a popup like that shown below.

Graphical user interface, application

Description automatically generated

A picture containing application

Description automatically generated**Chunk size** and **number of chunks** should be self-explanatory. What may be confusing is the square grid. This grid allows you to select how space is added or subtracted from the current terrain.

When the center square is selected, space is added/subtracted equally on all sides. If the top left square is selected, space is added/subtracted on the right and bottom sides. See the diagrams on the right for how resizing works.

In editor, the gimbal is always at the top left corner, so you can use that as reference. I’m planning to re-work resizing to include a mini-map for previewing the resized canvas to make the process a little less abstract.

## Exporting your game

All terrains generated by this plugin should work on all platforms because the GDnative scripts are not used during runtime, only in-editor.

The only exception is if you tinker with the plugin itself to run native scripts during runtime. Because I’ve provided dynamic libraries for only Windows desktop, Mac Intel, and Mac M1, these are the platforms that are supported. Just make sure that **Runnable** is enabled before exporting to these platforms if you plan on running GDnative scripts.

## Frequently asked questions (FAQ):

**What values are recommended for chunk size and num chunks?**

Overall, it doesn’t matter too much because in-game, the terrain is generated as a single mesh, not individual chunks. You can adjust both until you have your desired map size. If editing the terrain is slow, you can lower the **chunk size** and increase **num\_chunks** so you get your desired map size.

**How do I texture the terrain?**

You probably want to use a triplanar shader. A simple triplanar shader has been provided in the demo. Other than that, you can export the terrain as a .OBJ and perform UV unwrapping, but this might be a lot of work.

**This isn’t real vertex painting.**

I know. I’m sorry. Use shaders.

**Can I make non-square terrain?**

No.

**Can I make terrain editable in-game?**

By default, this function isn’t explicitly implemented, but there are functions provided in TerrainNode.gd that you can use to implement editable terrain yourself (see *draw\_at()*).

**Can I make overhangs or caves?**

No, terrain generation still uses heightmaps.

**Does this plugin include any LOD features?**

No, LOD has not been implemented. Only when I’ve managed to grasp the Transvoxel algorithm will I then try to implement LOD functions.

**Can I use this plugin to create large open worlds?**

While I won’t say no, I won’t recommend it. For very low-poly worlds—possibly, but for most other use cases, you would need a chunk and/or LOD system, both functionalities that this plugin does not currently provide.