Digger

Introduction

Digger is a simple yet powerful tool to create natural caves and overhangs on your Unity terrains directly from the Unity editor.

More and more AAA games add gentle overhangs and caves to their environment to make it more realistic, more interesting and more diversified.
A common way to do this is to create 3D models with an external modeling tool, and then place them manually over the terrain and blend them with it. This is tedious, unpractical and inefficient unless you have an army of 3D artists and level designers... and even in that case, you'd probably want them to focus on more valuable work.
This is where Digger comes to action. No more external tool required, no more loss of time, no more headache. It lets you create caves and overhangs on your terrain directly within the scene view, in a few clicks.

With this tool, you will be able to:

- Dig in your Unity terrain just like if it was a smooth voxel terrain.
- Create overhangs (the opposite of digging).
- Apply different textures on the overhangs, in the caves, etc.

However, you won't be able to:

- Dig in real-time and at runtime.
- Generate caves procedurally.

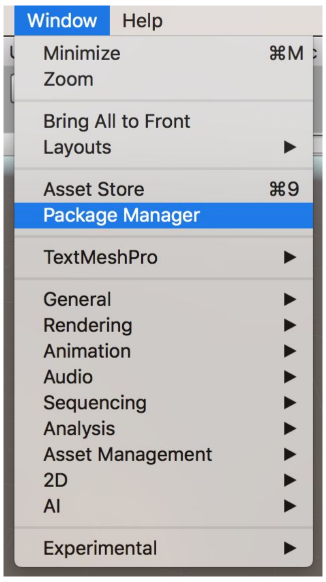
If you need these features, you should get a full voxel-based terrain solution, like Ultimate Terrains.

Digger can be downloaded from the Asset Store.

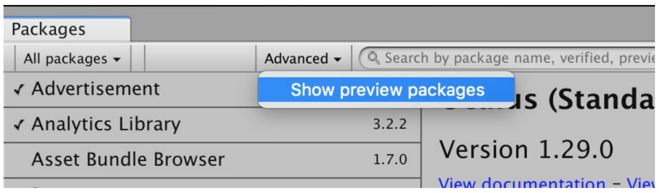
Getting Started

Digger is very easy to setup, but it requires you to install 3 packages.

Open a project, and open the Package Manager (menu *Windows* > *Package Manager*).



In the **Package Manager** window, click on "Advanced" and enable "Show preview packages".


Packages
All packages
Advanced
Search by package name, verified, previe
Show preview packages
✓ Advertisement
✓ Analytics Library 3.2.2
Asset Bundle Browser 1.7.0
Version 1.29.0
View documentation - View

Install the latest version of the packages "**Mathematics**", "**Collections**" and "**Burst**".

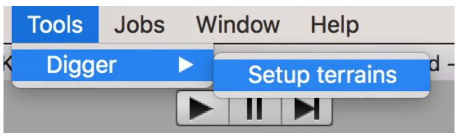
Import **Digger** into your project (from the Asset Store).

From now on, Digger should be imported and you should not have any error in the console. There should be a new menu: *Tools* > *Digger*.

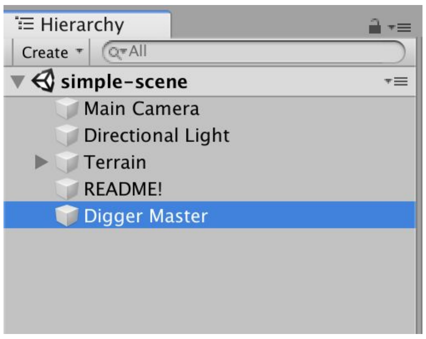
You are ready to use Digger.

Open a scene with a terrain (you can open "simple-scene" in *Assets/Digger/Demo* for example) or create a terrain in a new scene. Configure your terrain layers as usual and modify your terrain as usual (raise or lower height, etc.).

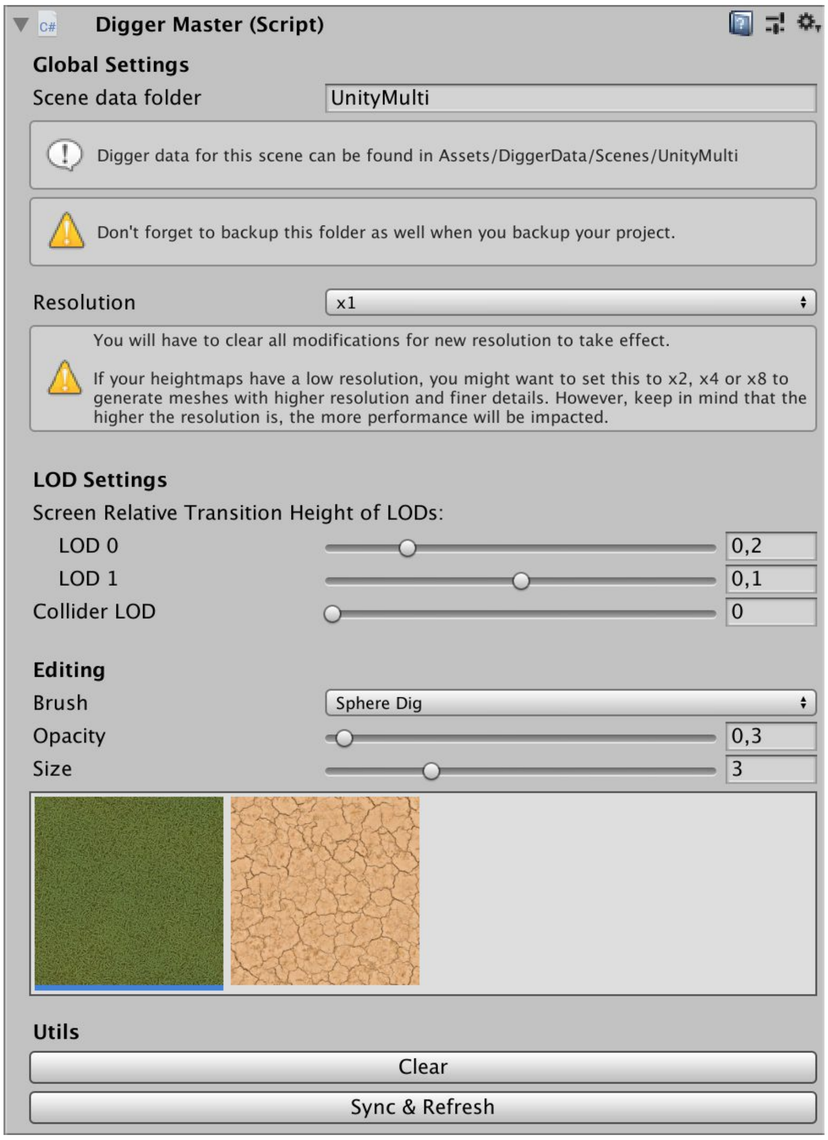
Once this is done, click on *Tools* > *Digger* > *Setup terrains*. This will prepare texture arrays for Digger material, add Digger System to all terrains in the scene and add Digger Master.



Click on Digger Master in the hierarchy of the scene to display the Digger Master inspector.



The Digger Master inspector looks like this:



Digger Master (Script)

Global Settings

Scene data folder: UnityMulti

Digger data for this scene can be found in Assets/DiggerData/Scenes/UnityMulti

Don't forget to backup this folder as well when you backup your project.

Resolution: x1

You will have to clear all modifications for new resolution to take effect.

If your heightmaps have a low resolution, you might want to set this to x2, x4 or x8 to generate meshes with higher resolution and finer details. However, keep in mind that the higher the resolution is, the more performance will be impacted.

LOD Settings

Screen Relative Transition Height of LODs:

LOD 0: 0,2

LOD 1: 0,1

Collider LOD: 0

Editing

Brush: Sphere Dig

Opacity: 0,3

Size: 3

Utils

To start digging, just click somewhere on your terrain!

Note: the first time you dig, Unity will freeze during about 1s. This is because the Burst compiler needs to compile internal Digger jobs.

Details of each field:

1. *Scene data folder*: Digger will automatically persist data in Assets/DiggerData/<scene-data-folder>. By default, this is the name of the scene. You can change it if you want, but don't forget to rename the directory as well.

2. **NEW!** *Resolution*: by default, Digger generates meshes that fit to terrain's mesh, which is directly related to heightmap resolution (and terrain size), but you can tell Digger to use a finer resolution (respectively, 2 times, 4 times or 8 times the terrain's mesh resolution) thanks to this parameter.

3. *Screen Relative Transition Height of LODs*: adjust these sliders to tell at which distance from the camera the cave/overhangs meshes should switch between LODs. The bigger it is, the closer you will have to be from the object to get the highly detailed mesh.

4. *Collider LOD*: lets you change the Level Of Details of the collider mesh. If you want accurate collisions that fit exactly to the ground, set it to 0. If you want better performance and don't mind to have a lower accuracy, increase it to 1 or 2.

5. *Brush*: lets you choose the action to perform between digging terrain, raising overhangs, reseting (reset to terrain height but do not restore terrain details objects), or painting.

6. *Opacity*: the speed at which you will dig/add mater to the terrain. This has no effect on reset and paint brushes.

7. *Size*: the size of the brush.

8. *List of textures*: lets you choose which texture to use.

9. *Clear*: this will **clear all modifications** you've made to the terrains with Digger, but **it won't restore terrain details objects**. **This cannot be undone.**

10. *Sync & Refresh*: forces Digger to synchronize with terrains and recompute everything. This is useful if you changed terrain textures or heights.

Integration with CTS

CTS (Complete Terrain Shaders) is supported by Digger, but as things stand, you won't be able to change textures in caves or on overhangs. It will pick-up the terrain texture. Future versions of CTS might allow to fix this.

Troubleshooting

When I do some raycasts, it still detects the terrain surface on holes. How can I do to raycast through terrain holes?

Until Unity releases official terrain holes feature, you have to use DiggerPhysics.Raycast method to solve this.

I changed the textures of my terrain, and now caves don't blend with terrain properly because they still use the old textures.

Click on 'Sync & Refresh' button of the Digger Master inspector.

If that's not enough, delete texture arrays belonging to the terrain in *Assets/DiggerData/TextureArrays* and click on 'Sync & Refresh' button of the Digger Master inspector. This will force Digger to recompute texture arrays.

I changed the heights of my terrain at some place, but Digger doesn't see it.

Click on 'Sync & Refresh' button of the Digger Master inspector. If you already dig/add mater at this place, you will probably have to reset it using the *reset* brush.

What should I save when I backup my project?

Digger persists everything in **Assets/DiggerData**. If you save this folder, you're fine.

Do NOT backup your project using only *Export package* feature of Unity, because it won't keep internal Digger data that is put in ".internal" folders for each terrain (typically in Assets/DiggerData/<scene-name>/<guid>/**.internal** folders).

Instead, you have to copy paste your project folder manually.

I exported *DiggerData* folder using *Export package* feature of Unity, but it didn’t keep modifications made with Digger.

Internally, Digger persists voxel data in Assets/DiggerData/<scene-name>/<guid>/**.internal** folders which are not exported by Unity (Unity ignores folders starting with a **dot**). You’ll have to copy past it manually.

Digger meshes do not perfectly blend with the terrain in general.

There should not be any visual difference between the terrain and the meshes generated by Digger.
However, there are a few reasons why it might happen:

1. Lightmapping is enabled on the terrain. You should disable it from terrain settings as Digger doesn’t support it.
2. When the terrain is selected in the editor, it is looks more *orange*. Just unselect it.
3. In some particular cases, because ‘*Draw instanced*’ is enabled on your terrain. See below:

Digger meshes do not perfectly blend with the terrain at some places, in particularly on slopes.

This can happen with some textures on some parts of the terrain if you enabled ‘*Draw instanced*’ in the terrain’s settings. This is because terrain normal is a bit different when this is enabled, so it can interact with lights a bit differently, making Digger mesh distinguishable.

In most cases, this is not noticable (all screenshots on the Asset Store page have been made with a terrain where *Draw instanced* is enabled for example), but if you find that it is too visible, you can simply disable *Draw instanced*.

Some of the terrain textures are missing in Digger inspector, and they are not rendered on Digger meshes.

This happen when you have some terrain layers that don’t have a normal map. Digger requires all terrain layers to have a normal map.
See this page for more information about terrain layers:
https://docs.unity3d.com/Manual/class-TerrainLayer.html

F.A.Q

Does Digger support the built in navmesh system?

No, because the navmesh system will still think the terrain is there (it won't take terrain holes into account).
We will have to wait for Unity 2019.2 (or later) and its official, clean, terrain hole feature for this.

Is this an extension to Unity's built-in terrain system or an entirely proprietary system?

It's an extension of the Unity's built in terrain system. Basically, it just cuts the terrain where needed and generate meshes for caves and overhangs. In the end, you get a Unity terrain + some meshes (with different LODs because Digger generates automatically different LODs for each mesh).

Transition between meshes and terrain is seamless because they basically use the "same" shader (actually two different versions of the same shader: one for the terrain and one for the meshes).

Can caves be altered/created at runtime? (AKA player terraforming)

No, only edit time. For real-time, you should get a full voxel terrain engine like *Ultimate Terrains*.

Does this affect collisions? Meaning I can walk my character in these holes and into the caves?

Yes it solves collision thanks to automatically generated triggers + collision meshes. At the end of the video (on Asset Store page), you can see some balls falling into the caves. There is no need to add anything to your objects to make it work.

However, to get Raycast working, you will have to use DiggerPhysics.Raycast method.

How can I do to make Raycast working through cave entrances?

Until official terrain hole feature (maybe in Unity 2019.2), you have to use DiggerPhysics.Raycast method.

Support

To get support, join us on Discord: https://discord.gg/C2X6C6s

Latest version of this documentation can be found here:
https://drive.google.com/open?id=1sw8liMCBqwMDise_CsL2c5hrLxZQZEQrvNyJ8XbSDJE