Tycoon Terrain Manual

Generating and Terraforming Terrain in Tycoon-, Sim- or Banished-like games

Innehåll

Tycoon Terrain Manual

Generating and Terraforming Terrain in Tycoon-, Sim- or Banished-like games

Introduction

This package was made after spending way too many hours playing games like Open TTD, Sim City, etc. The question arose as to why such games are not really made any more. The hopes of the creator are that the torch will be picked up and carried on by new game creators to help us waste our precious time even the more efficiently building cities and rail networks!

Note from the author:

This manual has been edited since version 1.2 of the package. If you have ideas, suggestions or think something is incorrectly described then please let me know at the forum thread dedicated to this package at:

http://forum.unity3d.com/threads/released-tycoon-terrain.333748/

or send me a forum message at:

http://forum.unity3d.com/members/mikael-h.261530/

or send an e-mail at mike@vikingcrew.net

Features

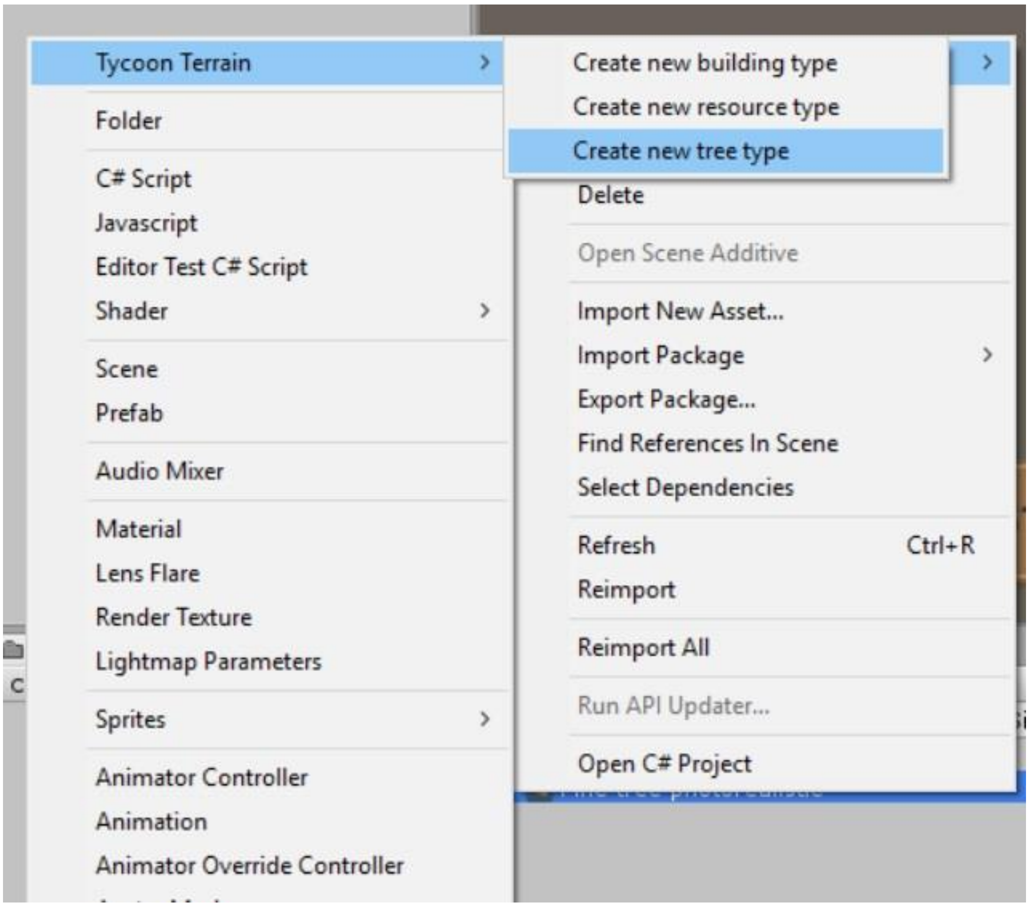
Creating new tree types

To create a new tree type you need to:

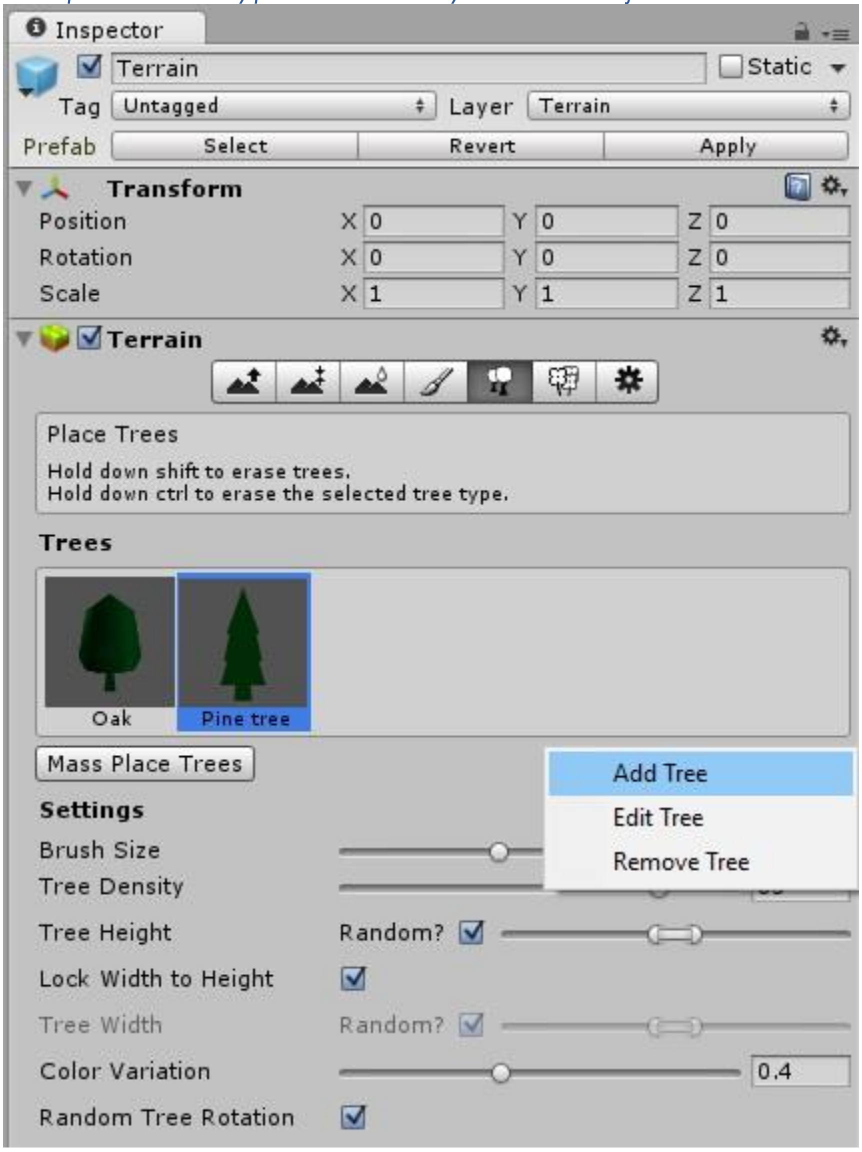
1. Create a new scriptable object to represent the data of the new tree type.
2. Setup the tree type in the Unity terrain object
3. Setup the tree type data
4. Add the tree type to the *World Behaviour* script.

Create Tree Type data

To add a new scriptable object representing the building type data you right click in the project view and select: *Create -> Tycoon Terrain -> Create new tree type.*



Setup the Tree Type in the Unity Terrain Object



☑ Terrain □ Static

Tag Untagged Layer Terrain

Prefab

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

Place Trees

Hold down shift to erase trees.
Hold down ctrl to erase the selected tree type.

Trees

Settings

Brush Size

Tree Density

Tree Height Random? ☑

Lock Width to Height ☑

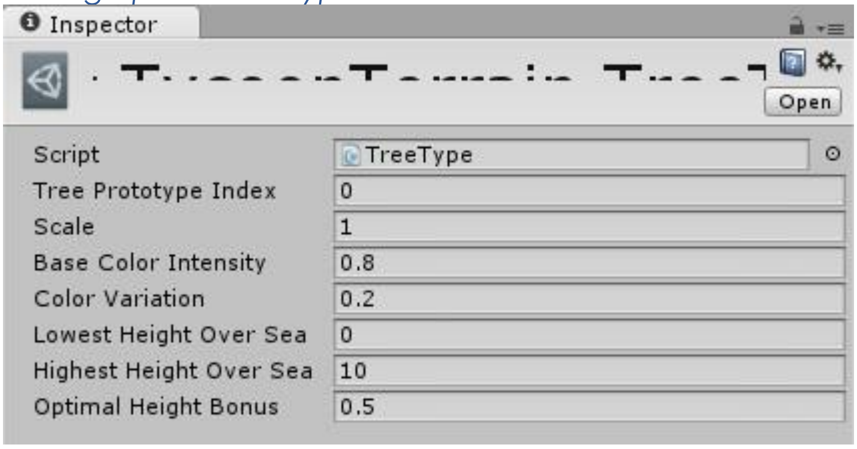
Tree Width Random? ☑

Color Variation 0.4

Random Tree Rotation ☑

Now you need to click *Edit trees* in the *Terrain* object and select *Add tree*. You will then set a prefab that will represent your new tree. Note that the *Terrain* handles trees as something it calls *Tree prototype indices* where for example the oak above has index 0 and the pine has index 1. Remember what the index was for your newly created tree in the next step.

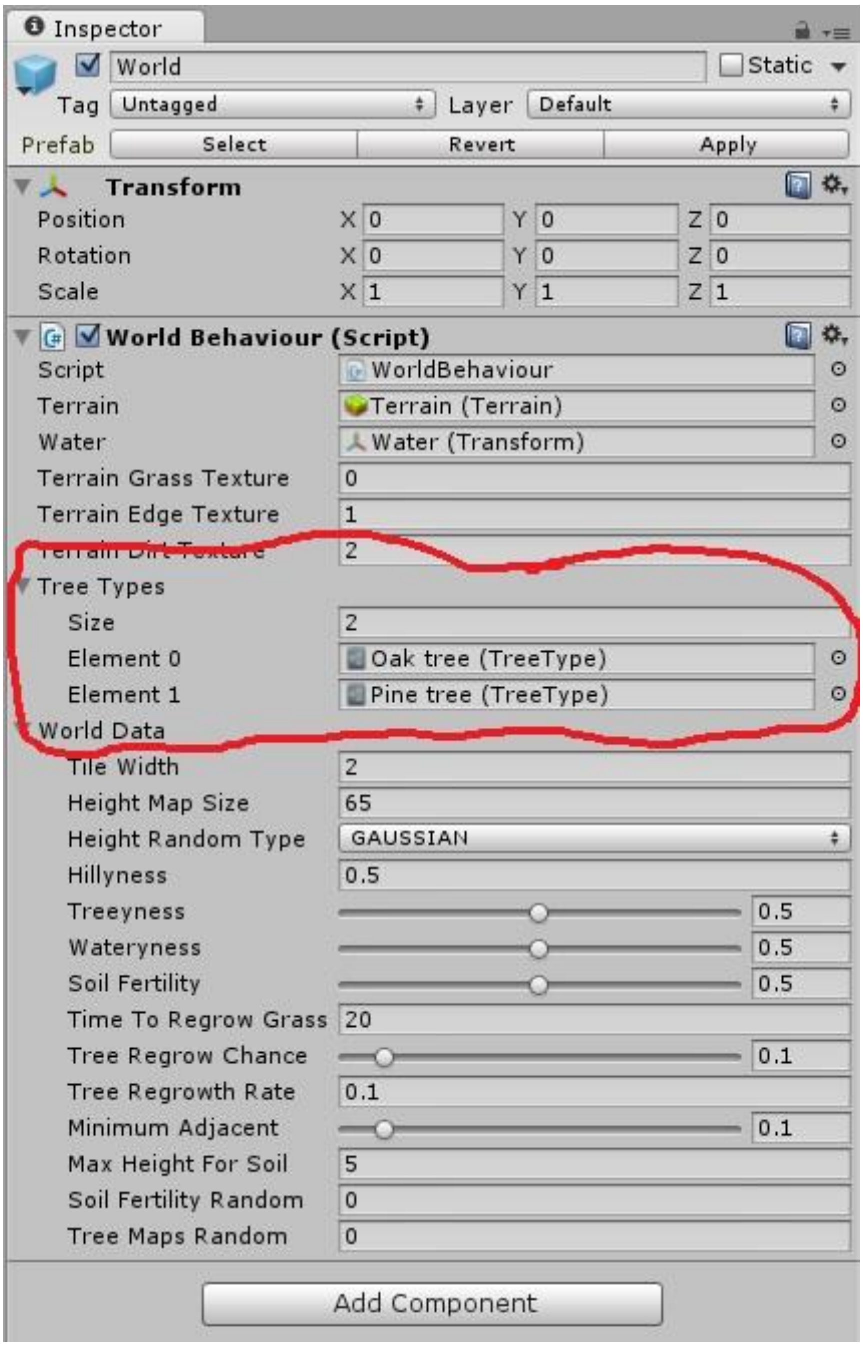
Setting up the Tree Type data



Script: TreeType
Tree Prototype Index: 0
Scale: 1
Base Color Intensity: 0.8
Color Variation: 0.2
Lowest Height Over Sea: 0
Highest Height Over Sea: 10
Optimal Height Bonus: 0.5

There are tooltips available that you can hover to get instructions for each of the values. Most importantly though, is that the *Tree Prototype Index* is set to the index of the prefab you setup in the previous step.

Add the tree to the world



Inspector

☑ World ☐ Static

Tag	Untagged	Layer	Default

Prefab: Select | Revert | Apply

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

☑ **World Behaviour (Script)**

Script	WorldBehaviour
Terrain	Terrain (Terrain)
Water	Water (Transform)
Terrain Grass Texture	0
Terrain Edge Texture	1
Terrain Dirt Texture	2
Tree Types	
Size	2
Element 0	Oak tree (TreeType)
Element 1	Pine tree (TreeType)
World Data	
Tile Width	2
Height Map Size	65
Height Random Type	GAUSSIAN
Hillyness	0.5
Treeyness	0.5
Wateryness	0.5
Soil Fertility	0.5
Time To Regrow Grass	20
Tree Regrow Chance	0.1
Tree Regrowth Rate	0.1
Minimum Adjacent	0.1
Max Height For Soil	5
Soil Fertility Random	0
Tree Maps Random	0

The last step is to tell the world that it should use this tree for world generation. Add you *Tree Type* scriptable object to this list and your tree will be used at next world gen.

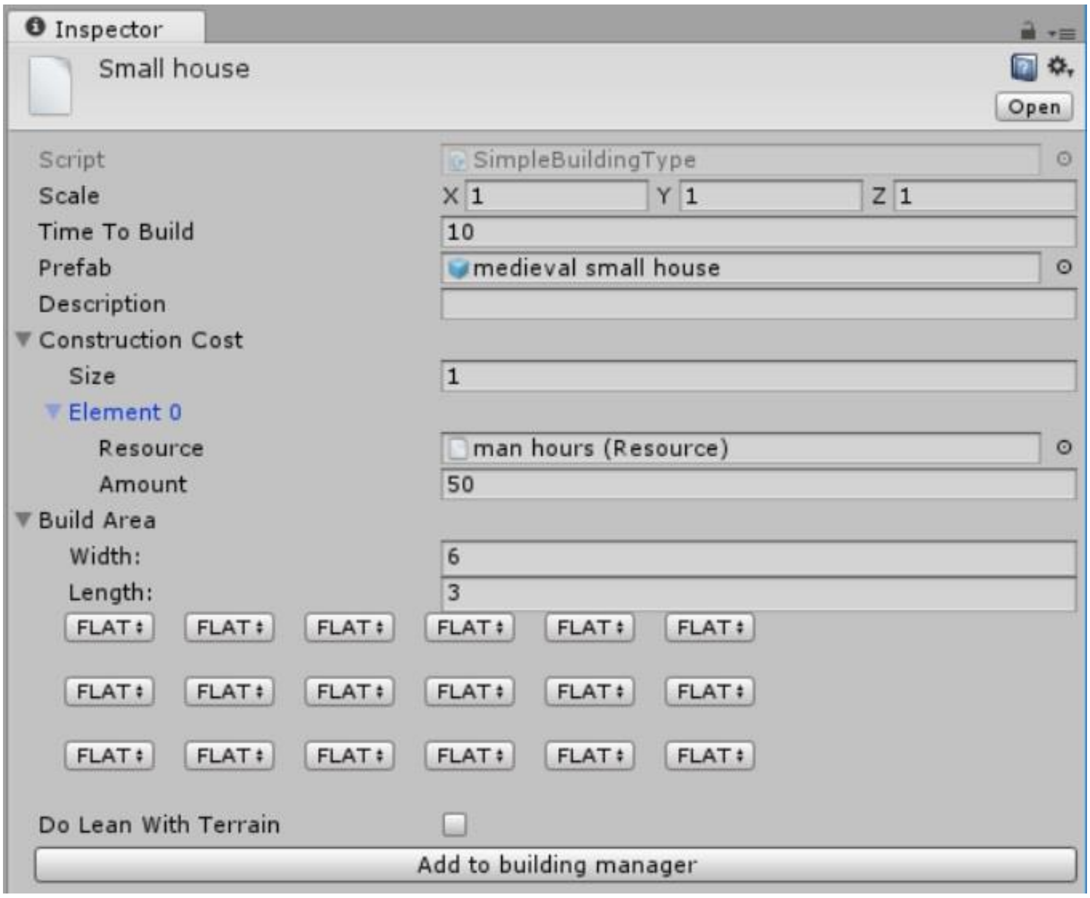
Creating new building types

To create a new building type, you need to:

1. Create a new scriptable object to represent the data of the new building type.
2. Setup a prefab for the building.
3. Add the building type to the *building manager*.

Create Building Type data

To add a new scriptable object representing the building type data you right click in the project view and select: *Create -> Tycoon Terrain -> Create new building type.*



Script: SimpleBuildingType
Scale: X 1 Y 1 Z 1
Time To Build: 10
Prefab: medieval small house
Description:
Construction Cost
Size: 1
Element 0
Resource: man hours (Resource)
Amount: 50
Build Area
Width: 6
Length: 3

FLAT	FLAT	FLAT	FLAT	FLAT	FLAT
FLAT	FLAT	FLAT	FLAT	FLAT	FLAT
FLAT	FLAT	FLAT	FLAT	FLAT	FLAT

Do Lean With Terrain □

Figure 1 By hovering over *Cash Cost* we activate the tooltip for that field. You can hover over any field to find out what it dose.

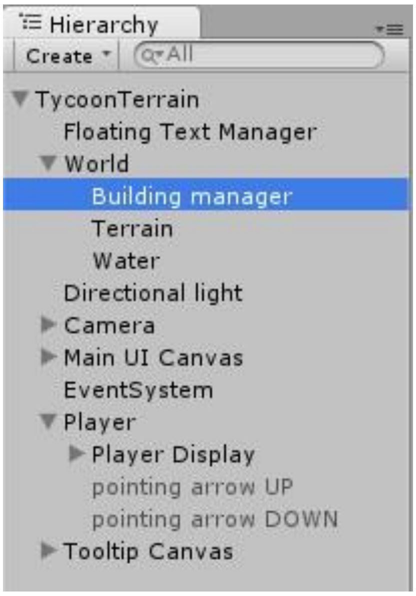
To see what a variable does in the SimpleBuildingType scriptable object just hover with the mouse over it to see the tooltip. Note that the button at the botton "Add to building manager" will check if add the building type in question to the building manager in the scene, which makes the step *Adding a building type to the building manager* unnecessary (added in version 1.2).

Setting up the prefab

Any prefab you want to use needs to have the script *SimpleBuildingBehaviour* attached to it. You do not *have to* set any of the values in this script, but, you can if you want setup the variables *Planned Building, Under Construction Building, Finished Building* and *Razed Building* by adding references to children of the prefab that you want to represent these states. The script will then make these active as the state of the building changes. In the demos the building prefabs are pretty simple with transparent models for *planned* and *in construction* states and some dust added when *in construction* but you could make prefabs with growing buildings, animated builders or scaffolding if need be.

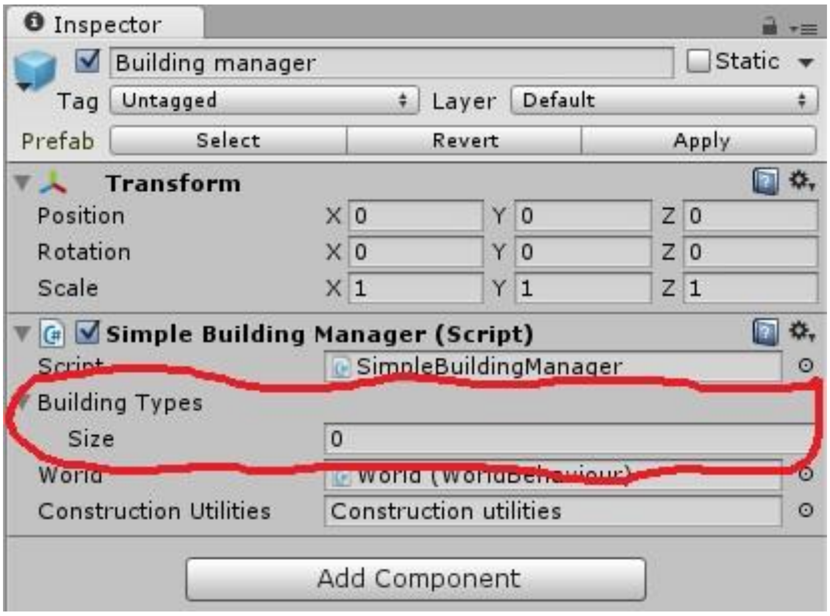
Adding a building type to the building manager

In the building manager script there is a list variable called *Building Types*. First, in the object hierarchy, select *Tycoon Terrain -> World -> Building manager.*



```
▼ TycoonTerrain
    Floating Text Manager
  ▼ World
      Building manager
      Terrain
      Water
    Directional light
  ▶ Camera
  ▶ Main UI Canvas
    EventSystem
  ▼ Player
    ▶ Player Display
      pointing arrow UP
      pointing arrow DOWN
  ▶ Tooltip Canvas
```

Next, in the inspector, add the newly created *SimpleBuildingType* to the *Building Types* list and it will show up on the building menu in game.



☑ Building manager □ Static

Tag: Untagged Layer: Default

Prefab

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

☑ **Simple Building Manager (Script)**

Script: SimpleBuildingManager

Building Types

Size: 0

World: World (WorldBehaviour)

Construction Utilities: Construction utilities

Note that there is a property in *SimplePlaverBehaviour* script that is called *AvailableBuildings* that right now simply maps to the building manager's *Building Types*. If you want to setup a research tree or something, then it is this property that needs to be overridden (and UI refreshed) to display the researched buildings.

Demo Scenes

There are currently five demo scenes in the package. They have been selected to each show a certain aspect of important features.

Terraforming scene

This scene has no buildings. All you do is terraform terrain. This can be a good scene to start working with or duplicate to start your own project.

Terraforming scene 2- different slope angles

This scene shows how a different slope angle could be used. Otherwise it is the same as *Terraforming scene.*

Town Sim scene

This scene has buildings that can be constructed in the world. It is made like any city simulation where you construct buildings as a mayor or higher political deity. The buildings currently have no effect on the world other than taking up space but you could script new behaviors for them to start generating resources or something like that.

Town Sim 2 - Banished style scene

This scene shows an example of how PBR materials can be used instead of classical Tycoon-style ones. It also shows how resource production and consumption could be applied.

Orthographic scene

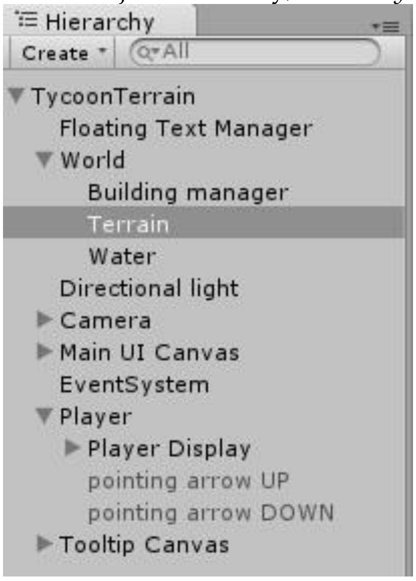
This scene was added on request from users who wanted to use 2D assets (like in the original Transport Tycoon games) and not 3D. It uses a camera with orthogonal projection instead of perspective. Not much has been done yet using this feature and if you want to provide a link to a demo you make that would be very appreciated.

Note that it might make sense to use a fully ambient lighting setup in this scene instead to retain the full illusion that you are in a 2D environment and this is the setup that exists in the scene currently.

Create a new scene

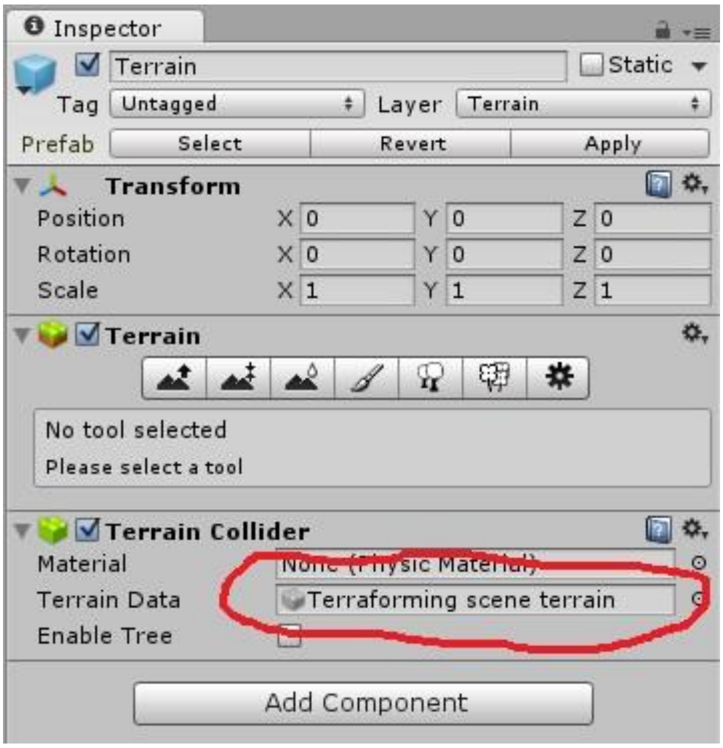
When creating a new scene, you can copy an old one. Unfortunately, Unity does not handle a copied scene gracefully when using terrain. If the copied scene has a terrain object it will still link to the old one in two places that must be set manually to a new terrain object.

1. Copy the old terrain asset (it's the .asset-file that holds all terrain data)
2. In the object hierarchy, select *Tycoon Terrain -> World -> Terrain.*



```
▼ TycoonTerrain
    Floating Text Manager
  ▼ World
      Building manager
      Terrain
      Water
    Directional light
  ▶ Camera
  ▶ Main UI Canvas
    EventSystem
  ▼ Player
    ▶ Player Display
      pointing arrow UP
      pointing arrow DOWN
  ▶ Tooltip Canvas
```

3. In the object inspector set the terrain collider to link to your new terrain asset you created in step 1.



☑ Terrain □ Static

Tag: Untagged Layer: Terrain

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

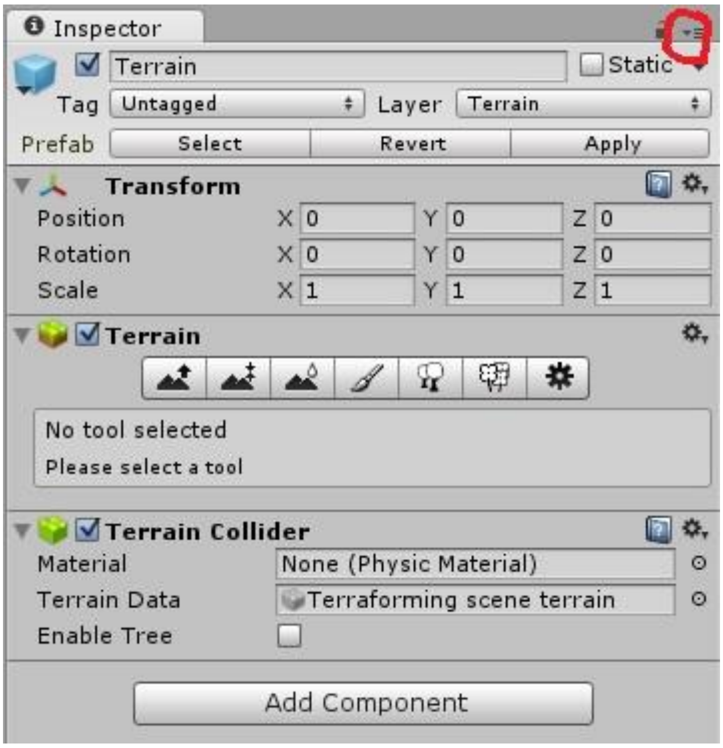
☑ **Terrain**

No tool selected
Please select a tool

☑ **Terrain Collider**

Material: None (Physic Material)
Terrain Data: Terraforming scene terrain
Enable Tree: □

4. Next we need to open up the debug options for the Terrain component as it for some reason does not show the link to the terrain asset otherwise. Do that by clicking in the top right corner of the inspector:



Inspector

☑ Terrain □ Static

Tag: Untagged Layer: Terrain

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Terrain

No tool selected

Please select a tool

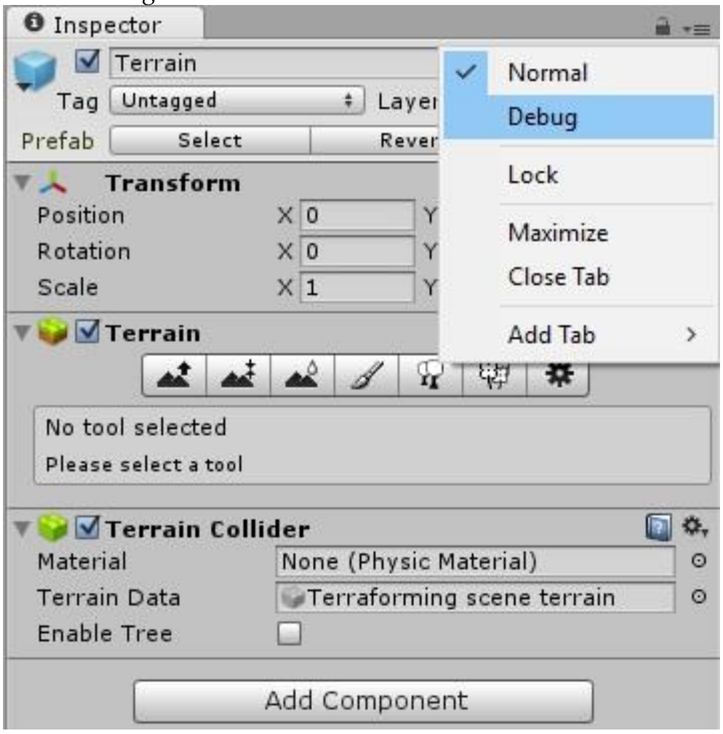
☑ Terrain Collider

Material: None (Physic Material)

Terrain Data: Terraforming scene terrain

Enable Tree: □

5. Select *Debug*:



Inspector

☑ Terrain

Tag: Untagged

Transform

Position X 0

Rotation X 0

Scale X 1

☑ Terrain

No tool selected

Please select a tool

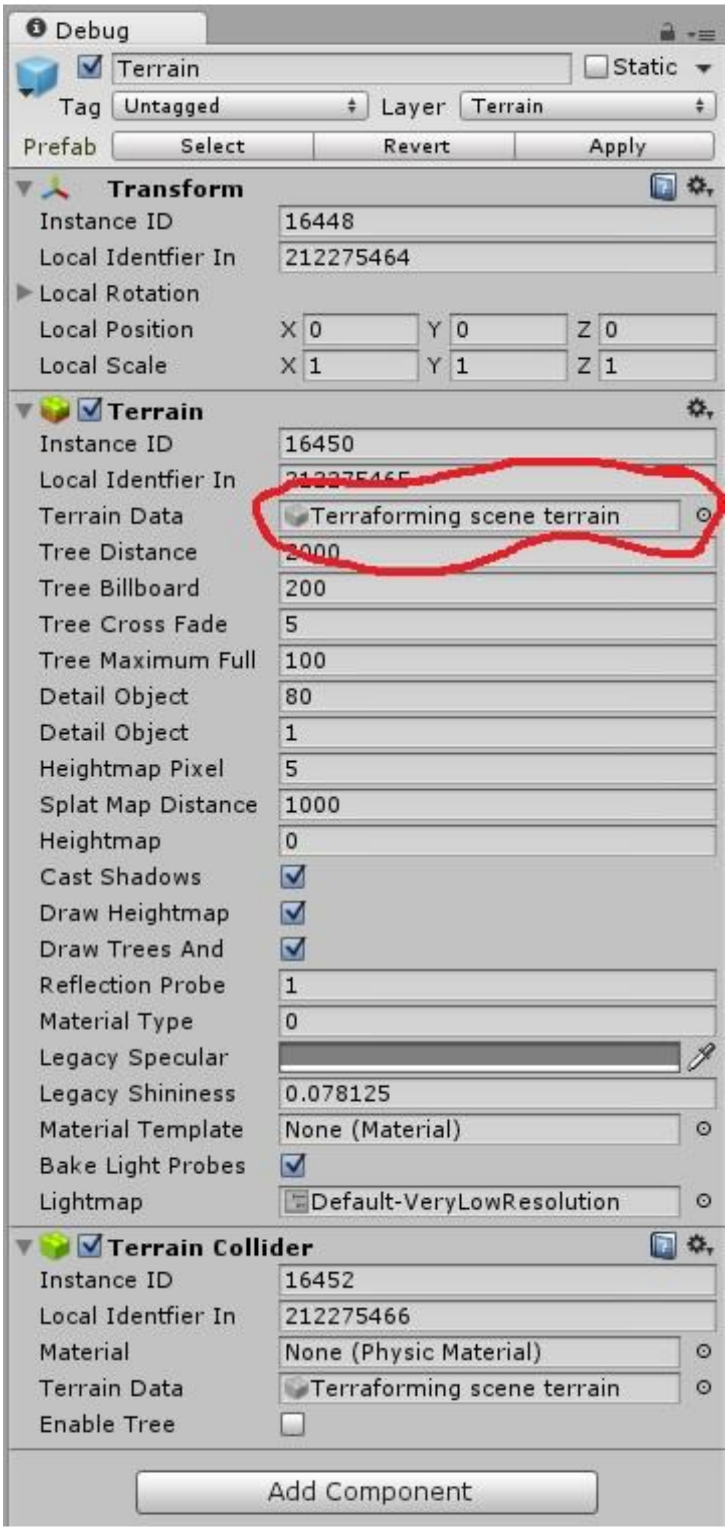
☑ Terrain Collider

Material: None (Physic Material)

Terrain Data: Terraforming scene terrain

Enable Tree: □

6. Next, set the *Terrain Data* link to link to the newly created asset-file.



Debug

☑ Terrain □ Static

Tag Untagged Layer Terrain

Prefab Select Revert Apply

Transform

Instance ID 16448

Local Identfier In 212275464

Local Rotation

Local Position X 0 Y 0 Z 0

Local Scale X 1 Y 1 Z 1

☑ **Terrain**

Instance ID 16450

Local Identfier In 212275465

Terrain Data Terraforming scene terrain

Tree Distance 2000

Tree Billboard 200

Tree Cross Fade 5

Tree Maximum Full 100

Detail Object 80

Detail Object 1

Heightmap Pixel 5

Splat Map Distance 1000

Heightmap 0

Cast Shadows ☑

Draw Heightmap ☑

Draw Trees And ☑

Reflection Probe 1

Material Type 0

Legacy Specular

Legacy Shininess 0.078125

Material Template None (Material)

Bake Light Probes ☑

Lightmap Default-VeryLowResolution

☑ **Terrain Collider**

Instance ID 16452

Local Identfier In 212275466

Material None (Physic Material)

Terrain Data Terraforming scene terrain

Enable Tree □

Add Component

7. Don’t forget to set the view back from *Debug* to *Normal* in the top right of the inspector.

FAQ

Q. *Can you switch the grid on and off in the scene?*

A. Yes, quite easily. The grid is part of the tile textures of the unity terrain object. If you don't want the grid, then just set the textures to a uniform color (or any other) texture that you would prefer!

Q: *How do I pre-calculate the cost of an operation like bulldoze?*

A: It is actually being done already in the BasicPlayerController.cs-script in order to check if the player can afford the terraforming operation before executing it. See e.g. line 216:

```
WorldBehaviour.TerraformCostCalculation cost = world.GetRaiseTerrainCost(currentTile);
```

Q: *Is there a method to get the slope steps lower?*

A: There is not a way implemented yet. If you want to have a go yourself then the place to start is the "Terrain Height"-value of the terrain object. You'll also need to edit the "heightPerStep"-property of WorldData.cs to take your changes into account. If this is important to you let me know and I'll look into this in more detail.

Q: *I imported this into my existing project and now I can't get it to work??*

A: This is placed on the asset store as "Complete Project" in order to keep layers intact. These are actually used for ray casting when the player wants to interact with the terrain or when the world needs to check if a tile is empty from blocking objects. If you already had a tags and layers asset in your project and did not import the one included in this package, then you'll need to set the layer masks of the "Player"-object and the "World"-object so that their ray casts can function properly.

Q: *My code editor does not find the WorldBehaviour-class (or some other class)?*

A: The code of this project has been placed in the namespace *TycoonTerrain*. This means that you need to add:

```
using TycoonTerrain;
```

to the top of the file where you need to access any of the scripts in this package. For more info on namespaces see: https://msdn.microsoft.com/sv-SE/library/z2kcy19k.aspx