UniTiled 3D Tutorial

How to create 3D maps in Unity using Tiled

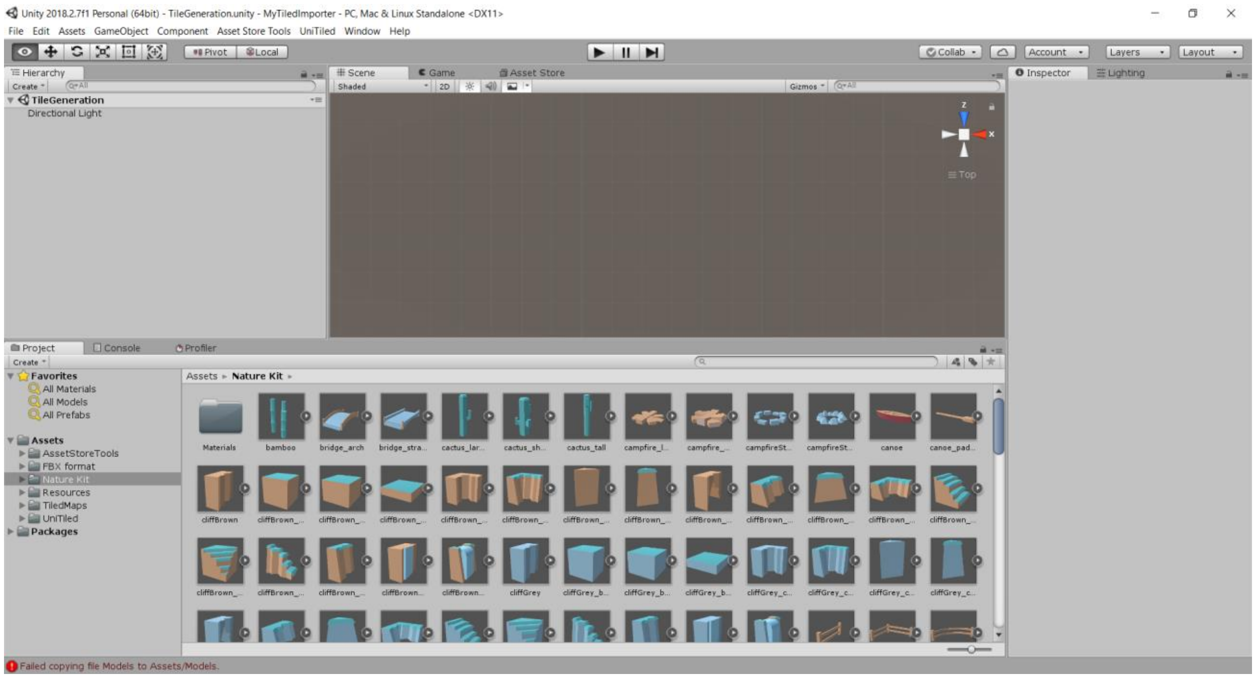
This is a step-by-step minimal tutorial on how to use 3D tiles with Unity and Tiled to create 3D maps. For further questions please do not hesitate to contact me directly by commenting on UniTiled page on itch.io (link below) or directly by email aljaffal [at] gmail.com

Before starting this tutorial, make sure you have UniTiled latest version added to your Unity project. You can download UniTiled unity package from https://yjaffal.itch.io/unitiled

Step1: Preparing 3D tiles

In order to build 3D maps, you need a set of 3D tiles added to your project. For this tutorial I am going to use Nature Kit from Kenney. You can download the package for free at https://kenney.nl/assets/nature-kit

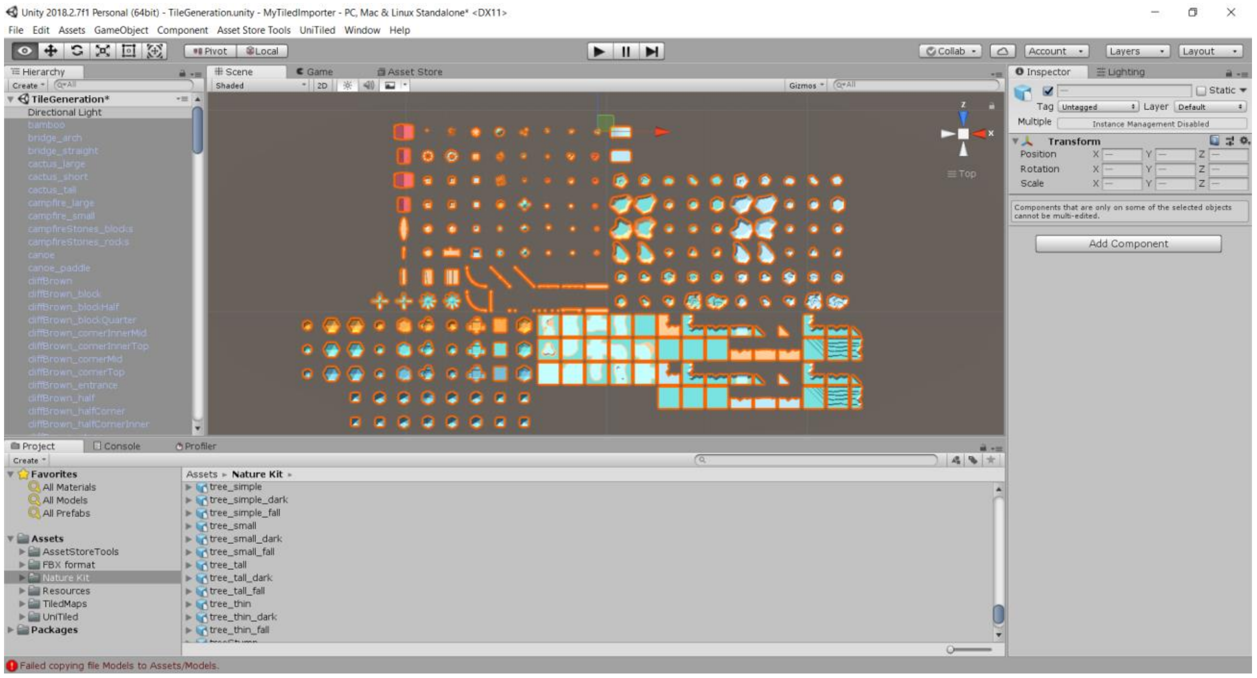
Download the kit and add its models to your Unity project assets, then create a new empty scene and save it as “TileGeneration” or any other suitable name. It is advisable that you use this scene only for tile generation and keep you game scenes separate.


Unity 2018.2.7f1 Personal (64bit) - TileGeneration.unity - MyTiledImporter - PC, Mac & Linux Standalone <DX11>

Failed copying file Models to Assets/Models.

In order to be able to build out map in Tiled, we need 2D sprites. We will generate these tiles from the 3D tiles we already have. You can create as many tile sets as you wish. For this Tutorial I am going to use all 3D tiles in nature kit to build a single 2D tile set. To create a 2D tile set follow these steps:

1. Add all models you wish to use to the scene and select them all.



Unity 2018.2.7f1 Personal (64bit) - TileGeneration.unity - MyTiledImporter - PC, Mac & Linux Standalone* <DX11>
File Edit Assets GameObject Component Asset Store Tools UniTiled Window Help
Pivot
Local
Collab
Account
Layers
Layout
Hierarchy
Scene
Game
Asset Store
Shaded
2D
Gizmos
Inspector
Lighting
Top
Project
Console
Profiler
Add Component

TileGeneration*
Directional Light
bamboo
bridge_arch
bridge_straight
cactus_large
cactus_short
cactus_tall
campfire_large
campfire_small
campfireStones_blocks
campfireStones_rocks
canoe
canoe_paddle
cliffBrown
cliffBrown_block
cliffBrown_blockHalf
cliffBrown_blockQuarter
cliffBrown_cornerInnerMid
cliffBrown_cornerInnerTop
cliffBrown_cornerMid
cliffBrown_cornerTop
cliffBrown_entrance
cliffBrown_half
cliffBrown_halfCorner
cliffBrown_halfCornerInner

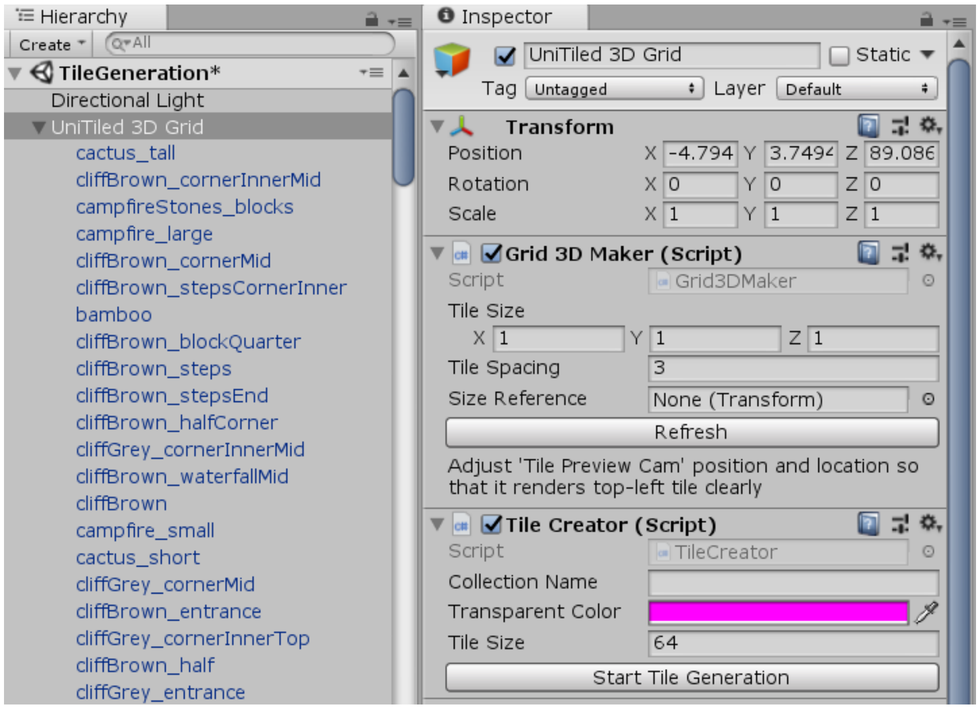
Tag Untagged Layer Default
Multiple Instance Management Disabled
Transform
Position X — Y — Z —
Rotation X — Y — Z —
Scale X — Y — Z —
Components that are only on some of the selected objects cannot be multi-edited.

Favorites
All Materials
All Models
All Prefabs
Assets
AssetStoreTools
FBX format
Nature Kit
Resources
TiledMaps
UniTiled
Packages

Assets ▸ Nature Kit ▸
tree_simple
tree_simple_dark
tree_simple_fall
tree_small
tree_small_dark
tree_small_fall
tree_tall
tree_tall_dark
tree_tall_fall
tree_thin
tree_thin_dark
tree_thin_fall

Failed copying file Models to Assets/Models.

2. With all your tiles selected, go to UniTiled menu and select "Create 3D Grid". This will add a new object to the scene called "UniTiled 3D Grid" and set it as parent to all of your models. This object has two scripts attached: "Grid 3D Maker" and "Tile Creator"



TileGeneration*
Directional Light
UniTiled 3D Grid
cactus_tall
cliffBrown_cornerInnerMid
campfireStones_blocks
campfire_large
cliffBrown_cornerMid
cliffBrown_stepsCornerInner
bamboo
cliffBrown_blockQuarter
cliffBrown_steps
cliffBrown_stepsEnd
cliffBrown_halfCorner
cliffGrey_cornerInnerMid
cliffBrown_waterfallMid
cliffBrown
campfire_small
cactus_short
cliffGrey_cornerMid
cliffBrown_entrance
cliffGrey_cornerInnerTop
cliffBrown_half
cliffGrey_entrance

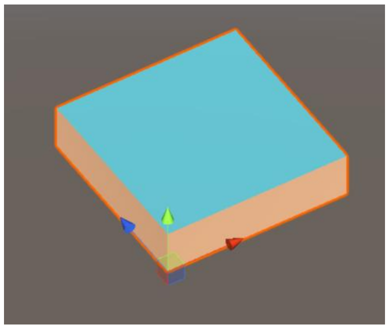
UniTiled 3D Grid
Tag Untagged Layer Default

Transform
Position X -4.794 Y 3.7494 Z 89.086
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

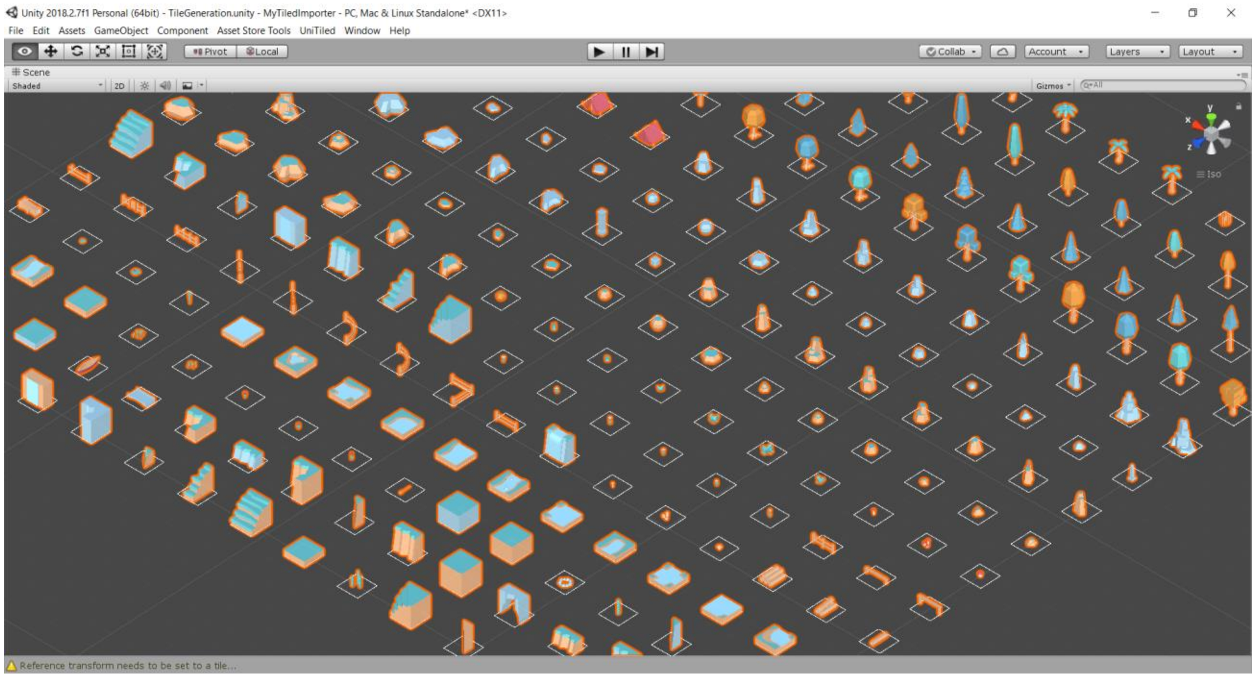
Grid 3D Maker (Script)
Script Grid3DMaker
Tile Size X 1 Y 1 Z 1
Tile Spacing 3
Size Reference None (Transform)
Adjust 'Tile Preview Cam' position and location so that it renders top-left tile clearly

Tile Creator (Script)
Script TileCreator
Collection Name
Transparent Color
Tile Size 64

3. Since generated 2D tiles will have the same size, you need a tile with suitable width and depth (XZ scale) to use as reference. This can be typically any complete ground tile. For this tutorial I chose **ground_dirt**. Drag this object to **Size Reference** field in **Grid 3D Maker**

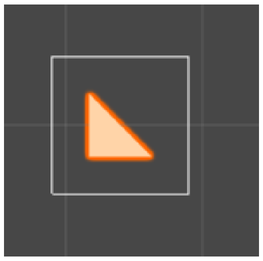


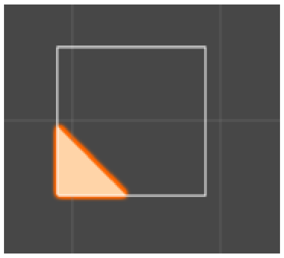
4. To arrange your tiles in a grid, set appropriate **tile spacing** in **Grid 3D Maker** and click **Refresh**. Try to keep tiles away from each other as possible. If they are too close, try increasing **tile spacing** value. For this tutorial I use 25. You may also make your tiles larger or smaller by setting **Tile Size** values. Your grid will now look like this


Unity 2018.2.7f1 Personal (64bit) - TileGeneration.unity - MyTiledImporter - PC, Mac & Linux Standalone* <DX11>
File Edit Assets GameObject Component Asset Store Tools UniTiled Window Help
Pivot
Local
Collab
Account
Layers
Layout
Scene
Shaded
2D
Gizmos
All
Iso

Reference transform needs to be set to a tile...

5. By default, all tiles are aligned to center. In some cases, we need to align tiles to other ends. For example, a corner tile like **cliffBrown_cornerMid** needs to be aligned to lower-left. Each tile has a script named **Tile 3D** attached to it. This script enables us to change alignment. Select the previously mentioned tile and change its alignment to **Lower Left**. Go back to **UniTiled 3D Grid** and click refresh to see the effect.

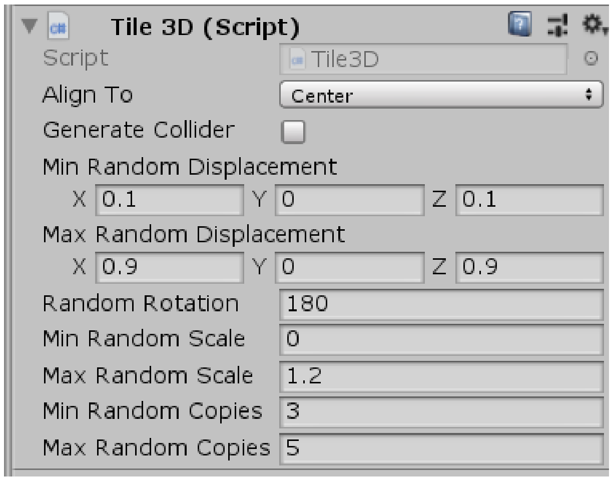

Center


Lowe left

6. You need to go through all tiles and make sure they have appropriate alignment. For instance, **cliffBrown_half** needs bottom alignment and so on.

7. Some tiles need additional processing to be suitable for 3D environments. For example, objects like **flower_red1** and other flowers are better to be repeated many times inside a tile, and these repetitions need to vary in size and position. To enable this, Tile 3D script offers the following properties, which are used to post-process each tile after constructing the 3D map:
    a. **Generate Collider**: whether this tile needs a collider to be attached. In the case of flower, for example, this should be disabled.
    b. **Min Random Displacement** and **Max Random Displacement**: set x, y, and z components of these vectors to values between 0 and 1 to define where the object should be randomly placed inside its tile space
    c. **Random Rotation**: the object will be randomly rotated around its y axis with a value between 0 and **random rotation**
    d. **Min Random Scale** and **Max Random Scale**: the tile object will be randomly scaled between these two values
    e. **Min Random Copies** and **Max Random Copies**: limits of count of random copies that will be created for the tile object.

8. Set **flower_red1** properties as shown below

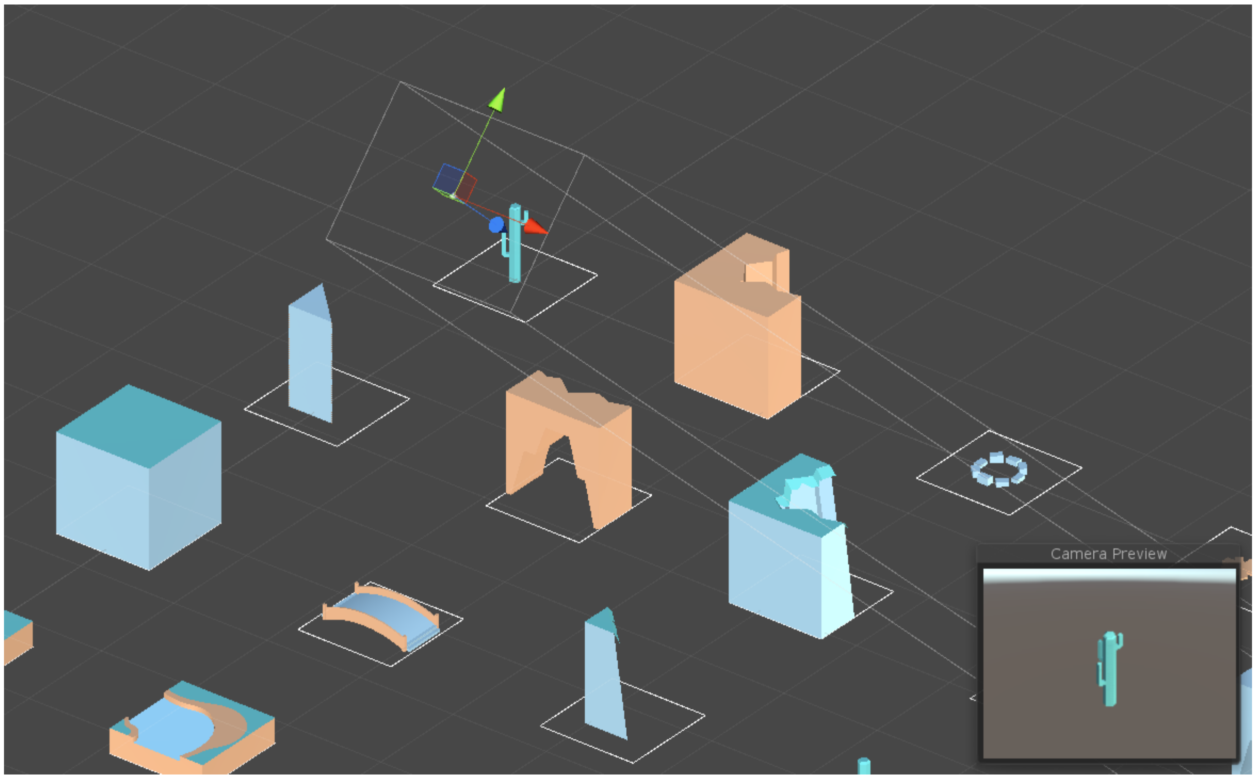

Tile 3D (Script)

Script	Tile3D
Align To	Center
Generate Collider	☐
Min Random Displacement	X 0.1 Y 0 Z 0.1
Max Random Displacement	X 0.9 Y 0 Z 0.9
Random Rotation	180
Min Random Scale	0
Max Random Scale	1.2
Min Random Copies	3
Max Random Copies	5

9. Do not forget to save your scene after you have finished setting properties for all your tiles!

Step 2: Creating 2D Tiles

Now you are ready to generate your 2D tiles from your 3D tile grid. Before going on, we need to setup a camera that will generate a 3D preview for our 2D tiles. This will help us recognizing our 3D tiles in Tiled, since many tiles will look similar from a top view. Fortunately, this camera is already created for you and added to the scene. The camera is called **Tile Preview Cam** and is positioned at the first tile in the grid. All you need to do is move, scale and rotate this camera a little so it shows a suitable preview of your first tile. Do not forget to count for tiles that have larger objects when customizing your camera.



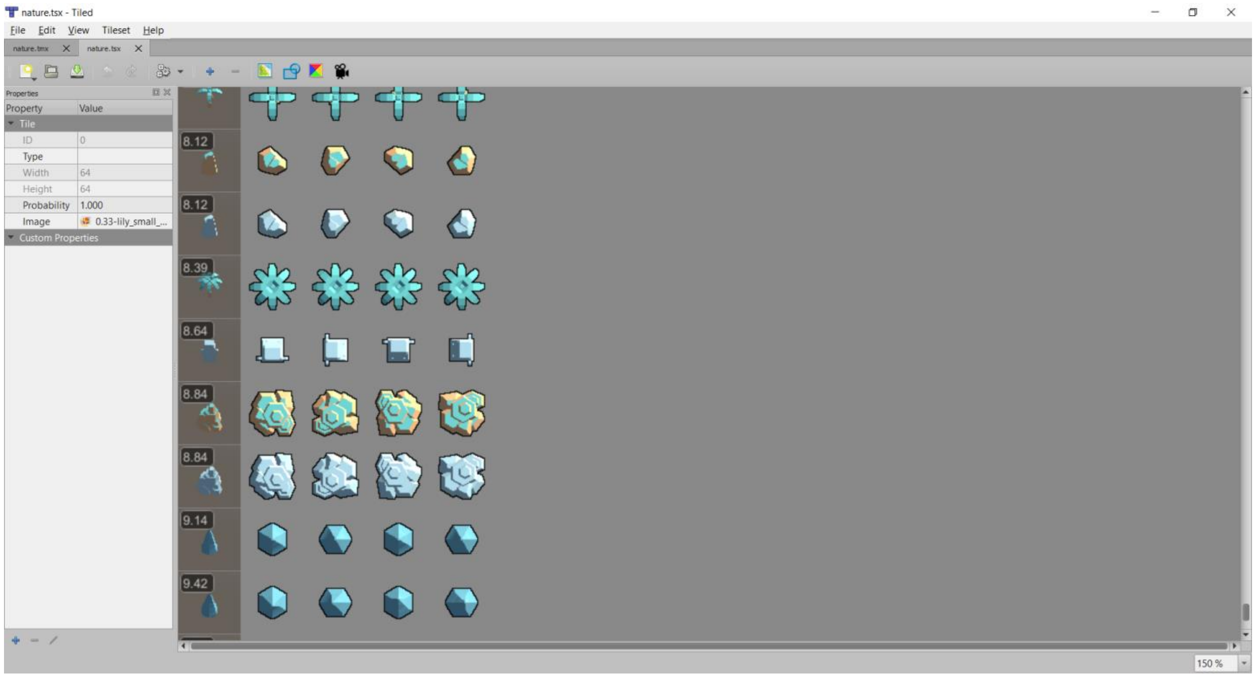
Once you have your tile preview camera ready, head to **Tile Creator** script on **UniTiled 3D Grid** object. First you have to set a name for your generated collection (**nature** for example) and specify the pixel size of your generated 2D sprites (64 for example). You might also change the transparent color, but it is only needed if the default pink color is used in your 3D tiles.

To start 2D tile generation, click **Start Tile Generation** button or simple enter play mode. Unity will now start taking screenshots and saving your tiles. The operation might take some time. When finished, Unity will automatically exit play mode and re-import newly generated 2D tiles. Following files and folders will be added to your Assets folder:

- Resources
  - **3DTilesAtlases** – contains a created prefab of your 3D grid
  - **Generated2DTiles** – contains generated 2D sprites inside a subfolder with collection name
- **TiledMaps** – place your Tiled maps here
  - **Tilesets** – contains tile set of generated 2D sprites to be used in your Tiled map

It is important to notice that you MUST NOT rename any of generated files and folders. 2D tiles have names indicating their rotation and the height of their matching 3D tiles. For example, a tile with file name **0.65-rock_smallFlat1_180** represents **rock_smallFlat1** tile, which has the height of 0.65 units, and is rotated by 180 degrees. Each tile has four versions with 0, 90, 180 and 270 degrees, which is necessary since Tiled rotated tiles are NOT supported by UniTiled for 3D maps. Additionally, each tile has a preview image that is placed with the tiles to make easier to identify in Tiled.

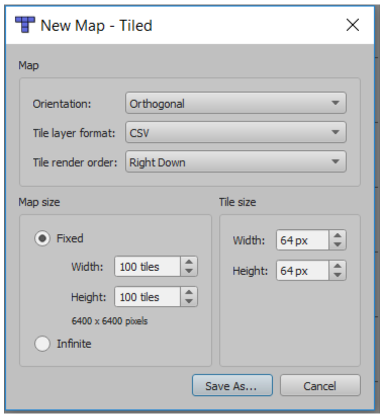
Open Tiled and then open the generated tile set file **nature.tsx** and notice how tiles are organized. Each preview image has a number on top-left indicating the height of the tile. This makes it easier to position tiles with the same height next to each other.



Property	Value
Tile	
ID	0
Type	
Width	64
Height	64
Probability	1.000
Image	0.33-lily_small_...
Custom Properties	

Step 3: Building your map in Tiled

You are now ready to start constructing your map. Create a new map in tiled and save it in **TiledMaps** folder. Make sure to set map type to orthogonal, format to CSV and tile size to the same size you used when creating your tiles.



Map

Orientation: Orthogonal

Tile layer format: CSV

Tile render order: Right Down

Map size

(●) Fixed

Width: 100 tiles

Height: 100 tiles

6400 x 6400 pixels

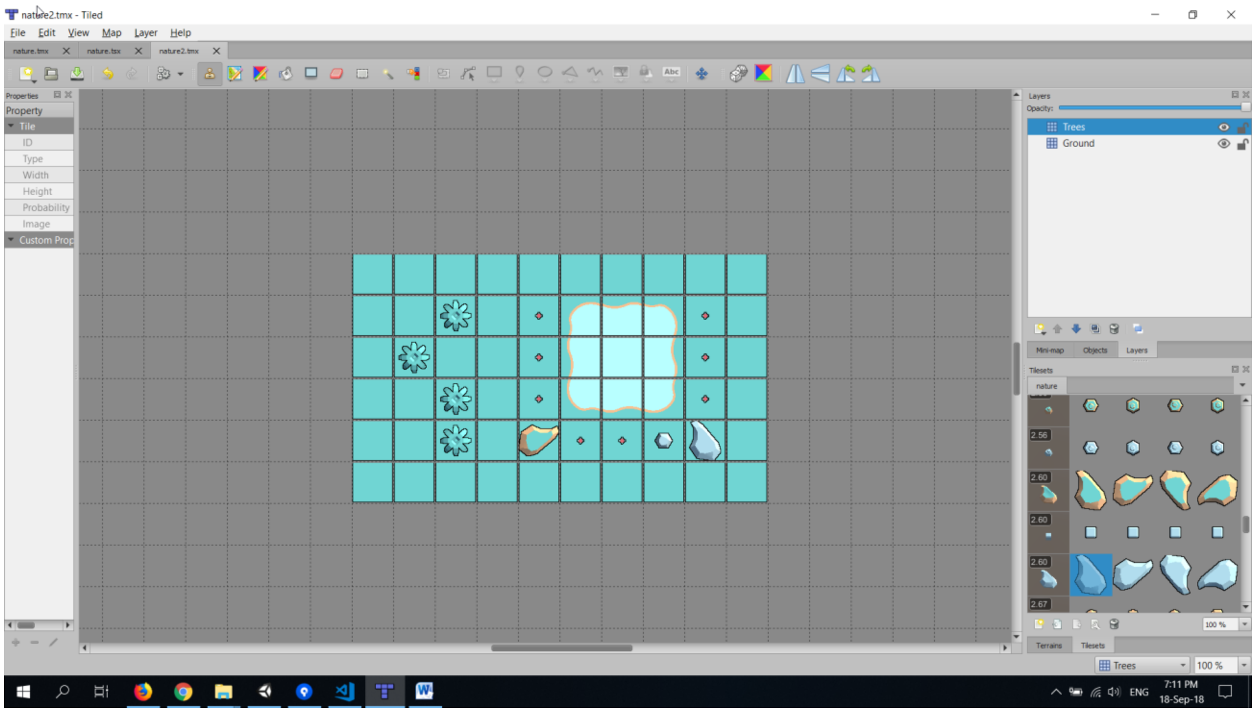
( ) Infinite

Tile size

Width: 64 px

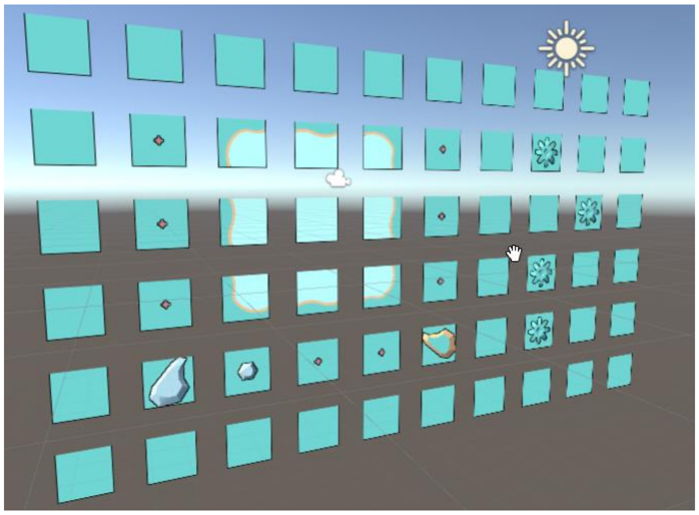
Height: 64 px

You can use Tiled layers to specify which models will be placed on top of others when building the 3D map. For example, you can use the first (default) layer for ground then create a new layer on top of it for trees and other props such as stones.

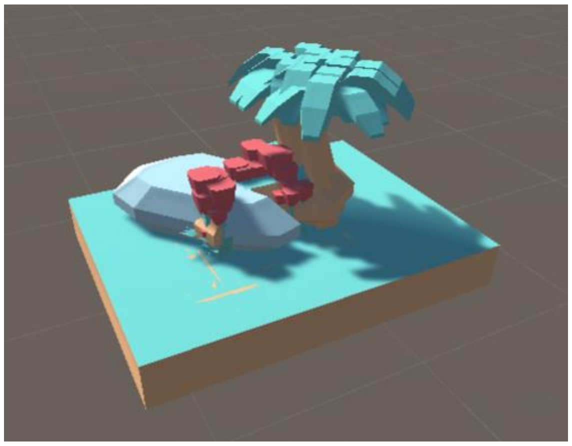


If you use the **flower_red1** that we customized earlier, you will notes the effect of the changes we have made. When your map is ready, save it and go back to Unity. UniTiled will automatically import the map as 2D map. We need to customize the importer a little before we get the final result.

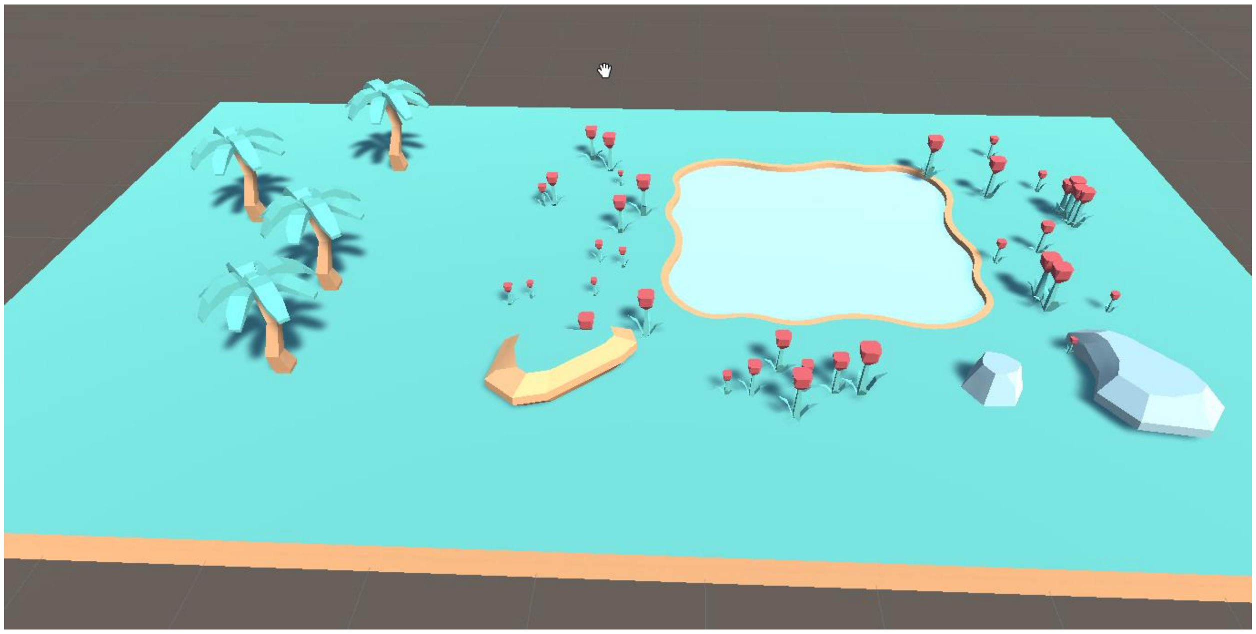
First, you might notice that the map is imported as 2D tiles rather than expected 3D.



To fix this, open the import settings of your map in the inspector and check **Build 3D Model** option then click **Apply**. Your 3D map is now constructed, but you might notice something is not right with it.



This is because Unity distance units are used to place the tiles in the scene. However, the tiles seem to have much larger scale. To fix this, set **Tile Scale** option under **Model Mapper** section of import settings to 0.1, 0.1, 0.1. This will reduce the size of the tiles and make your map finally as expected.



You can now check the colliders that are added to the tiles and notice they are missing from flowers. You can manipulate your objects manually and your updates will not be overridden when the map is re-imported. If you want to undo your modification, you can simply click **Revert** on the top of map object inspector to reconnect it to the prefab.