3D Tilemap Manual:

Use the 3D Tilemap package to easily and quickly white-box and create levels for your games, placing tiles and prefabs with just a click; you can also bake your levels mesh, combining all the meshes into one, and save the resulting mesh.

For example, use 3D Tilemap to create a 3 floors level with the included **self tiling floor**, using it both as the floor and the walls in your level; then bake and save the all the floors into one mesh, allowing you to use the level you just created in any scene you want.
The 3D Tilemap package also allows you to add your own prefabs to use as tiles, and save the Tilesets that you make to use them later on.

To install this package, follow the instructions in the [Package Manager documentation]:
https://docs.unity3d.com/Packages/com.unity.package-manager-ui@latest/index.html.

Table of Contents:

Quick Start:

To start using the 3D Tilemap:

- Open the Level Creator Window from the "Tools" tab.

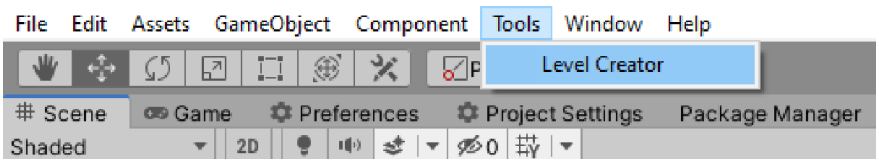


Image - 1
(Window Creation)

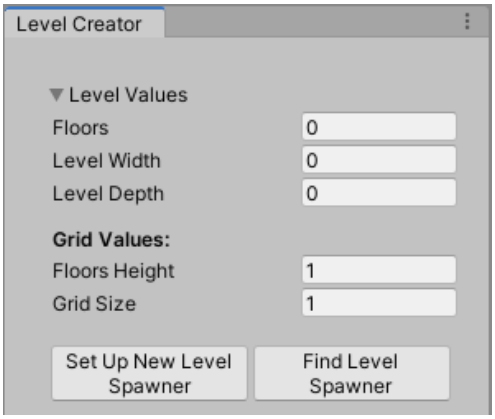


Image - 2
(Level Creator Window)

- After you open the window, set your levels values, Number of Floors, Width and Depth.
- You can also set the height of every floor, and the size of each tile in your level.

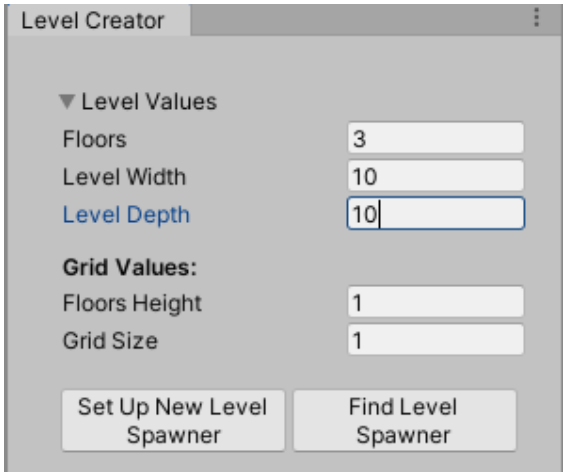


Image - 3
(Setting Level Values)

- Once you have set the values, you can press "***Set Up New Level Spawner***", this will set up a Level Spawner object in your scene.

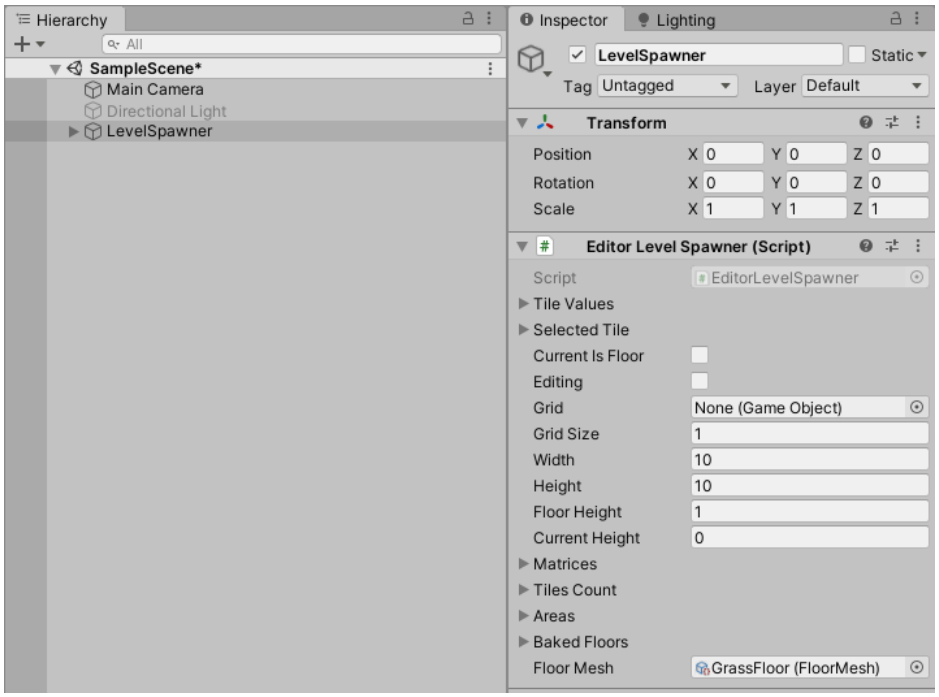


Image - 4
(Created Level Spawner)

- After this you will be able to press "Start Editing" and create your own levels!

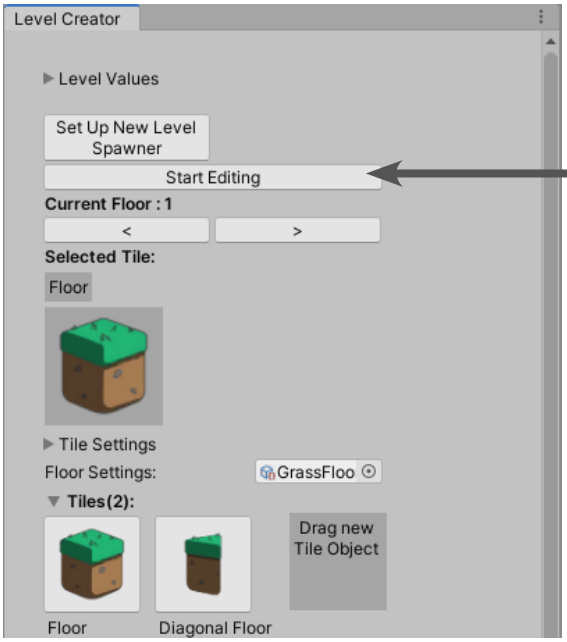
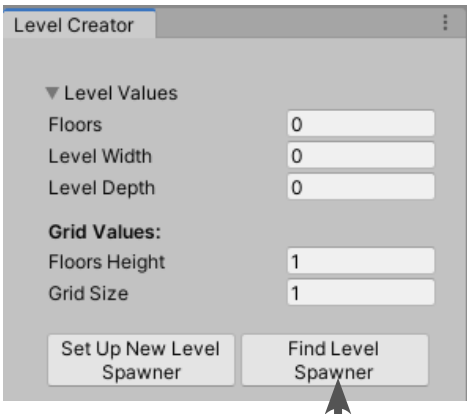


Image - 5
(Start Editing)

Image - 6
(Find Level Spawner)



NOTES:

- If your scene already has a Level Spawner Object, you can press **"Find Level Spawner"** to set it as your designated Spawner.
- If there is already a Level Spawner in the Scene and you press "Set Up New Level Spawner" the original object will be DESTROYED, along with ANY TILES you had already placed.

Edit Mode:

While Edit Mode is turned on, you will be able to create your levels inside of your scene, by creating different tiles in the level Grid.

Edit Mode Hot-keys:

Create Tile - **Left Mouse Click**

Delete Tile - **Ctrl + Left Mouse Click**

Change Current Floor - **Square Brackets [&]**

Rotate Tile* - **Space Bar**

* Only Tiles set as "Can Rotate" such as the Triangle Floor, will be rotated.

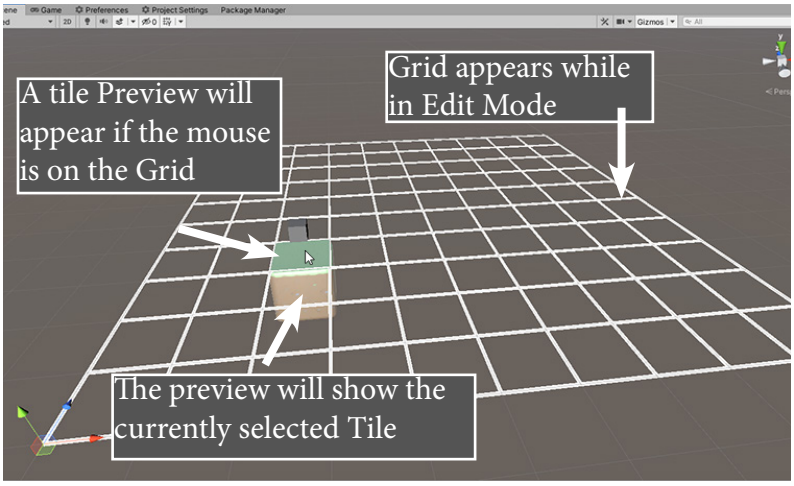


Image - 7
(Edit Mode Grid)

Image - 8
(Delete Example)

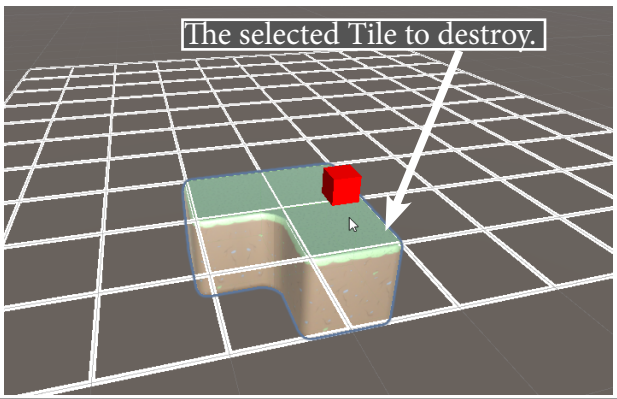
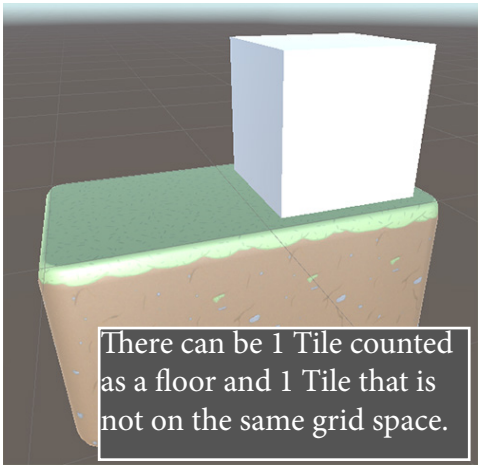


Image - 9
(Floor Tiles)



With the 3D Tilemap Package, creating a level is as simple as choosing a tile and clicking on the grid.

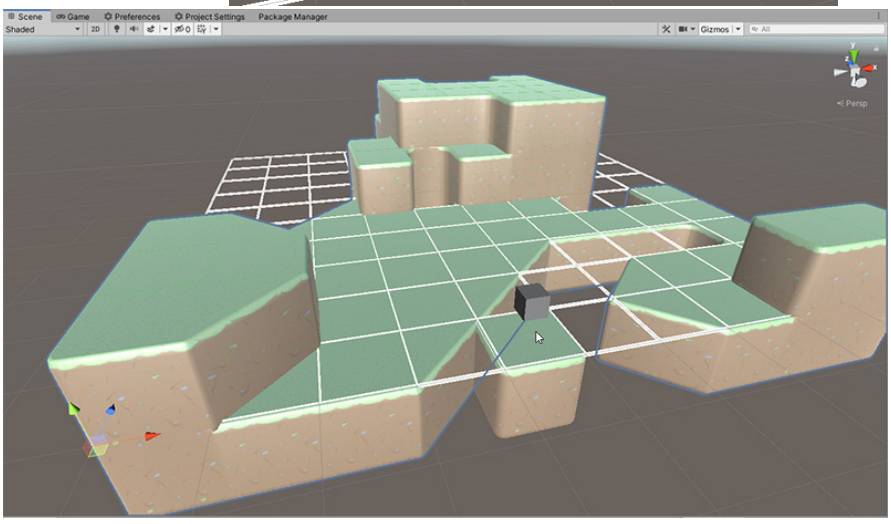


Image - 10
(Example Level)

Tileset:

In the Tileset Section of the Window, you can find these options:

- Selected Tile & Tile Settings: This area will display the currently selected Tile, along with its properties. The properties will only be able to be edited for any new Tile, the Default Tiles values cannot be edited.
- Floor Settings: The settings that the Self Tiling Floor will use, includes the Meshes and Material used.
- Tiles (Tileset): This will show your current tileset, always starting with the 2 Default Tiles: Floor, and Diagonal Floor.
- Drag new Tile Object: Drag a Game Object or mesh to create and add a new Tile.
- Save Tileset: After adding at least 1 new Prefab (Tile) to your Tileset, you will be able to save it, to allow ease of access at a later time.
- Load Tileset: If you have a Saved Tileset you can Load it through this option. If you do, your current Tileset will not be saved.
- Reset Tileset: You can use this option to reset your tileset to the default one, deleting any extra tile that has been added. If you do, your current Tileset will not be saved.

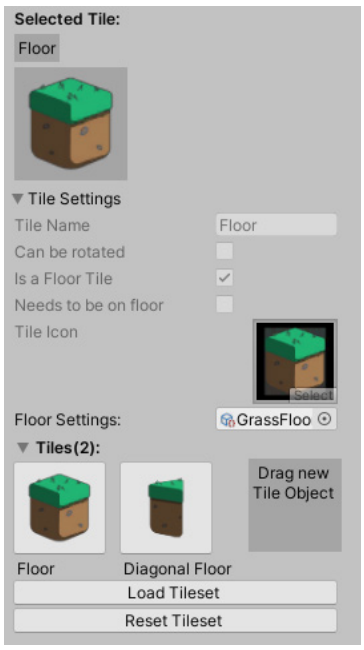


Image - 11
(Tiles Areas)

Selected Tile and Tile Settings:

In this section the currently selected Tile will be shown, along with its properties. You will be able to edit the properties of any of the custom tiles added by you. (The properties of the default tiles will be shown, but you will not be able to edit them).

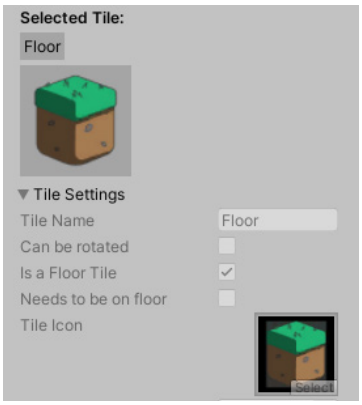


Image - 12
(Selected Tile)

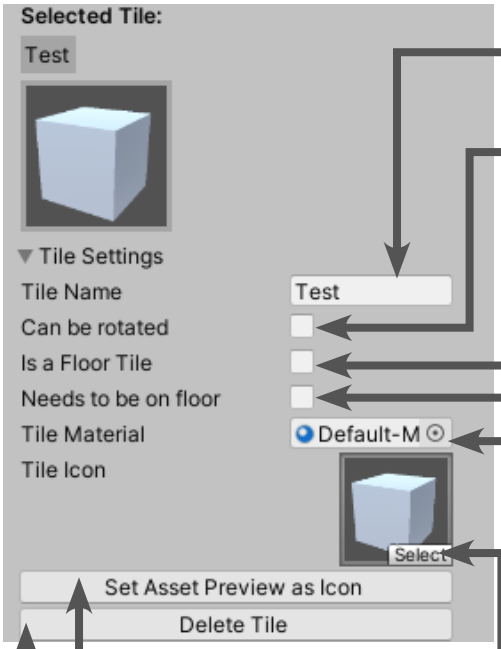


Image - 13
(Tile Options)

The name you want displayed for your Tile.

Set to true if you want to be able to rotate the tile while placing it. (90° increments)

Set to true if you want the tile to be counted as a Floor Tile.

Only needed for the tiles that are NOT set as a floor.

If set to TRUE, when you create one of these tiles, if there is no "Floor Tile" on the same square, a Default floor will be spawned.

If the tile object has a Mesh Renderer, it will allow you to change its assigned material from here.

Its recommended to place the Pivot Point on this tiles on the bottom of the mesh .

Its recommended to place the Pivot Point on this tiles on the top and centre of the mesh .
Its also recommended to have them be 1 Unity unit in size.

The Game Object or Prefab* that you want to use as a Tile.

If this field is left empty the object Asset Preview will be used for its Icon.

If you want to set the Asset Preview as the Tile Icon press this button.

If you make changes to the Tile's prefab, you will need to select the prefab in the hierarchy before pressing this button. This will update the preview of the prefab.

If after adding a custom tile you wish to get rid of it, you can delete it using the "Delete Tile" Button.

Floor Settings:

Image - 14
(Default Floor Settings)

This field will only take a FloorMesh ScriptableObject*, and will be used to set the material that the Self Tiling Floor will use, along with all the meshes it needs in order to function correctly.
The Package already includes the Default Settings that will be used, as a more advanced process, the user can edit the included textures or create new ones to achieve a different look on the Self Tiling Floor**.** To see how, look at the Advanced Options at the end of the Manual.
*For more information on Scriptable Objects visit:
https://docs.unity3d.com/Manual/class-ScriptableObject.html

Tileset:

The "Tiles" drop-down will show all the tiles included in the current Tileset. The number in brackets beside the title will show the amount of tiles included in the current tileset.
The Tool will always start with the 2 Default tiles.

Image - 15
(Default Tiles)

Default Tiles:

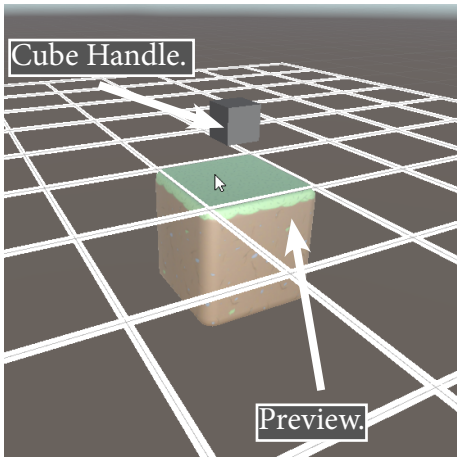
The default Tiles that come with the Package, are:
The Self Tiling ***Floor***, and the Self Tiling ***Diagonal Floor***.
Both Default Tiles are counted as Floor Tiles.
This Tiles can neither be deleted or edited.

Image - 16
(Floor Button)

Floor (*Self Tiling Floor*):

The Floor is the main Tile used for creating levels, used by clicking on any of the squares on the grid. This tile has a cubic shape.

Image - 17
(Floor Preview)



While hovering the mouse over the grid, a preview of the floor will appear, along with a grey handle above the preview.

This, along with the Diagonal Floor are the 2 special Tiles included with the package. When its created on the grid, it will analyse the neighbouring tiles and change its appearance based on its neighbours. (This uses the Floor Settings to get the appropiate meshes and materials).

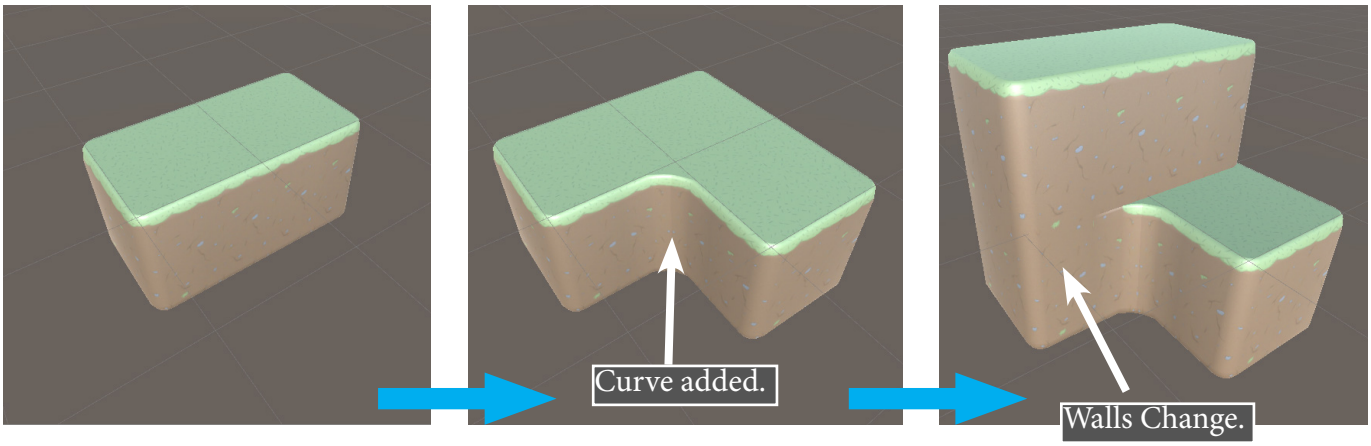


Image - 18
(Self Tiling Floor Example)

Diagonal Floor (*Self Tiling Floor*):

The Floor is the second main Tile used for creating levels, used by clicking on any of the squares on the grid. This tile has a triangular prism shape.

Image - 19
(Diagonal Floor Button)

While hovering the mouse over the grid, a preview of the floor will appear, along with a grey handle above the preview. While previewing the tile, you can press the Space-bar to rotate it 90°.

The Diagonal Floor, along with the regular Floor are the 2 special Tiles included with the package. When its created on the grid, it will analyse the neighbouring tiles and change its appearance based on its neighbours. (This uses the Floor Settings to get the appropiate meshes and materials).

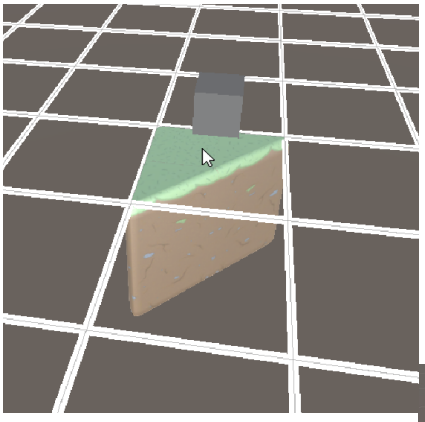


Image - 20
(Diagonal Floor Preview)

Image - 21
(Diagonal Floor Example 1)

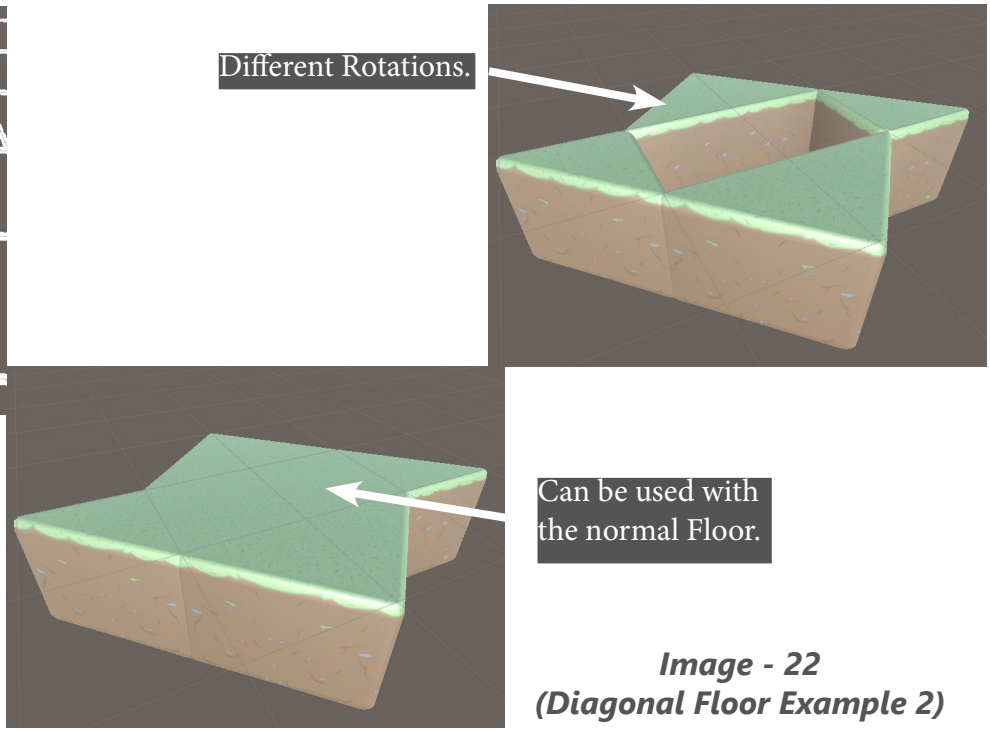


Image - 22
(Diagonal Floor Example 2)

NOTE:
Since the "Self Tiling Floors" will scale up and down to match the set "Floor Height" and Grid Size, if the values are not even, some stretching will appear in this tiles textures.

Because of this, if you plan on using the Default Tiles; it is recommended to use the same or close values in both to avoid the stretching.

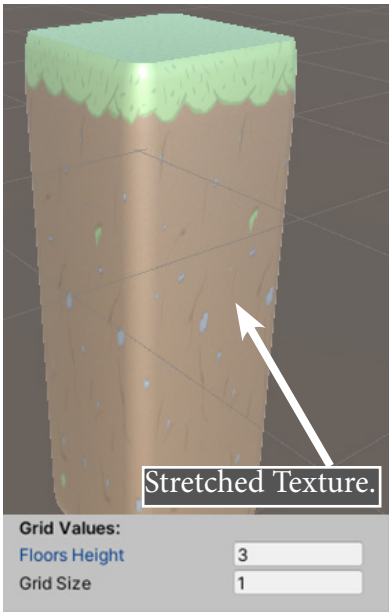


Image - 23
(Stretched Texture)

Add new Tile:

To add a new tile to your tileset, Drag and Drop either a Prefab, a Game Object or a Mesh to the "Drag New Tile Object" square found in your Tileset. This will immediately create a new Tile, using the Game Object or Mesh you added.

Image - 24
(Add New Tile)

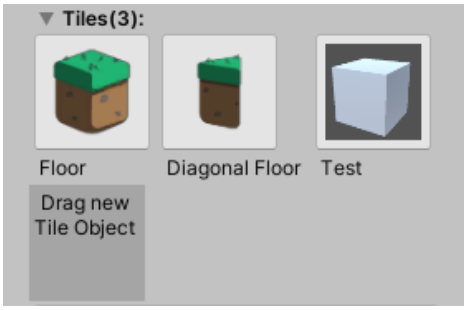


Image - 25
(Tile Added)

If you use either a Mesh or a GameObject found on the scene, a Prefab with either the Mesh or Object will be created in your Assets Folder.

This is done for the Mesh to be able to spawn it in the scene.

This is done for the Game Object so that if the Object in the Scene is destroyed, you don't lose the reference to the object.

Some examples of Tiles you could add to your Tileset are:

- Floor meshes created by you.
- Dressing objects for you scene such as: trees, rocks, crates, barrels, etc.
- Prefabs* with special behaviours.

*For more information on Prefabs, visit:
https://docs.unity3d.com/Manual/Prefabs.html

Save Tileset:

After Adding at least 1 new Tile with the **Add New Prefab** option, you will be able to save your current Tileset, allowing you to Load it at any point.

Once you click the "Save Tileset" button, you will be prompted to choose the location and name of your Tileset.

Its recommended to create a separate folder for your tilesets.

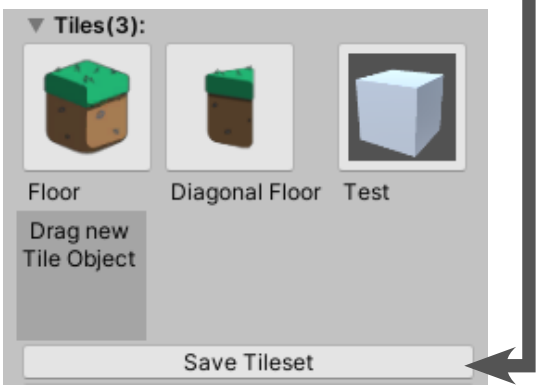


Image - 26
(Save Floor Button)

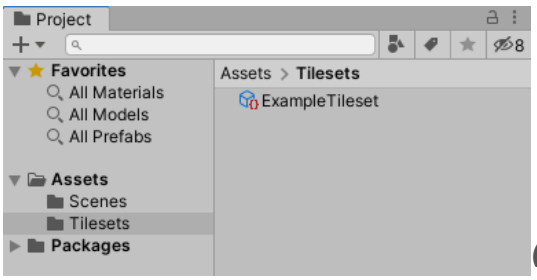


Image - 27
(Saved Tileset Example)

Load Tileset:

At any point you can change your current Tileset for any of your previously saved Tilesets. Be aware that when you load a Tileset the **current one will not be saved Automatically.**

When you click the "Load Tileset" button, you a window will appear asking you to choose the saved Tileset you wish to Load. If you select an Asset that is not a saved Tileset, it will not be loaded and your current Tileset will stay the same.

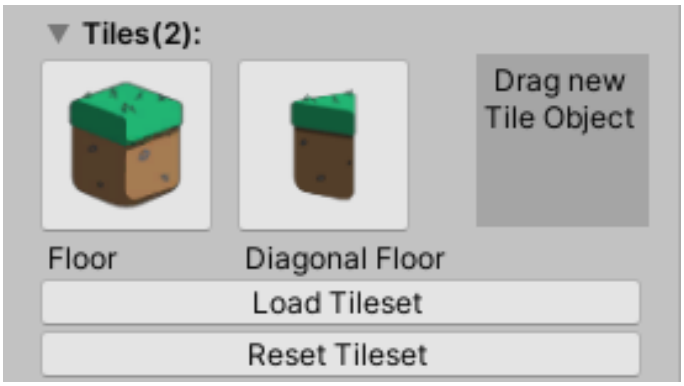


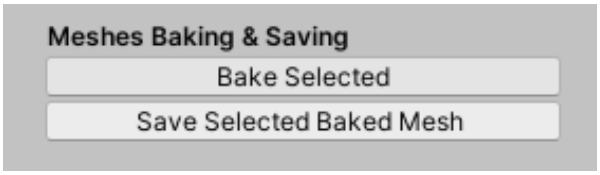
Image - 28
(Load Tileset)

Reset Tileset:

To reset your current Tileset to the Default Tiles, click the "Reset Tileset" button.
This will delete all your current tiles, without saving them, leaving only the Default Tiles.

Mesh Baking and Saving:

Image - 29
(Mesh Baking and Saving Area)



"Bake Selected":

As an added feature, you can bake the meshes of any selected object and any child objects, merging them into a single mesh. This can be used not only to save the levels you create as a mesh, but using this feature can also help you optimize your scene.

The objects you can "bake" are not restricted to the tiles you create. You can bake any selection of objects that you have in your scene.

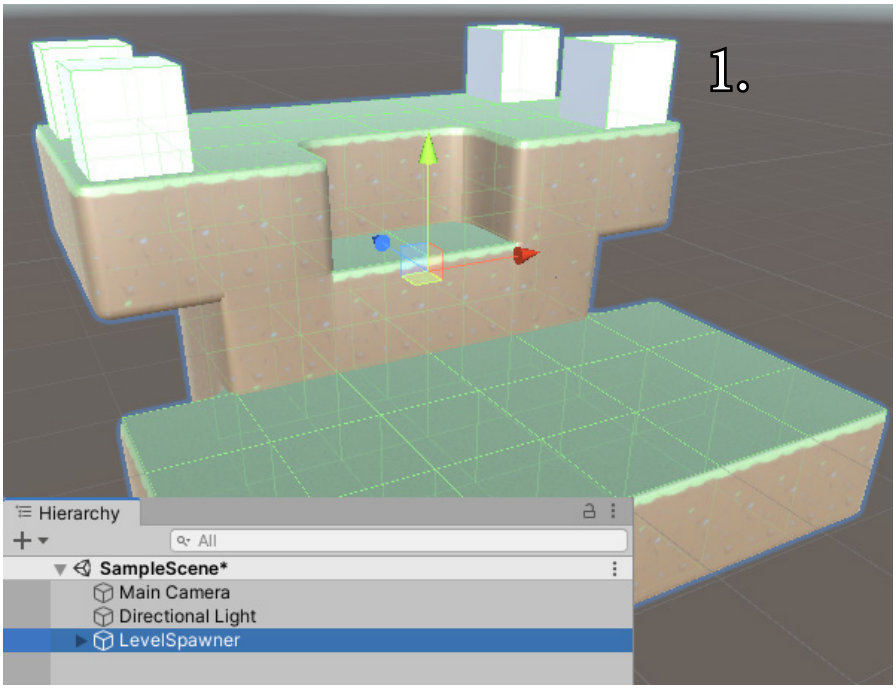
"Save Selected Baked Mesh":

Apart from being able to bake meshes, the 3D Tilemap package allows you to save in your project any mesh that you bake. To do this, select the Game Object created by the Bake Selected function, and press the "Save Selected Baked Mesh" Button. This will open a window for you to choose where you want to save your mesh and its name.

Notes:

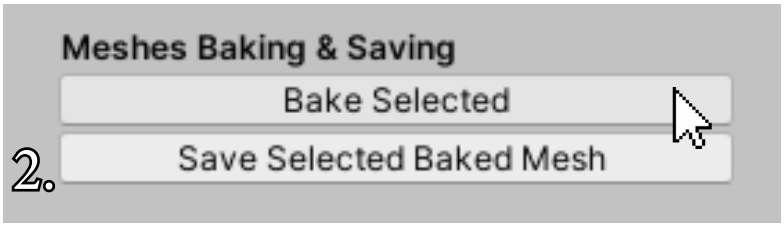
- This button can only save meshes that you have baked.
- The meshes that you save are NOT FBX or OBJ, they are unity .asset meshes. They will work inside of the engine, but if you want to export the you can use the FBX Exporter Unity Package. For more information in this, look at the Advanced Options at the end of the Manual.
- If you bake and save a mesh with more than 1 different Material, when you place the saved mesh on a scene you will have to manually set the Materials to the correct quantity and set every material in the order they appear in the baked object.

Example:

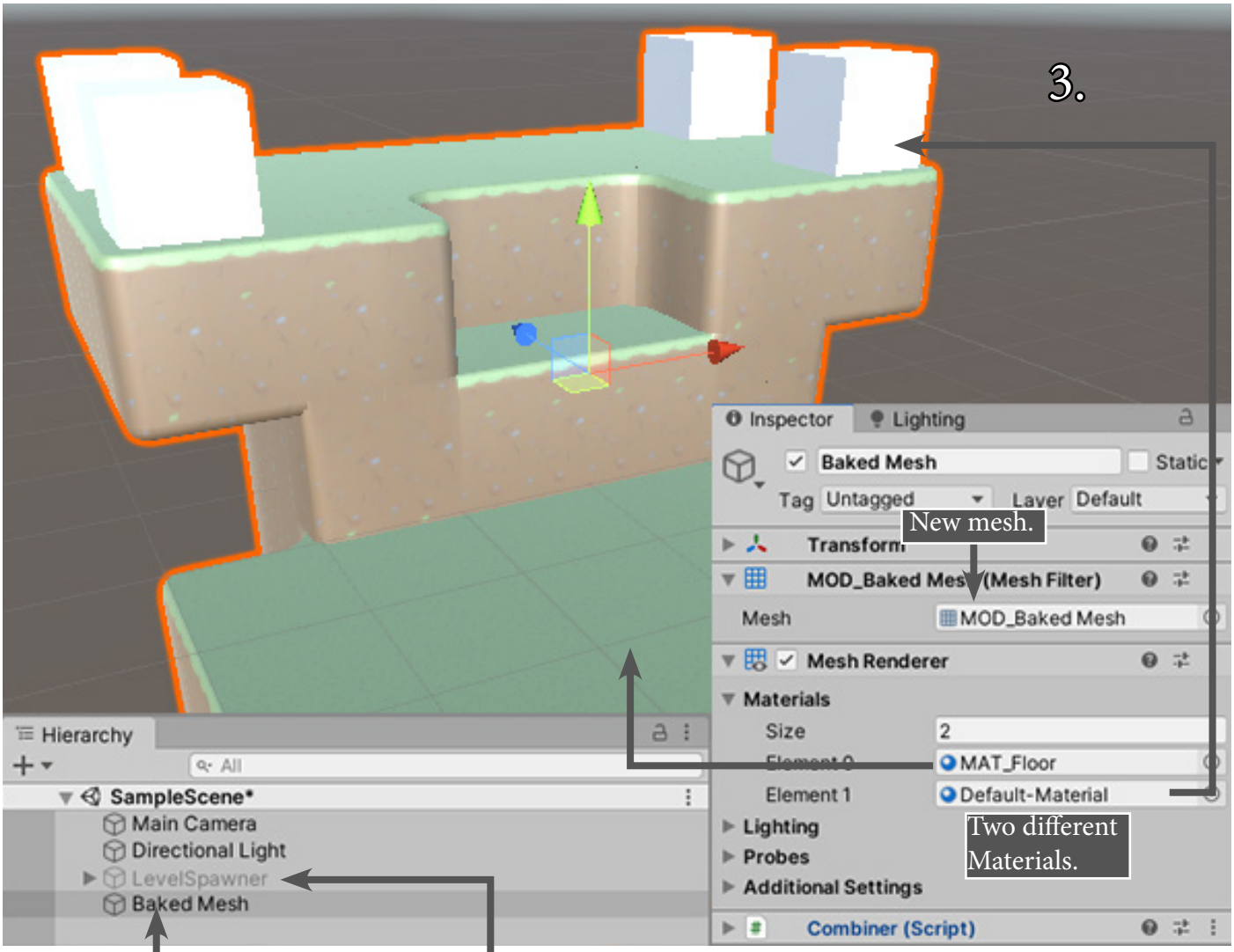


1. Create a level and select the "Level Spawner" Game Object in the Hierarchy.

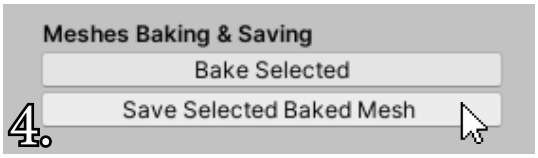
This will allow you to bake all the floors into one mesh.



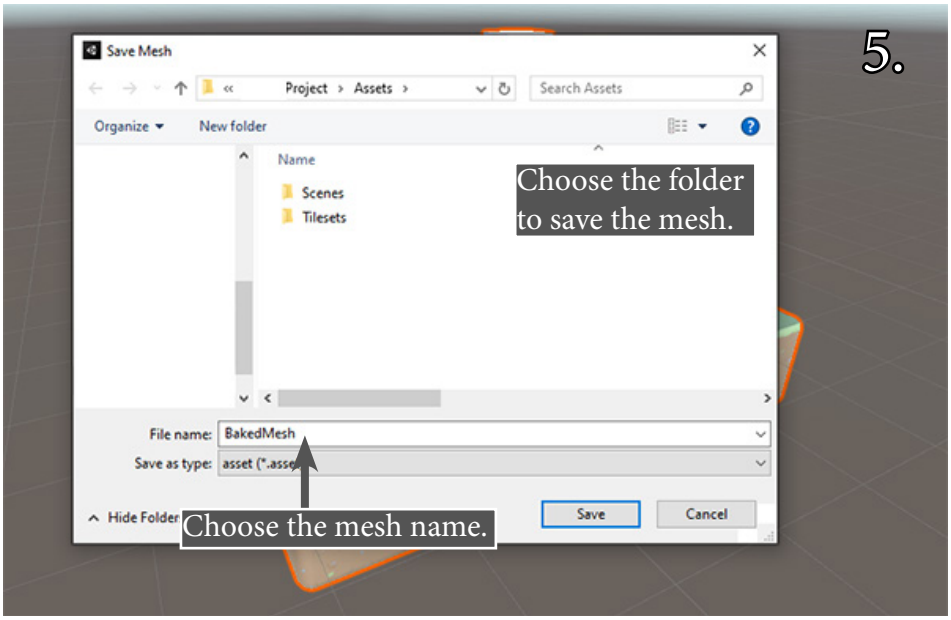
2. Click Bake Selected.



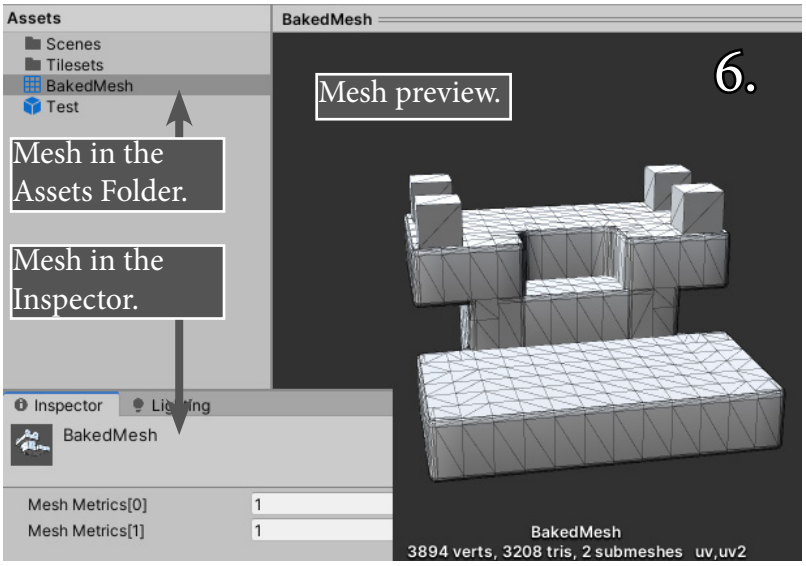
3. A new object is created with the baked mesh. And the objects we baked are set as inactive.



4. While the newly created object is still selected, click “Save Selected Baked Mesh”.



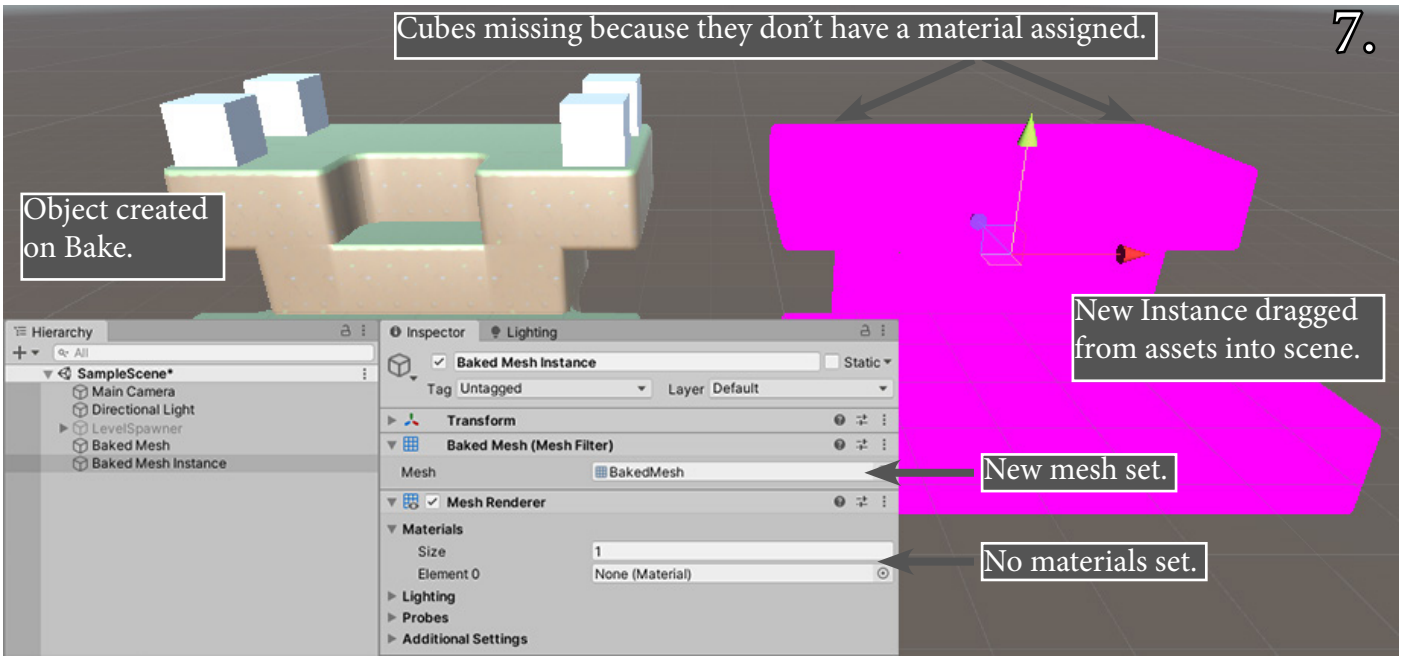
5. A window will pop-up. Choose the folder and name to save the new mesh, click Save.

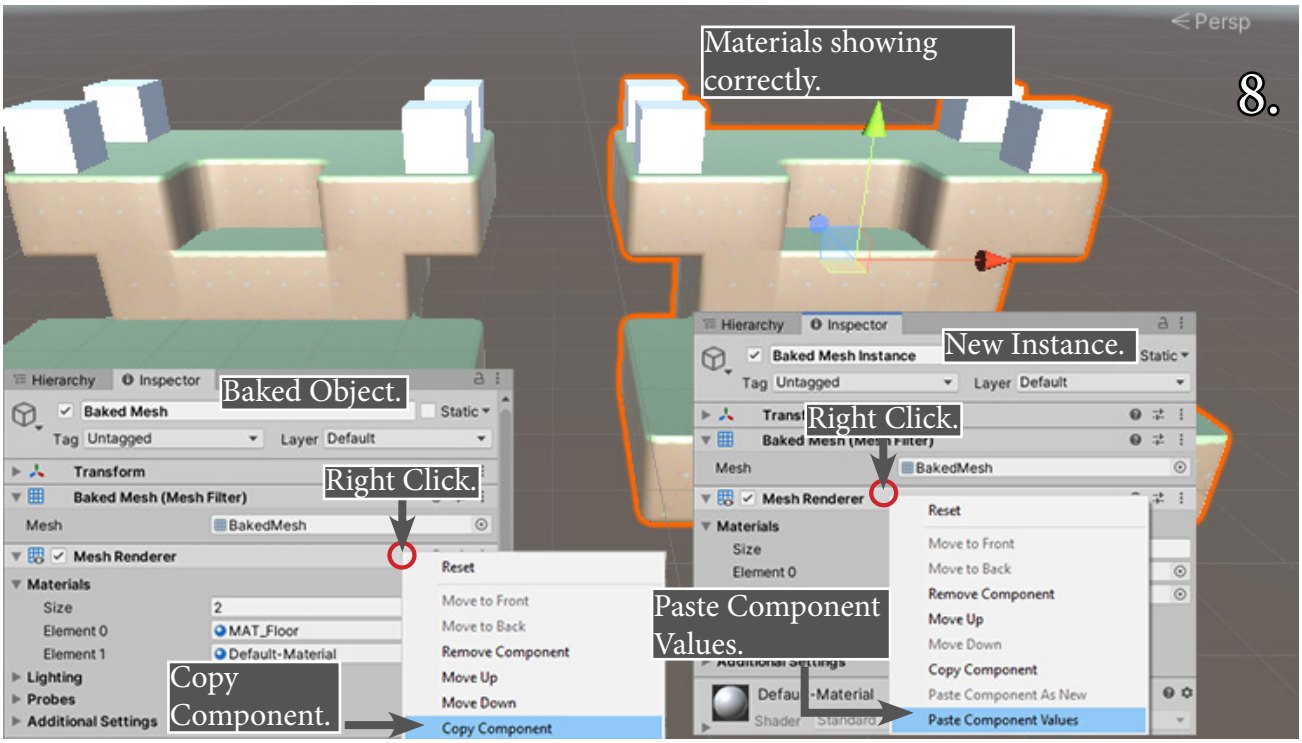


6. After saving, the mesh can be found in your assets, in the folder you chose to save it.

7. Drag and drop mesh into the scene to create a new Instance. It will not have materials assigned.

8. Copy the Mesh Renderer component from the baked Object, and paste the values on the new Instance. OR select and set the materials manually.
Now you're DONE!





Advanced Options:

The advanced options described below, are processes or options that will require a better understanding of both Unity as well as the 3D Tilemap package. Using them will allow you to personalize even further your project and take more advantage of the package.

Custom "Floor Settings":

To change the look of the "Self Tiling Floor" you can create a custom material, changing the appearance of the floor, so you can adapt it to different biomes.

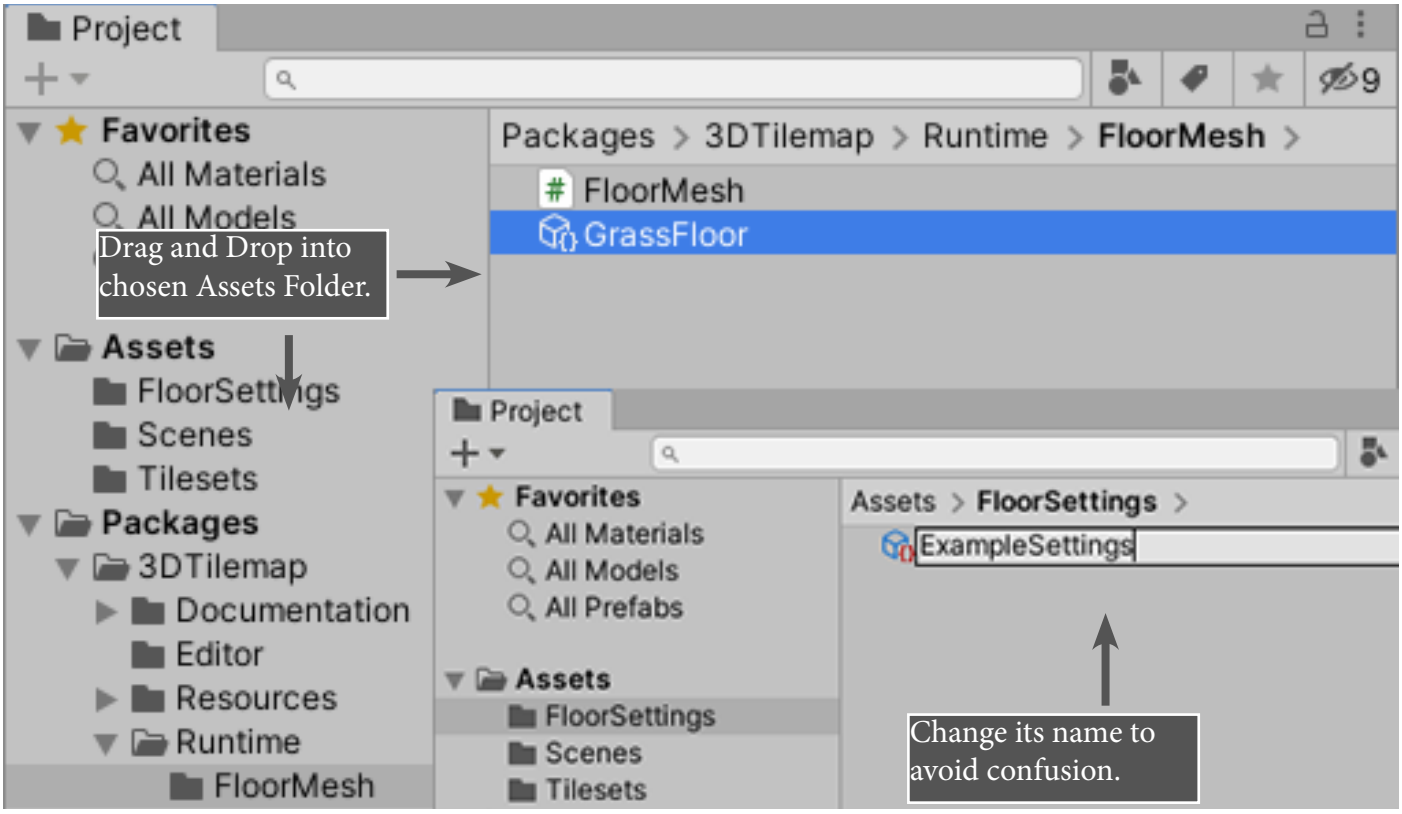
Create Custom Material :*

***Requires Knowledge in texturing.**

If you create a Custom Material for the "Self Tiling Floor" you will be able to slightly change how it looks. You may not want to have grass floor, you could make the floor look like rocks, or sand. This is the easiest way to change the appearance of the "Self Tiling Floor".

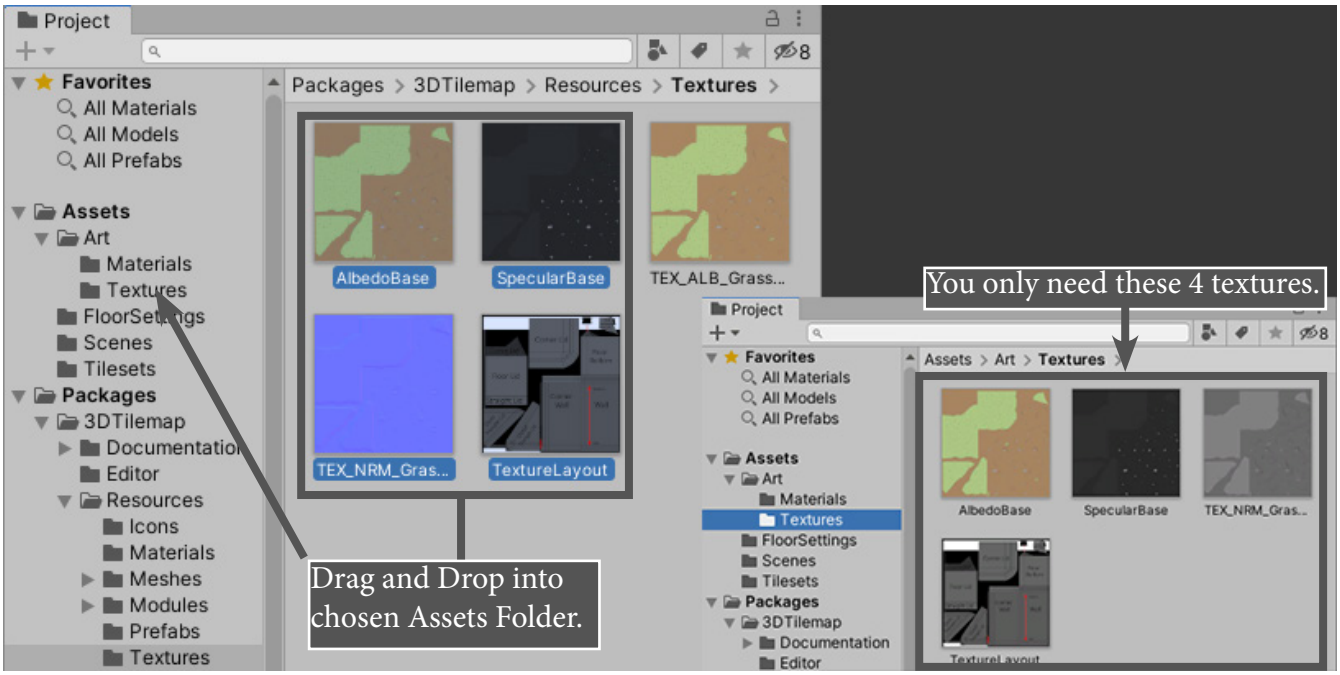
To create a Custom Material you can do one of two things, you can either create a whole new material with the Texture Layout found at: ***Packages/3DTilemap/Resources/Textures/***, or you can copy and edit the included textures for the floor in the same folder.

The 1st step in both options is to copy the included "Floor Settings" called "GrassFloor" into your assets, and change its name.



Edit Included Material:

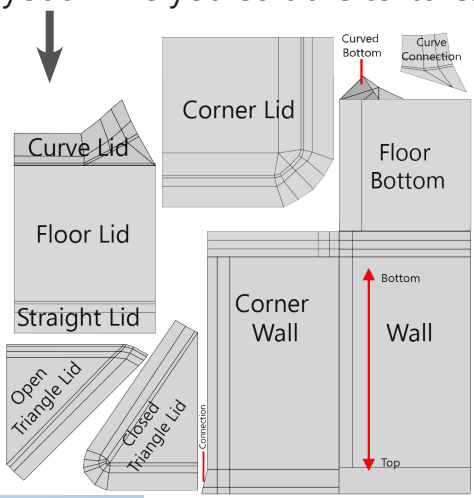
1. Get all the textures included in the "/Packages/3DTilemap/Textures/" folder, these will include: **"AlbedoBase"**, **"SpecularBase"**, **"TEX_NRM_GrassFloor",** and **"TextureLayout"**. Copy them to a folder just as the "Floor Settings" on the previous step.

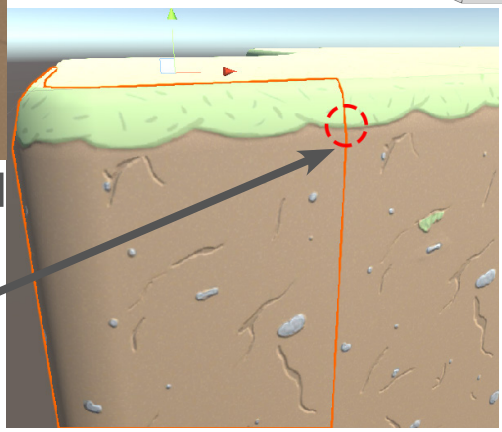


2. After copying the textures, you can start editing them on your preferred editing software (i.e. Photoshop or Gimp). Here you can change the look of the textures in any way you want, while sticking to the UV's shown in the "Texture Layout Image.

Tips:

- Place and lock on the uppermost layer the Texture Layout while you edit the texture.
- Try to keep the parts on the edges of the UV's at a similar place, so that they connect correctly in-between them.





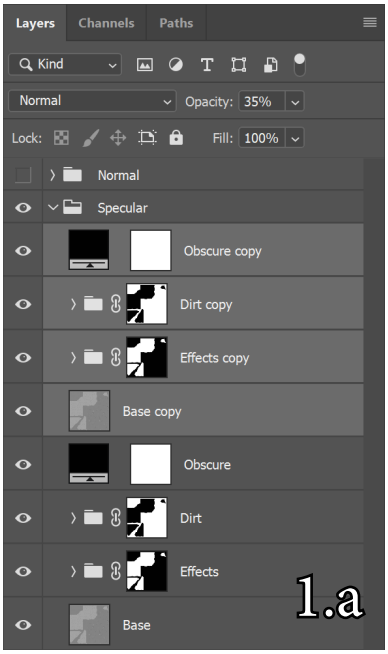
This will help the edges match on the self tiling floor.

If you want to better understand the Layout for the textures, check the "Matching Guide".

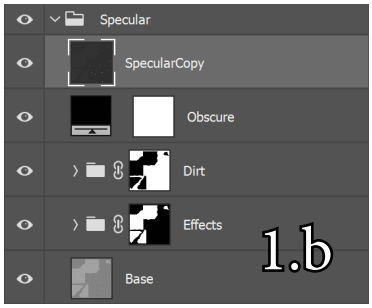
3. Now you can start editing the texture to fit your vision better. You can change the colors, add or substract details. Remember to update both the specular map and the normal map with the changes you make.

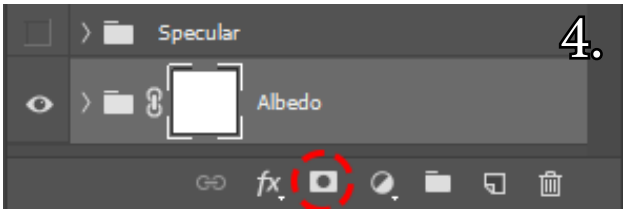
The normal map is made as a grey-scale, 50% is the base height. Black means down and white means up.

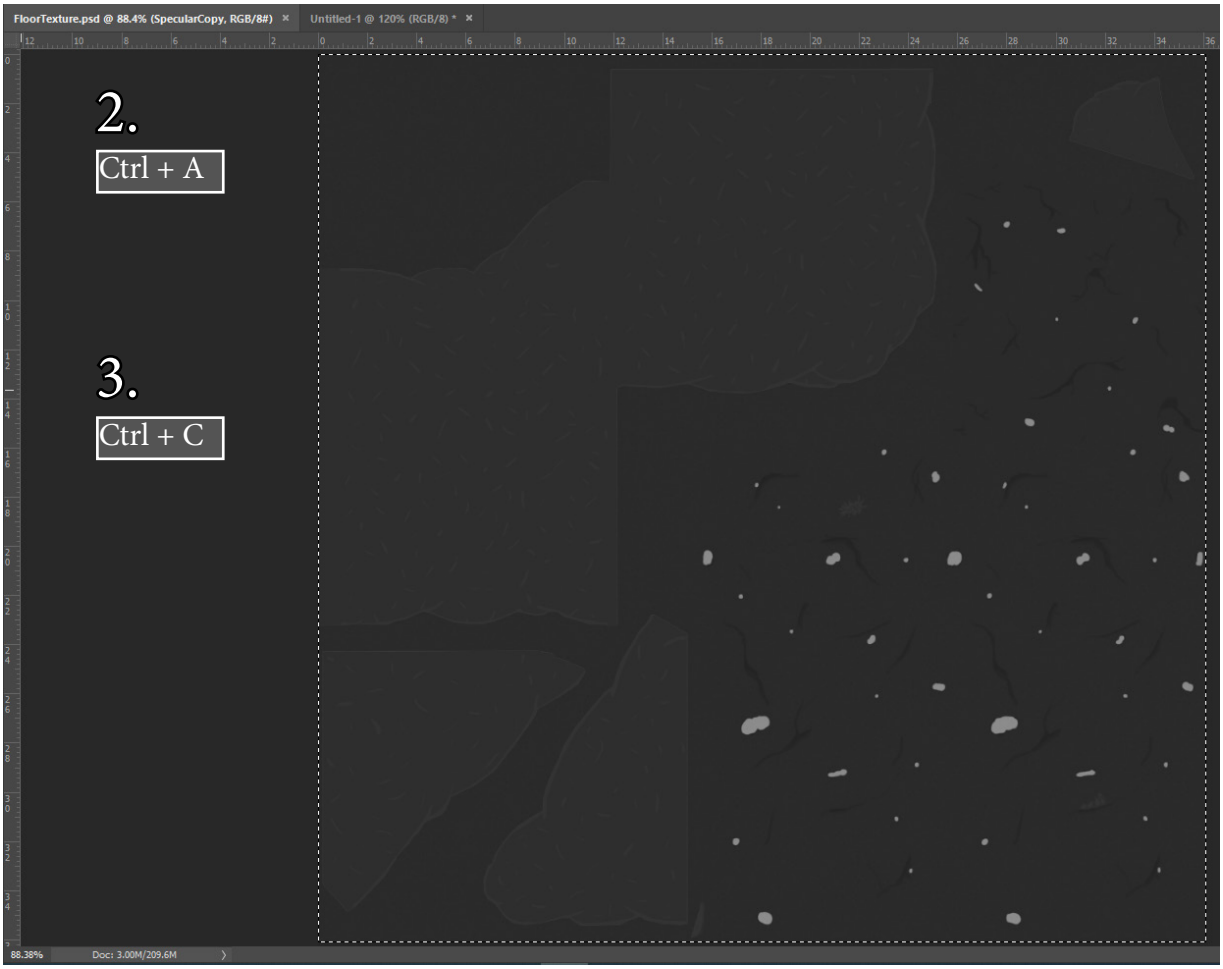
3. Once you finish editing the textures, you have to set up the Specular map as the Albedo Alpha. To do this in Photoshop:

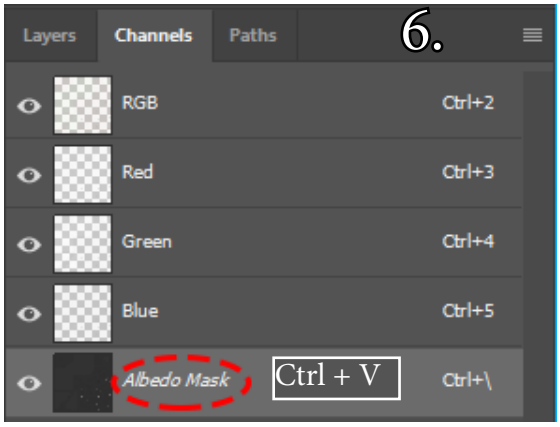


1. Select the layers with the Specular map and duplicate them, after duplicating them, combine them into one (Ctrl + E).
2. Select the combined layer and select all (Ctrl + A).
3. Copy the selection (Ctrl + C).
4. Create a mask for your Albedo texture.
5. Switch from layer to Channels while the Albedo is selected.
6. Select the previously created mask, set it as visible and paste the Specular map (Ctrl + V).
7. Go back to Layers, and export your Albedo as a PNG.









Once you paste the specular mask, the image will turn slightly red and mostly transparent, that is normal.

Import Settings:

4. Once you have all of your textures, import them back to Unity.

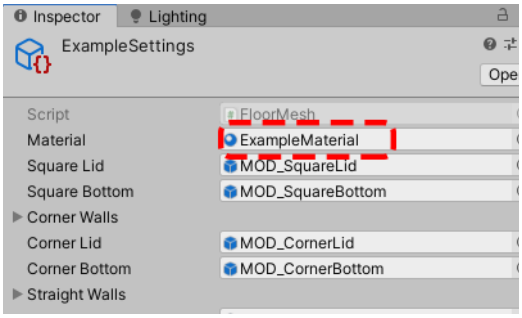
5. Set the Albedo's "Aniso Level" as 5.

6. Set the Normal Map properties:

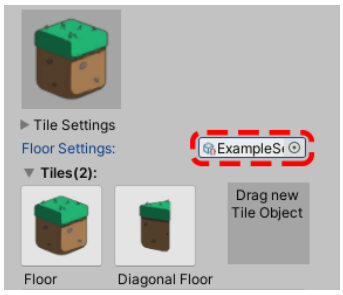
- Texture Type = Normal Map
- Create from Gray-scale = true
- Bumpiness = 0.044
- Filtering = Smooth

7. Create a new Material and set the textures in it. Use the Albedo Alpha as the Source of Smoothness.

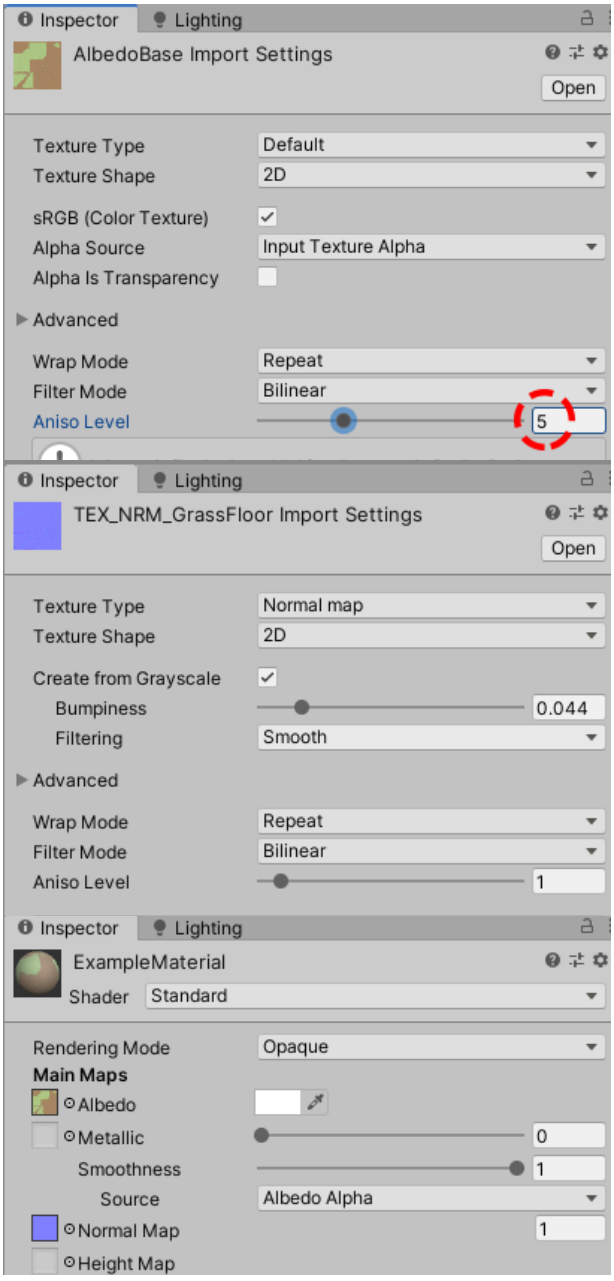
8. Place the new Material in the "Floor Settings" that you copied.



9. Place your custom settings on the field above the Tileset.

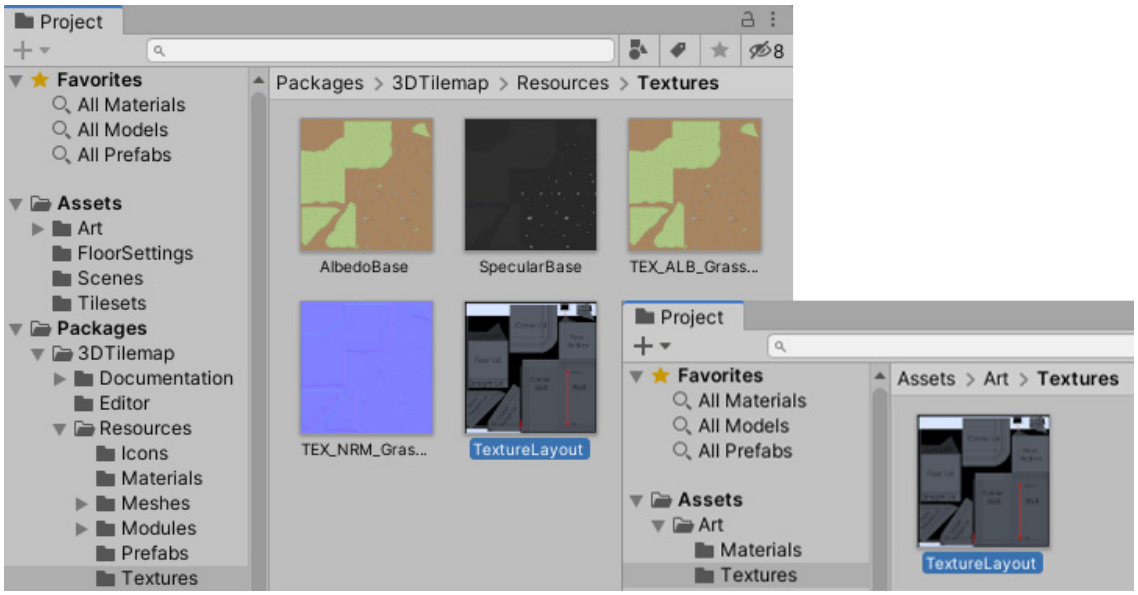


Now you can use your custom Material on the Self Tiling Floor!



Create New Material:

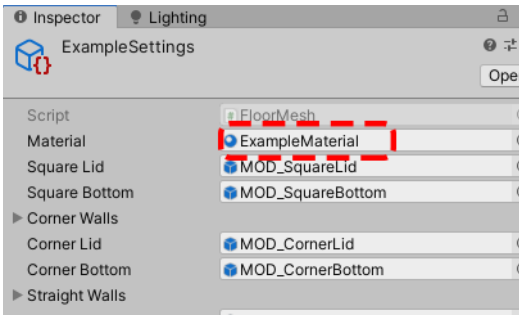
If you want to create a whole new texture for the "Self Tiling Floor" you will only need to get the Texture Layout from /Packages/Resources/Textures/. This image includes the UV Layout for the meshes used by the "Self Tiling Floor", so that you can create your own custom Textures based on your needs.



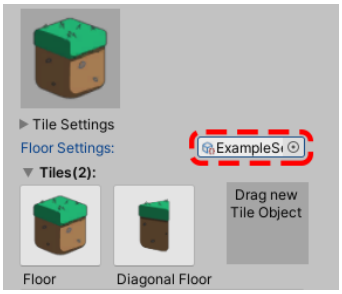
After getting the "TextureLayout" create the textures you want, sticking to how the UVs are laid out. You can use the "Matching Guide" included in the manual to better understand the UVs.

Once you finish creating all the textures (i.e. Albedo, Specular/Roughness, Metallic, Normal, Emissive) you need for your Custom Material, import them into Unity and create a new Material using them.

When you finish setting up your material, place it on the "Floor Settings" that you duplicated.

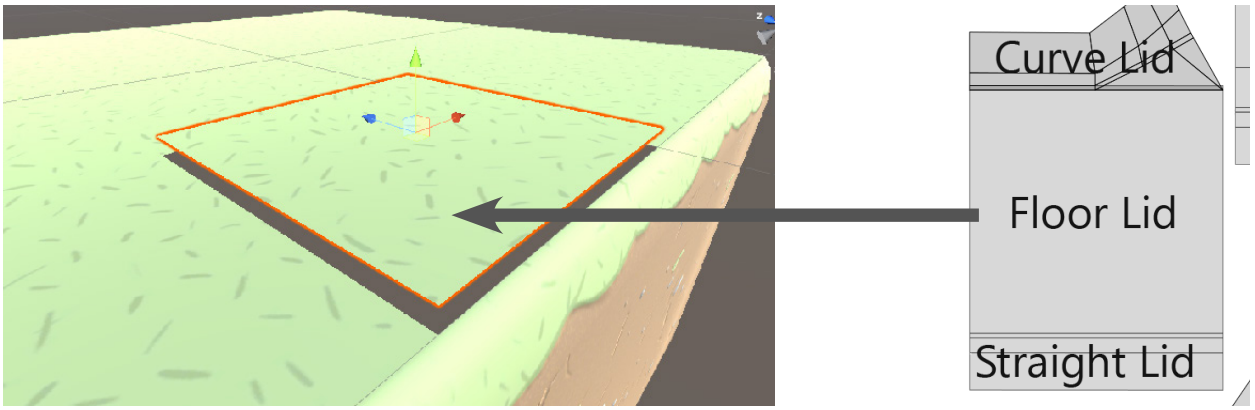


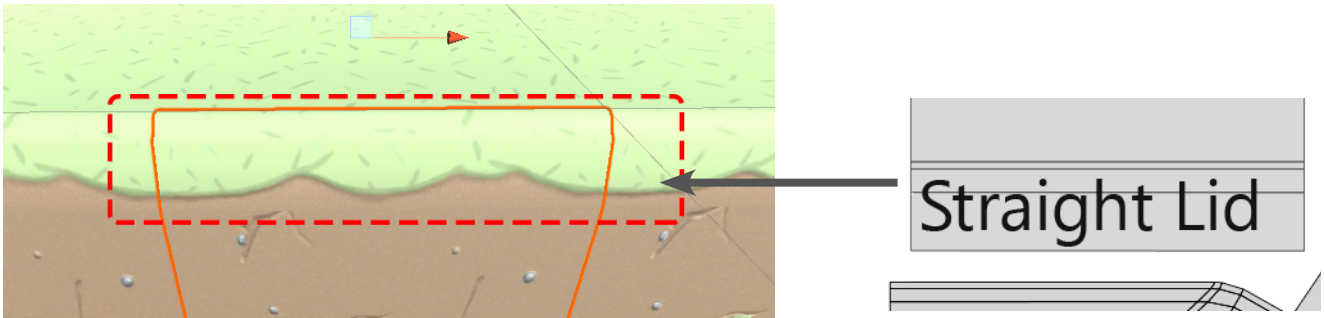
Lastly, place your Custom "Floor Settings" in the Field above your Tileset.

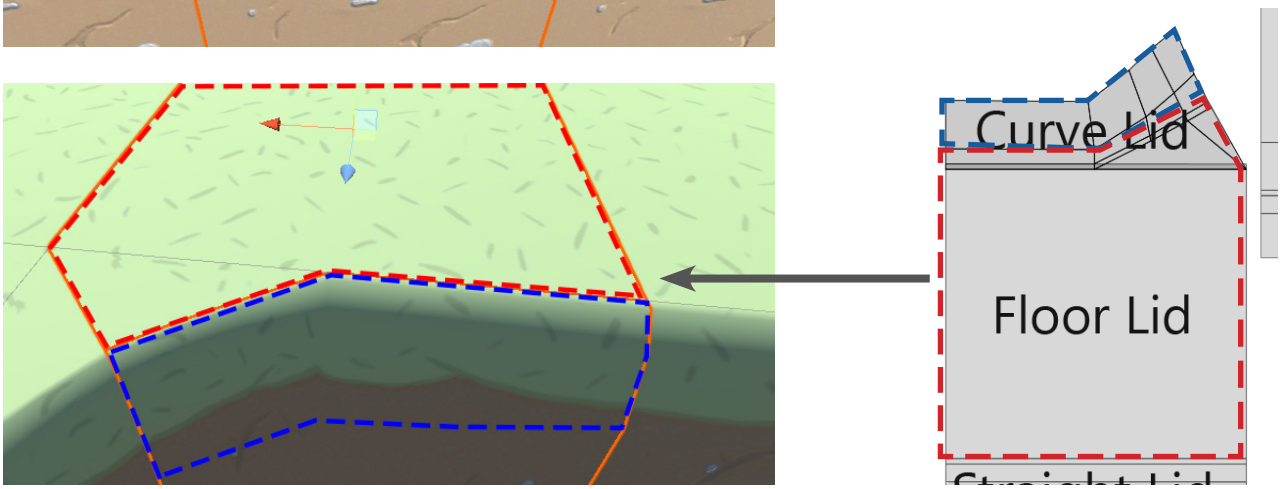


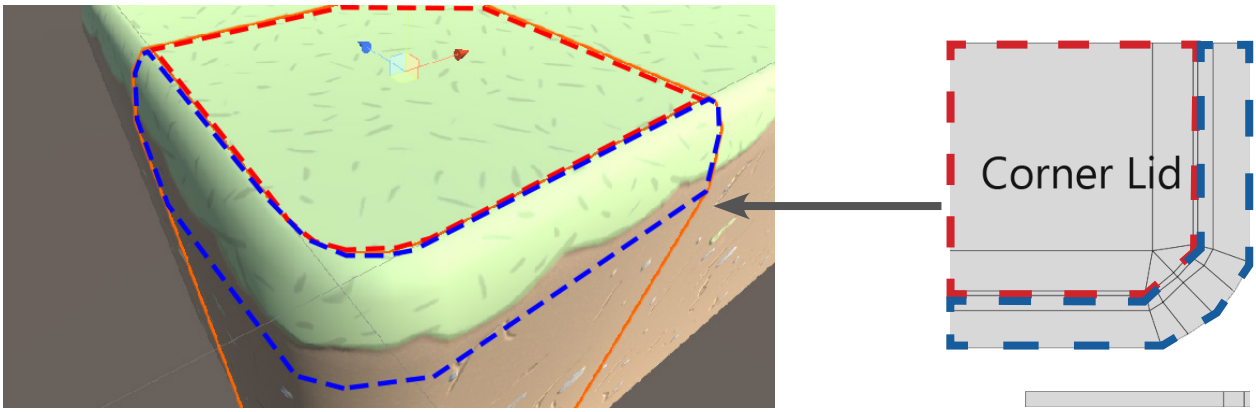
Now you can use your custom Material on the Self Tiling Floor!

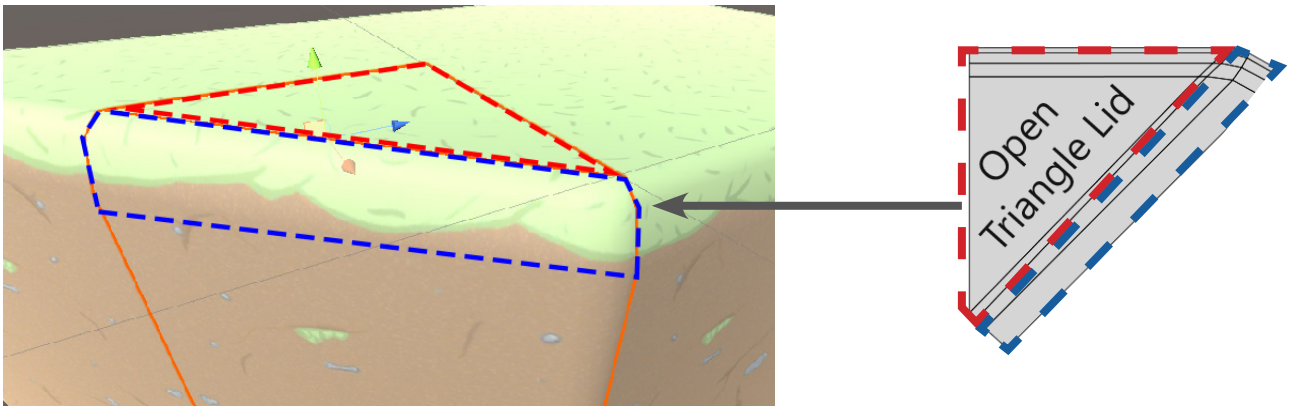
Matching Guide:

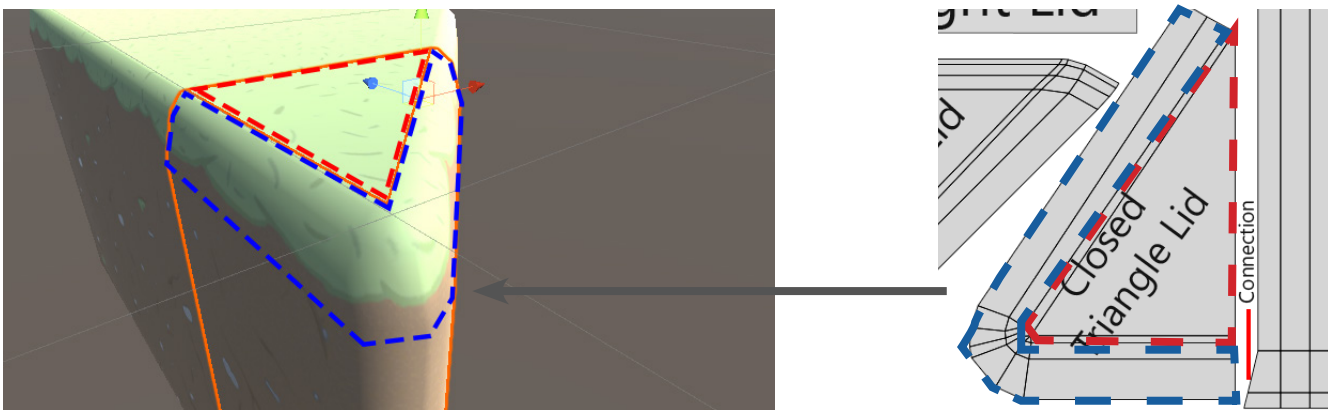


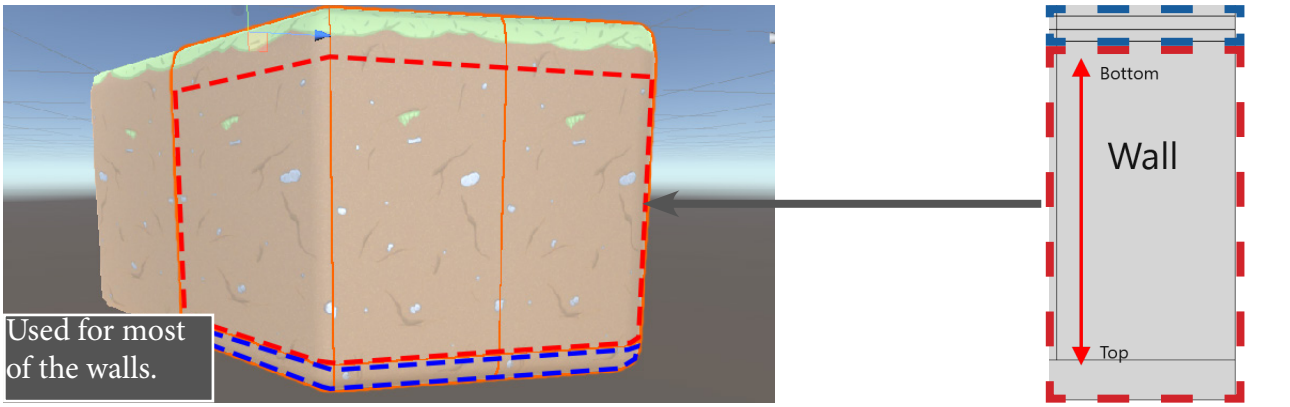


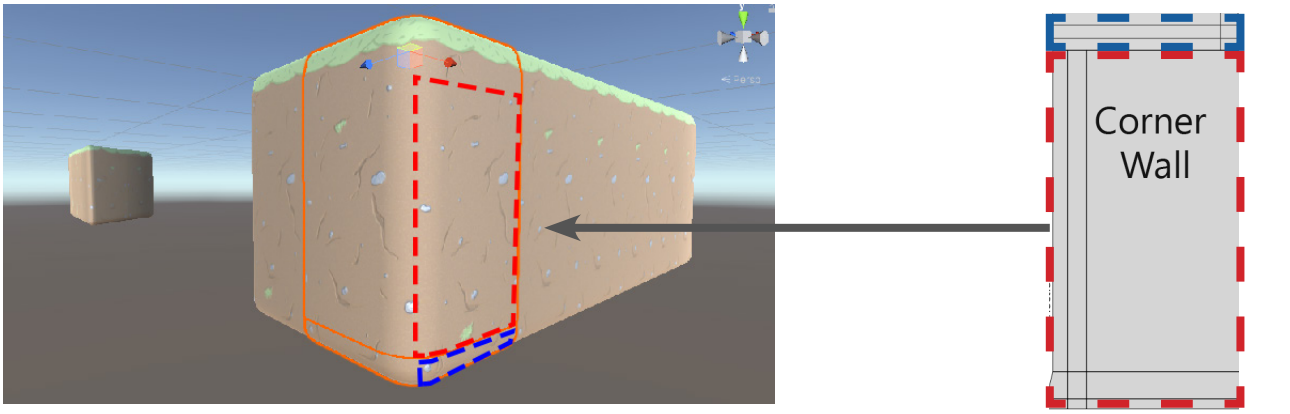


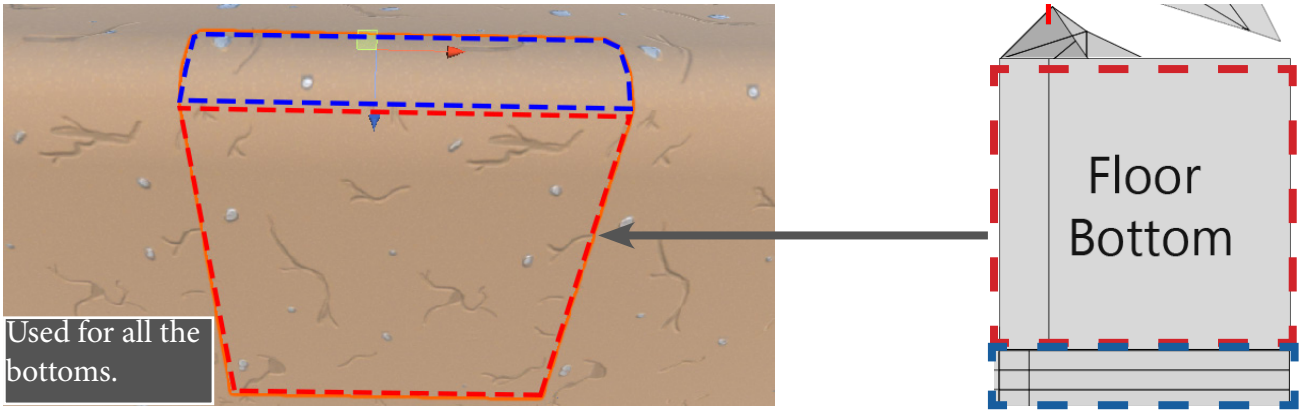


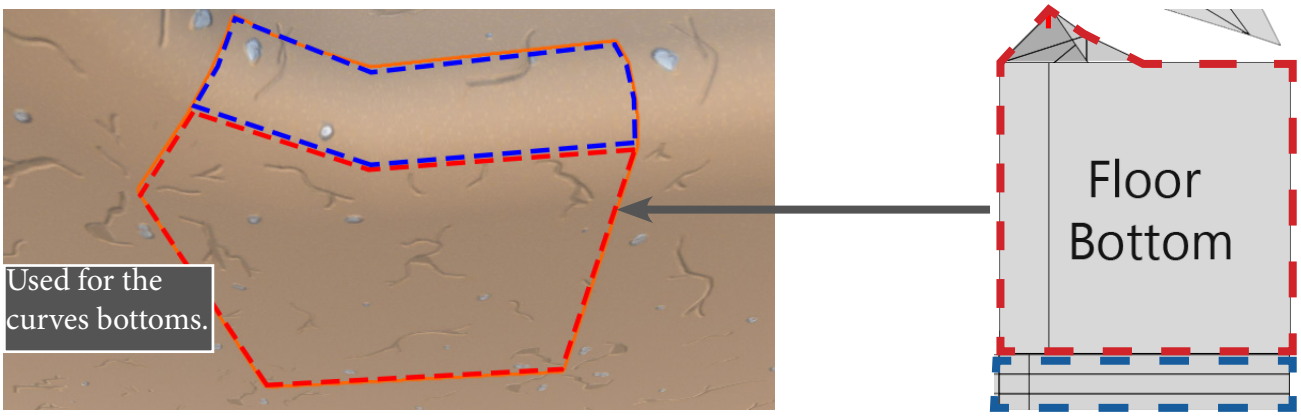

Open
Triangle Lid


Closed
Triangle Lid
Connection

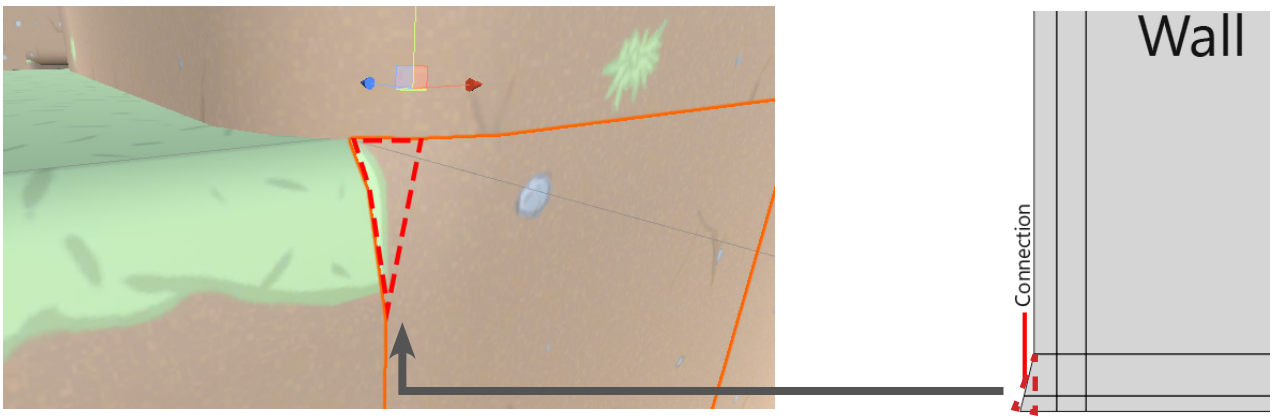

Used for most
of the walls.
Bottom
Wall
Top


Corner
Wall


Used for all the bottoms.
Floor Bottom


Used for the curves bottoms.
Floor Bottom

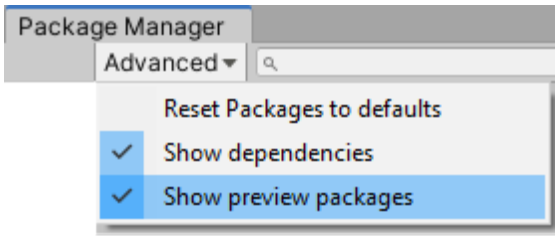

Curve Connection


Wall
Connection

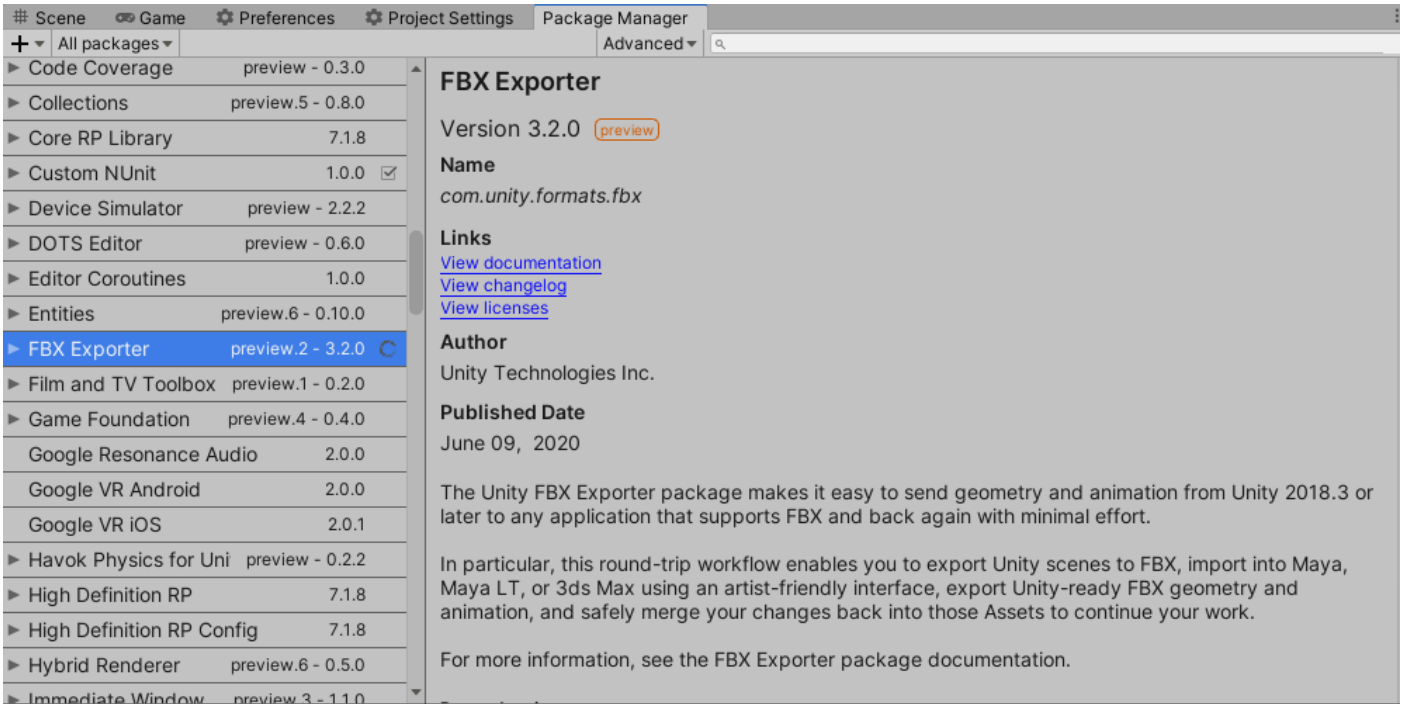
Export "Baked Mesh":

If you want to export one of the meshes that you have baked and saved, you can do so by downloading the Unity FBX Exporter Package from the Package Manager. Whit it you will be able to convert your baked meshes into fbx files allowing you to easily export them and edit them in any 3D modelling software.

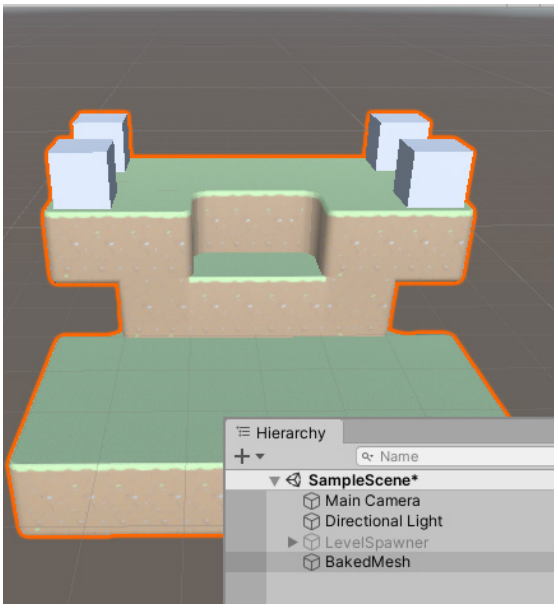
1. If you don't have the FBX Exporter Package installed, go to the Package Manager window and download it.



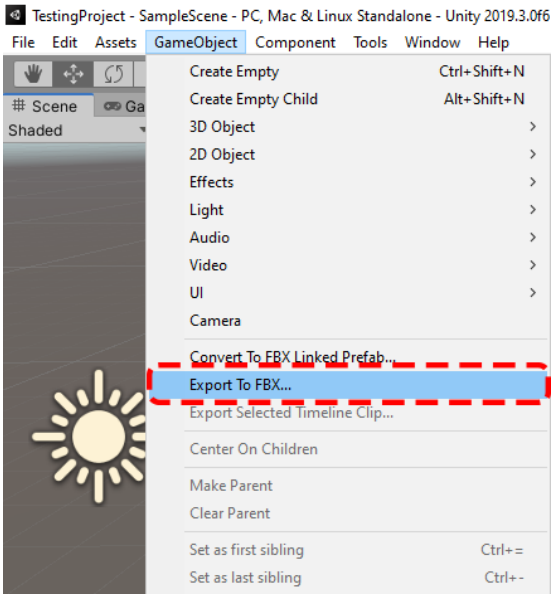
Make sure you have enabled the "Show Preview Packages" Option.



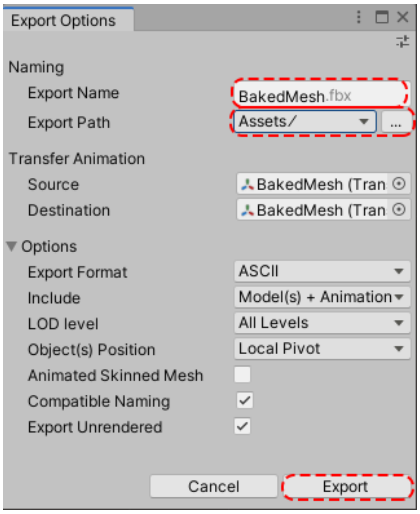
2. Once the FBX Exporter package is installed, place in your scene the mesh you want to export and select it.



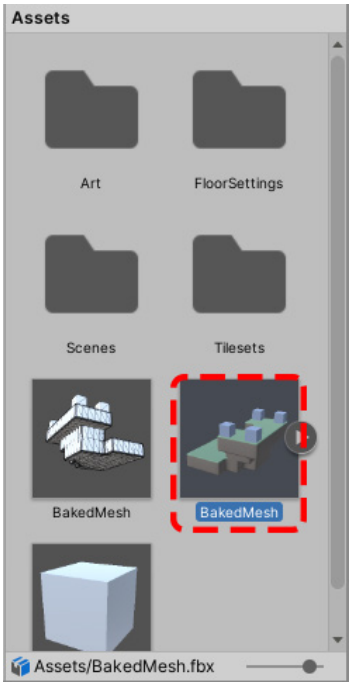
3. Click on the GameObject Menu and select “Export to FBX”.



4. A new window will appear, select the name you want for the Mesh and the path to save it at, then press “Export”.



5. The FBX will appear on the path you chose.



Now you can export it from Unity and edit it in any 3D Modelling Software!