

COMP1150/MMCC1011 Week 7 Prac

Topics covered:

- The Terrain Editor
- Sculpting and Texturing Terrain
- Dressing Terrain with Trees and Grass
- Other Terrain Tools and Features

On the Terrain Editor

Unity is a complex and evolving tool with many different features. Some of them are well-polished and powerful, some of them are new and slightly buggy, and some of them are old and in need of replacement. The terrain editor straddles all three of these categories. It lacks a lot of the features that you'd really like (e.g. node based vegetation), but also has some new and interesting features when combined with the new Terrain Toolbox package (e.g. erosion effects). As an aside, if you plan to make a game which features a lot of outdoor scenes, we'd recommend you find an alternative tools to model your terrain and implement various environmental elements, and then import them into Unity to use in your game. Unfortunately, these alternatives are generally quite expensive, which is why we've stuck to the included terrain editor for this prac.

Getting Setup with Assets

Get ready for the prac by downloading and installing the following packages:

- **Terrain Toolbox** – An extension package of the already existing Terrain component. This can be imported from your Package Manager (**Package Manager > Packages: Unity Registry**)
- [Terrain Sample Asset Pack](#) – Contains a small sample of vegetation, materials, and other terrain details. Also includes several sample terrain heightmaps, as well as some sorely needed additional terrain layers and textures.
 - Note: Some materials will not work in Unity 2021.2.9 without manually installing ShaderGraph (**Window > Package Manager > Packages: Unity Registry > Shader Graph** (latest version))
- The [Environment kit](#) on iLearn, which contains a small assortment of SpeedTree's (Connifer, Broadleaf, and Palm Tree's), as well as some water (one basic, and two advanced (with either Planar or Specular Reflections)).

Additional (optional) terrain assets that you may find useful:

- [Yughues Free ground Materials](#) (NobiAx / Yughues) – A good assortment of some quality materials.
- [Grass Flowers Pack Free](#) (ALP8310) – An assortment of several grass and flower billboard textures.

Note: SpeedTree (in the Environment kit above) is used in several games (Monster Hunter World, Resident Evil Village, The Witcher 3: Wild Hunt) and film/animation (The Jungle Book (2016), Avatar (2009), Avengers: Infinity War, Despicable Me 3, Kingsman: The Golden Circle, Jurassic World). SpeedTree also has its own editor, where you can customise Tree's before exporting them. You can find more about SpeedTree and the SpeedTree editor [on their website](#).

Check out your project

Fetch and pull the 3D project you started last week from your GitHub repo. We'll start by creating a new Scene and building our terrain, and then copy our Stonehenge from last week's scene into our new scene, with the terrain. Get your new scene ready by choosing **File > New Scene** from the menu and select the **Basic (Built-in)** option, which will create a new scene with a camera, directional light and Skybox. **Save** your new scene and you will be prompted to name it (e.g. **Terrain**) and choose its save location (in the **Scenes folder** with assets).

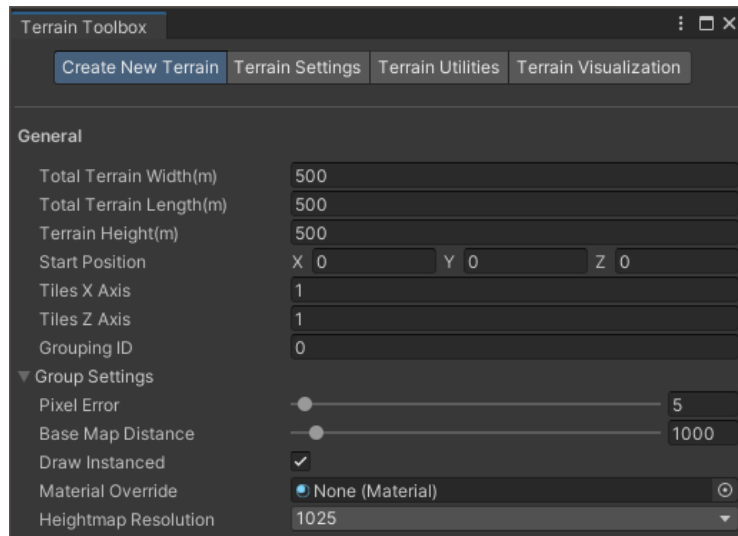
Important Note: As mentioned above, the terrain editor is one of the less stable features in Unity and may crash from time to time. **Make sure you save your scene regularly** or you may lose your work. It's a good idea to use GitHub Desktop to commit and push your project whenever you make any changes. That way you will have a backup of your work in your repo, should you accidentally lose anything.

Creating a Terrain object

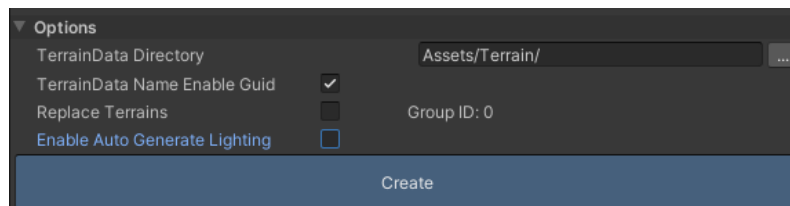
Initial Instructions

Open up the terrain toolbox package we just installed by going to **Window > Terrain > Terrain Toolbox**. This menu has a slew of features, most of which are great when developing several terrain "tiles" (chunks of terrain that can make up a large map, or in some cases an entire game world), but for now we only need one. Select the **Create New Terrain** tab, and change the following parameters:

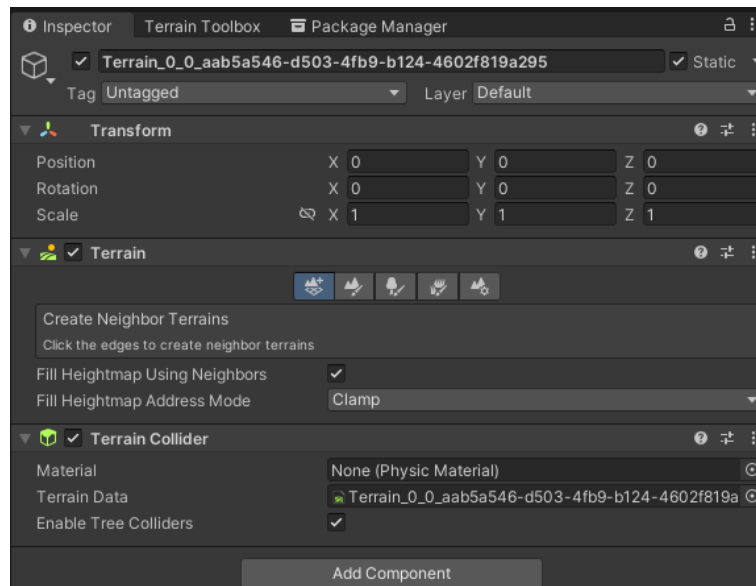
- **Total Terrain Width(m):** 500
- **Total Terrain Length(m):** 500
- **Terrain Height(m):** 500
- **Group Settings > Heightmap Resolution:** 1025



Once that is done, at the bottom of the panel in the **TerrainData Directory**, type where your terrain data should be saved (the default location should be fine), make sure **Enable Auto Generate Lighting** is left unticked (for performance reasons), and click the **Create** button.



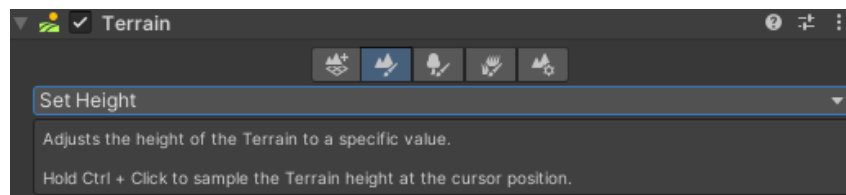
In the Hierarchy you should notice two new objects, 'TerrainGroup_0' and a child object (with the number next to 'Terrain_0_0' being its 'GUID' – Globally Unique Identifier). Select the child object, which is where our Terrain component is located.



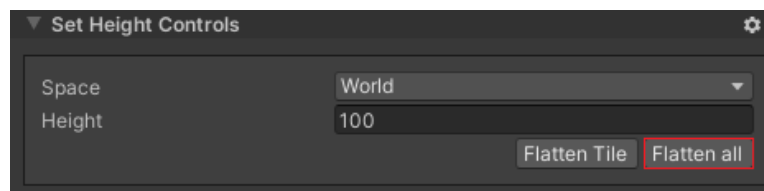
As per usual, the object has a 'Transform' component. It also has a 'Terrain Collider', a unique collider to handle other objects colliding with the terrain. Most importantly, it has a 'Terrain' component which contains several tools for manipulating the terrain.

Terrain is defined as a 'height map'. Each point on the surface has a 'height value'. You can raise and lower points to make hills and valleys. Note, however, that this does not allow caves, overhangs, or any other kind of terrain where one surface is above another (by default, though you can find third-party assets which allow this).

The initial height of any new Terrain object is 0. This allows us to raise the terrain to create hills, but we cannot lower it any further to create valleys or pools (height cannot be negative). To change this, select the **Paint Terrain** tab (second from the left). Then, in the dropdown menu at the top of the component, select **Set Height**.



This tool allows us to set the height of the terrain to a specific value. There are brushes we can use to "paint" height onto the terrain, but for now there is a quicker way. Set **Space** to **Local**, and then set the **Height** slider to **100** and press the **Flatten All** button.

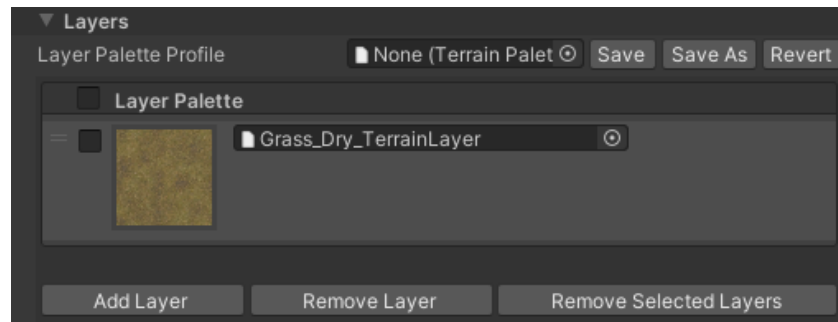
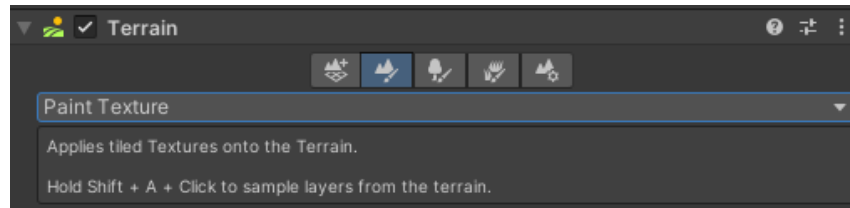


This sets the height of every point on the terrain to 100, giving us room to go both up and down. You should notice the terrain move up in the Scene view.

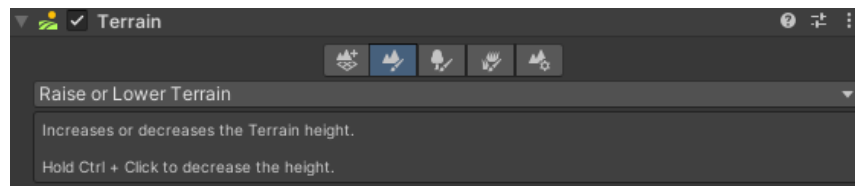
Sculpting Terrain

To help with terrain sculpting, it is a good idea to add a basic colour makes the changes easier to visualise.

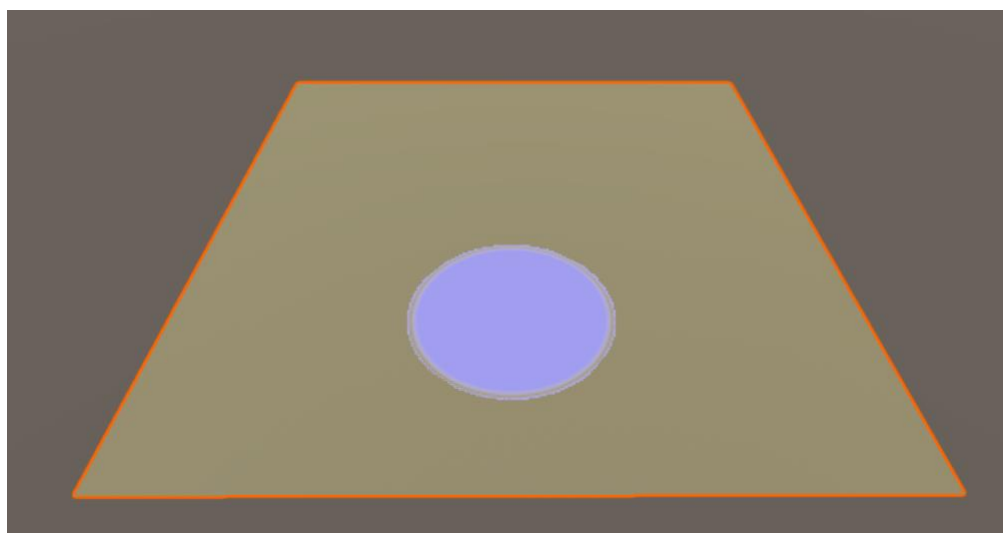
In the **Paint Terrain** Tab, select **Paint Texture** from dropdown menu. Scroll down to **Layers > Layer Palette**, and click the **Add Layer** button, bringing up the object select menu. Choose a suitable terrain texture layer (it doesn't have to be perfect, as we will colour/texture our terrain in detail later in the prac).



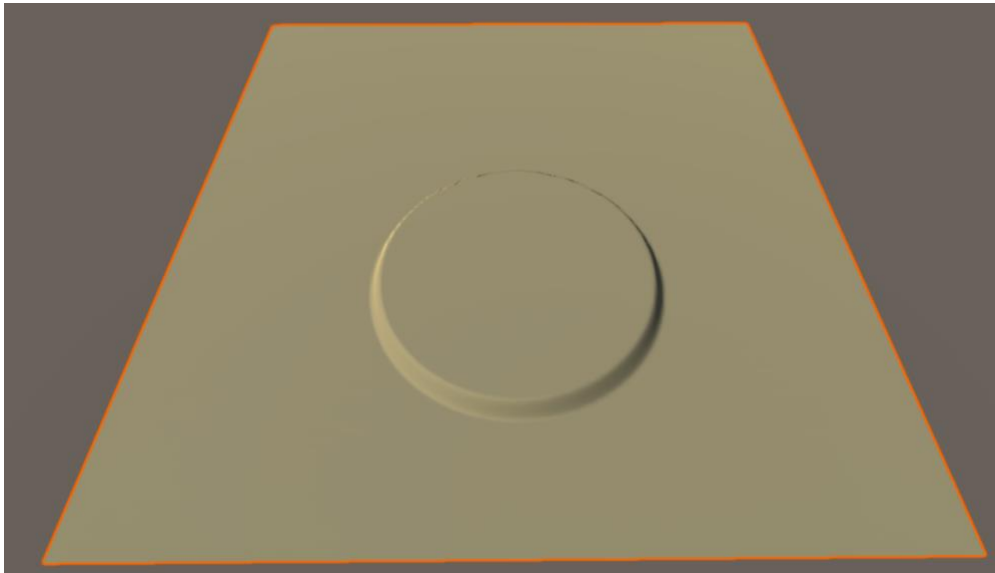
Once applied, this should immediately paint your entire terrain with the chosen texture. Now we're set to sculpt our terrain mesh. Still using the **Paint Terrain** tab, change the selection in the dropdown menu to the **Raise or Lower Terrain** tool.



Now go to the Scene view and move your cursor over the terrain. You should see a blue blob under your cursor. This is your 'brush' with which you can paint height onto the terrain.



Hold down the **mouse button/trackpad** and **drag** the cursor around to raise the terrain under your brush.

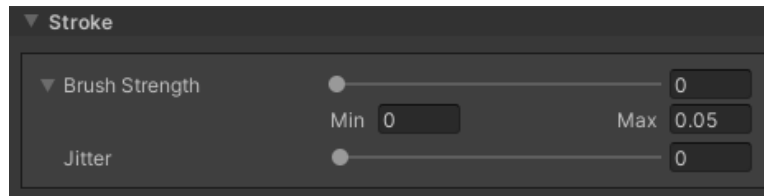


Press the **Ctrl/Cmd key** while painting to lower the terrain. Remember, you can lower the height to a minimum value of 0.

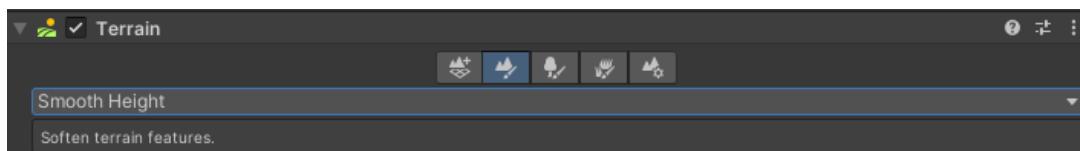
To make the sculpting of terrain smoother (with less 'staircasing') you should customise your brush tool. First, adjust the brush **Strength** and **Size**. 'Strength' controls how strongly the brush affects the terrain height. An incredibly low number is recommended for the first three brushes (less than 0.05).



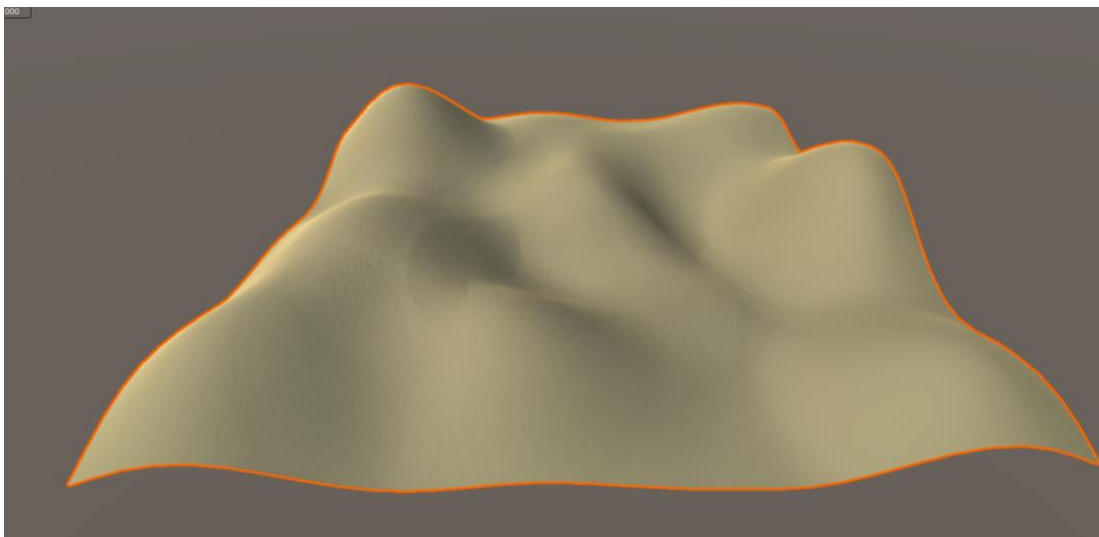
You can also use keyboard shortcuts to help you sculpt the terrain with the brush tools more quickly. These shortcuts include **A** to toggle brush strength, **S** to toggle brush size, and **D** to change brush rotation. You should also change the **Min** and **Max** values for a brush's slider by clicking the dropdown under each triangle (which limits the selectable values for the keyboard shortcut). You can also find 'Jitter' in this menu, which determines the extent to which the brush will automatically modulate between the desired 'min' and 'max' values (i.e. a shortcut for random noise).



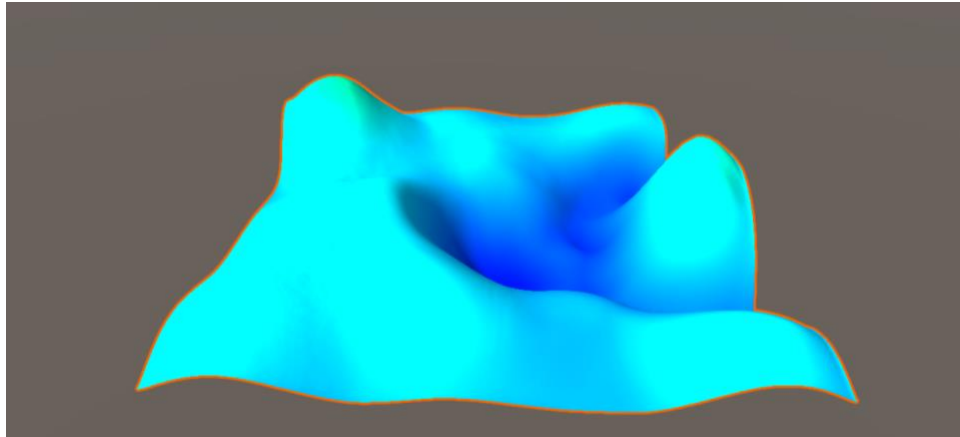
Another option you might find useful is **Smooth Height**. Height smoothing is useful for getting general smoothing of terrain, as well as getting rid of minor “staircasing” (steep inclines/declines in terrain that result in obvious pixelation), it is advised to have your Smooth Height brush set to a high strength. You can smooth height by holding down the Shift key while sculpting your terrain or by switching to that tool in the terrain paint texture tab (just be sure to change back to the **Raise or Lower Terrain** option when you want to go back to shaping your terrain).



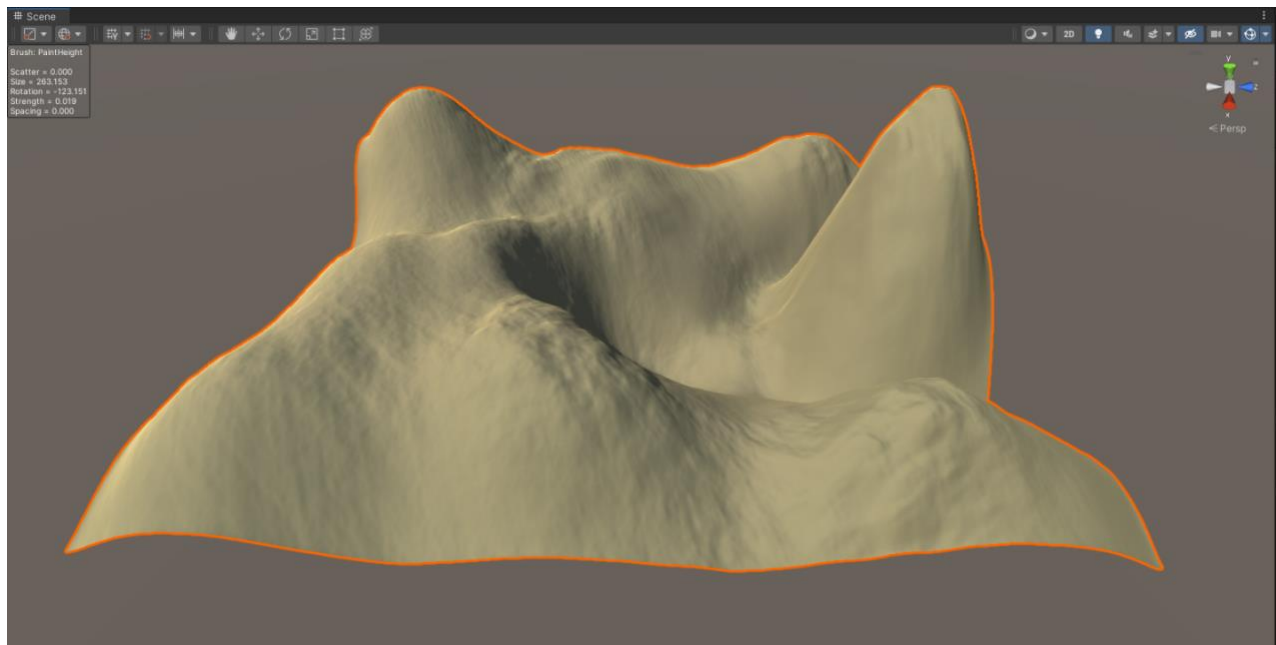
Quickly spend a few minutes making a basic terrain (no more than 10, you can tweak it later). It would be ideal to make a couple of hills, and maybe a path where some water could pass through.



When changing terrain height, you might want to make sure that the height does not get too great and surpass your height limit. You can visualise your height and its closeness to this limit via an ‘Altitude Heatmap’. View your ‘Altitude Heatmap’ by going to **Terrain Toolbox > Terrain Visualization (Tab) > Visualization Modes > Altitude Heatmap**. Your terrain should look something like this (the colours may vary depending on their height, obviously). To go back to sculpting you’ll need to set the **Visualization Modes** back to **None**.



After you've checked your heatmap and finished shaping/smoothing your overall terrain, it is worth quickly adding some natural variation back in. Spend a few minutes using in-built brushes 4 to 9 to add some roughness/noise to your terrain to make it look more natural.



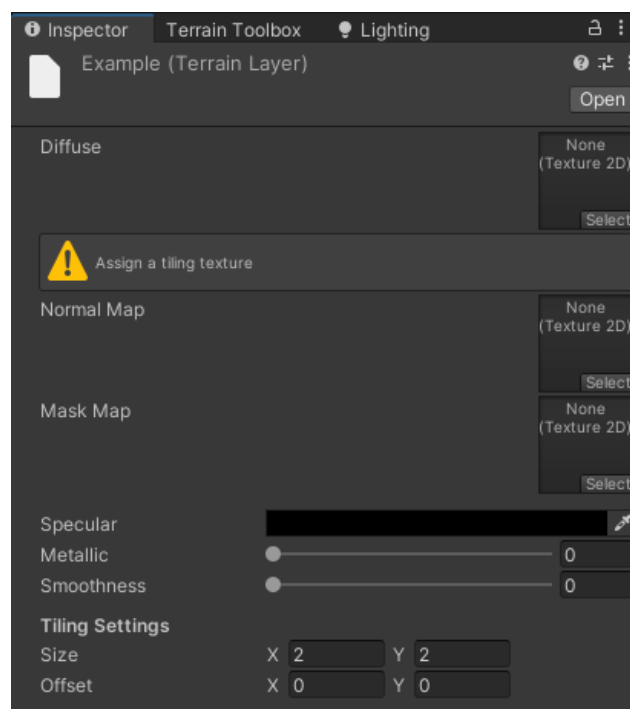
REMINDER: Save, Commit and Push to GitHub

Adding Textures

Now that you've finished sculpting a basic terrain, it's time to paint it with some appropriate textures. To do this we will be using premade 'terrain layers' (from the 'Terrain Sample Asset Pack') to create a palette of textures appropriate for painting our terrain.

Information on Terrain Layers

Painting terrain incorporates the use of 'Terrain Layers', which function similarly to how 'Materials' function. As such, 'Terrain Layers' are nearly identical to create and configure. Like materials, you will need a **Diffuse map** and a **Normal Map** (you can ignore 'Mask Maps' for now). You can also change a 'Terrain Layers' **Metallic** and **Smoothness** values, as well as its **Tiling**. You can create terrain Layers by clicking **Assets > Create > Terrain Layers**, or in the **Create New Layer** field, underneath **Paint Terrain > Paint Texture > Layers**.



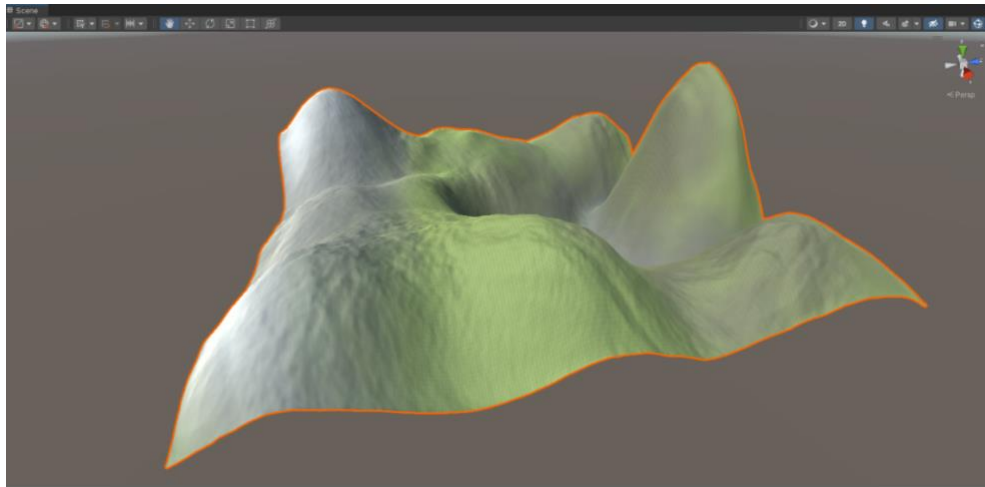
Go back to **Paint Terrain > Paint Texture**, and scroll back down to the **Layer Palette**. Remove the existing 'Terrain Layer' if you'd prefer (using the **Remove Layer** button), and add at least four 'Terrain Layers' using the **Add Layer** button. Choose layers/textures that you think will fit the biome/theme of your terrain you'd like to create (e.g. for a desert, a mountain, a forest, another planet, etc). For example, you might like some of the "Pebble" layers if you're going for something volcanic, or a mixture of pebbles, grass, and snow layers if you're thinking of an alpine biome.

You will need to make copies of some pre-existing layers if you are going for something fairly specific (e.g. if you are making a desert, you may need to make a copy of the "Sand" terrain layer, but slightly change its **Metallic** or **Smoothness**), and blend the terrain layers together.

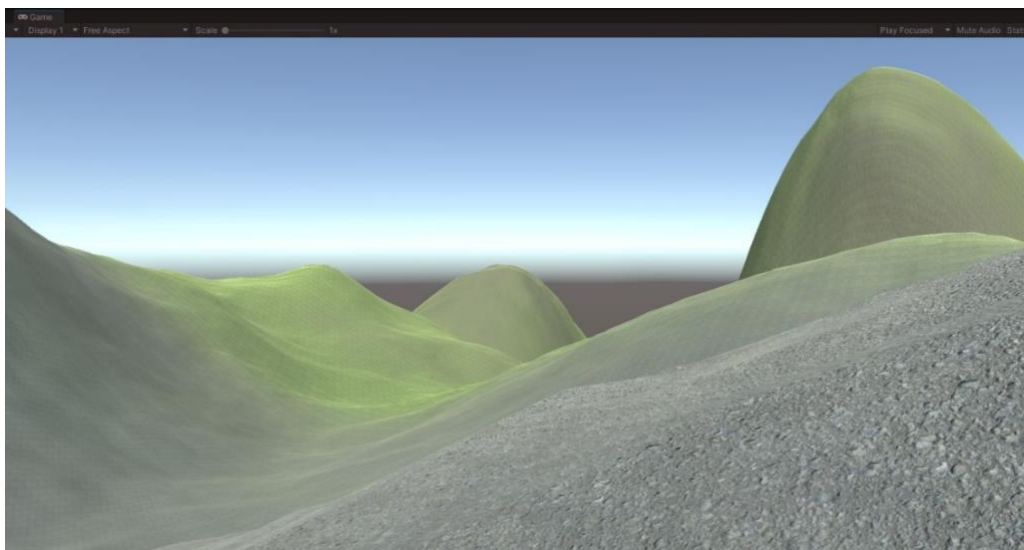
It is worth noting that these painted terrain layers work sort of similar to Photoshop layers, where one layer is painted on top of one another. Therefore, if you move the layers around, your terrain will change appropriately.

Because of this, once you have decided on the layers you are going to use it is good practice to **Save** your Layer Palette, so that you don't accidentally misplace or reorder the layers. This is also a great way to set up terrain palettes suited for different biomes.

Once you have decided upon your terrain layer palette, paint your terrain appropriately (again, don't spend more than 10 minutes on this).



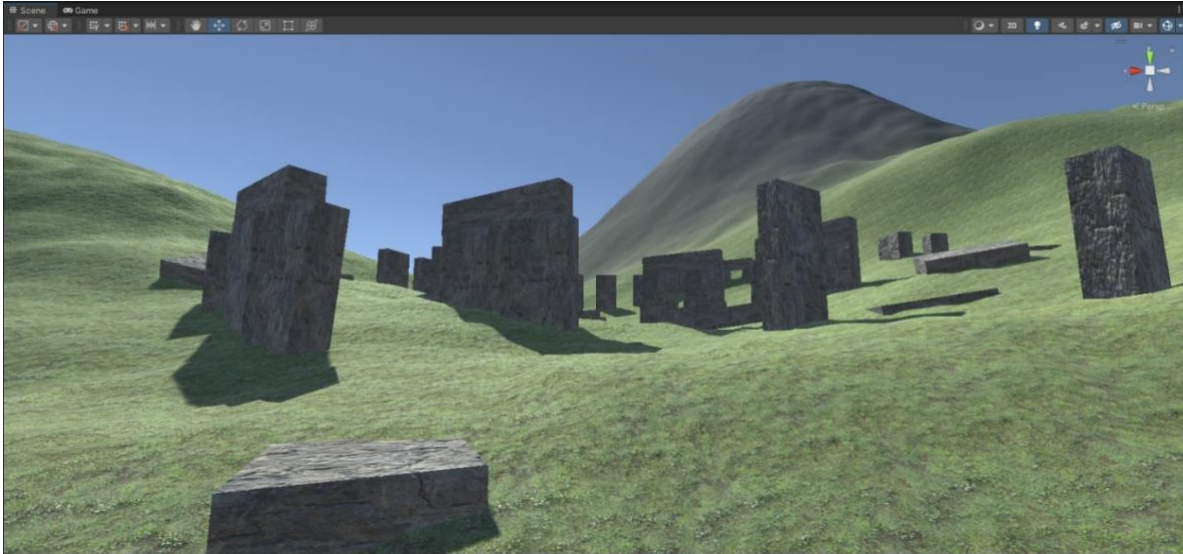
Now add your **Player** prefab from last week. Make sure you position it slightly above the terrain and delete the Main Camera, then play the game. Go for a walk around the world you have created and try not to fall off!



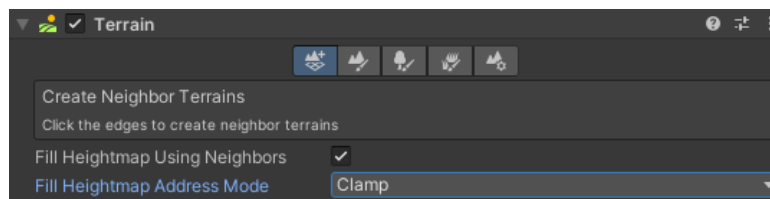
REMINDER: Save, Commit and Push to GitHub

Adding Stonehenge (and extending your terrain)

Save this scene and then re-open the old scene in which you built Stonehenge. In the Hierarchy, group all the stones that make up your henge under a single empty parent, called “Stonehenge”. Select this parent and **drag it into the Assets** panel. It should automatically create a prefab called ‘Stonehenge’. You can now use this prefab to put a copy of the henge in your new scene. Open the scene with your terrain in it, and drag the Stonehenge prefab into the Scene. Move it into an appropriate spot in your world. If it doesn’t fit (there are floating stones), just try and make it fit as best as you can, or manually rotate/move objects if you’d prefer.



If you don’t have room for your Henge prefab on your terrain, you can create an additional adjacent terrain tile by using the **Create Neighbour Terrains** tool.

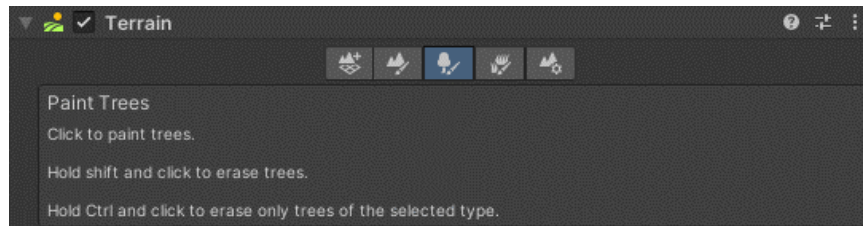


This tool allows you to click next to your terrain tiles, and add additional terrain objects, which join seamlessly. This is handy if you run out of room on your terrain, and would like to add more space to it. By ticking **Fill Heightmap Using Neighbours** and setting **Fill Heightmap Address Mode** to **Mirror**, Unity can give you a head start on your new terrain tile, by finishing off height details from the previous terrain. Alternatively, leave **Fill Heightmap Using Neighbours** unticked, and adjust your neighbour terrain height using **Paint Terrain > Set Height > Flatten Tile** (make sure you **do not select** Flatten All!). This should provide plenty of space to add your Henge prefab to your Terrain scene.

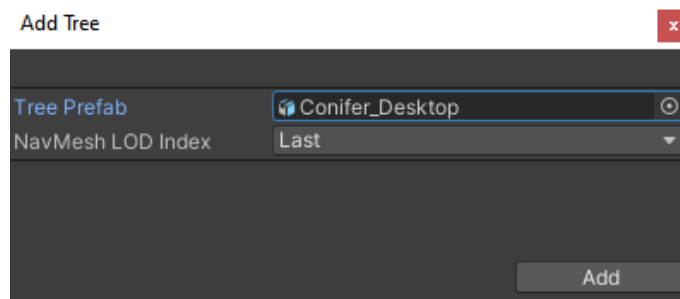
REMINDER: Save, Commit and Push to GitHub

Adding Trees

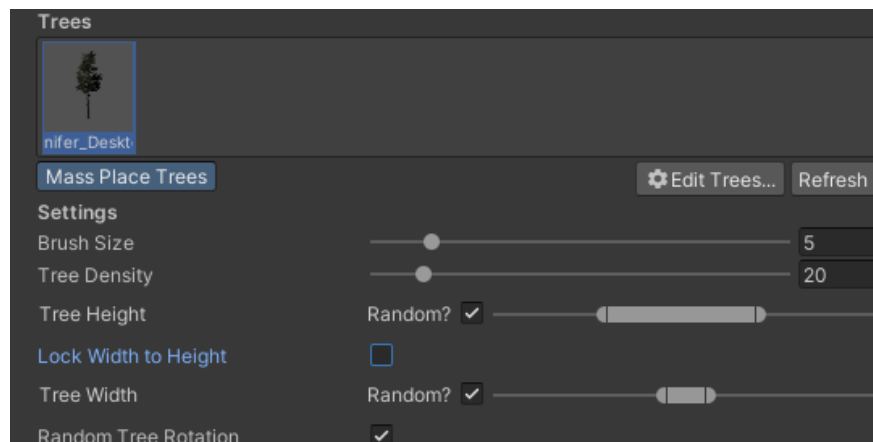
Select the **Paint Trees** tab in the Terrain component.



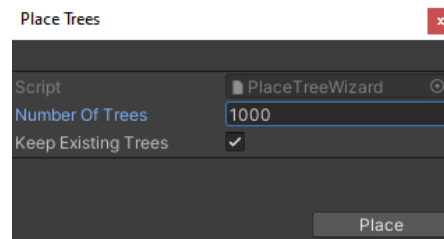
Select **Edit Trees... > Add Tree** to add a new tree to the palette. A window will appear prompting you to assign a tree. Open **Assets/SpeedTree/Conifer** and drag the **Conifer_Desktop** tree into the **Tree Prefab** slot, or simply use the object picker button, then press **Add**.



You can now paint trees onto your landscape. To make sure there is a bit of variety in your scene, tick **Tree Height Random?** and change the slider so there is a good variety of heights, as well as untick **Lock Width to Height**. You may want to lower the **Tree Density** and **Brush Size** variables first, to make sure you don't add too many trees at once. You can also lower the **Tree Height** slider if the trees that appear look too large for your scene.

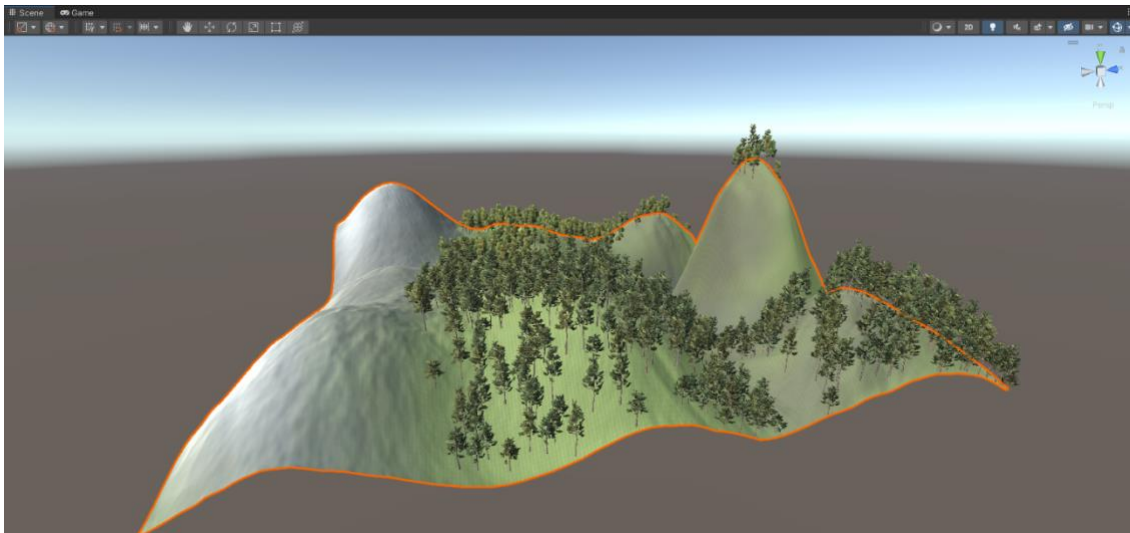


To help you get started with tree placement, you can press **Mass Place Trees** button (under the Trees palette). This lets you instantly place a number of trees on your terrain according to various parameters. If you are using this tool, set the **Number of Trees** to a number that is both appropriate for the scale of your terrain and doesn't overburden your machine (i.e. if you're using a laptop, you might want to keep this number under 100). Press **Place** when you are ready to place the trees in your scene.



From here, you can manually paint trees on your terrain using a brush tool, as well as remove trees (by holding **Ctrl/Cmd**) that don't fit your scene that may have been added by the 'Mass Place Trees' tool. Set up the Brush Size and Tree Density accordingly.

Painting trees can be rather haphazard, and there is no option to adjust their positions once they have been painted. If the position of a tree matters to you, you can drag the prefab directly into your scene instead. Again though, don't spend more than 10 minutes on this.

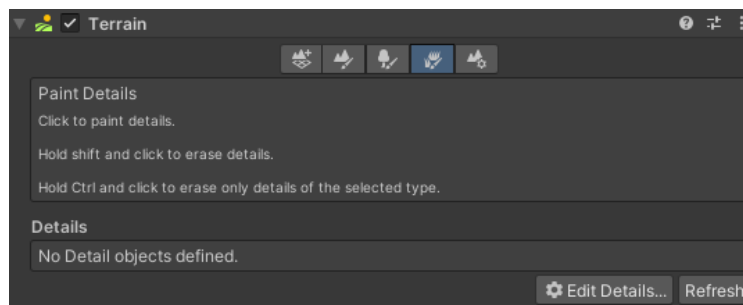


REMINDER: Save, Commit and Push to GitHub

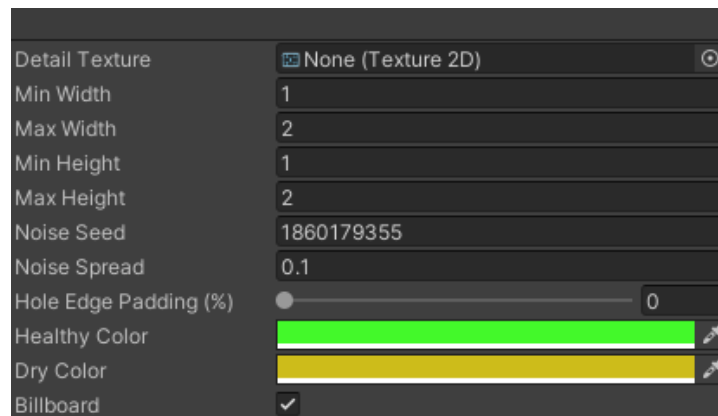
Adding Grass

Historically, grass in most video games has not been in 3D, but has instead been called 'billboard details', with a 'billboard' being defined as a 2D image which is used to represent a 3D object in the world. The image is drawn standing vertically, and always faces the player so it appears to be 3D. Billboards are useful for adding a large amount of detail without having to build 3D models. As long as the player doesn't look too closely, they appear realistic. Billboards can still be pretty computationally expensive to draw, so usually we try to use as little as possible to get away with the right effect.

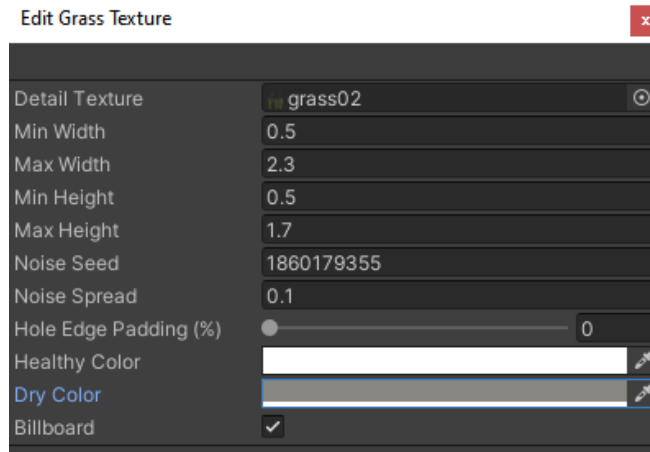
Select the **Paint Details** tab in the Terrain component:



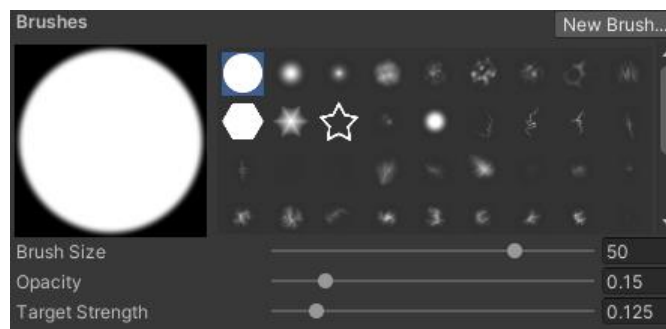
First we need to add some grass to our palette. Down below the brush settings in the Details section, select **Edit Details... > Add Grass Texture**. A new window will appear, prompting you to assign a detail texture.



Click the object picker button for 'Detail Texture', and find a suitable grass image (e.g. **grass02** from the 'GrassFlowers' package). The editor will automatically add some randomness to the height, width and colour of the grass it paints to add more realism. The default grass is rather bulky. Change **Min Height** to 0.5, **Max Height** to 1.7, **Min Width** to 0.5 and **Max Width** to 2.3, for more realistic grass painting parameters. Also make sure to edit the **Healthy Colour** to be white, and **Dry Colour** to be slightly grey (or brown). Once you're done, press **Add**.



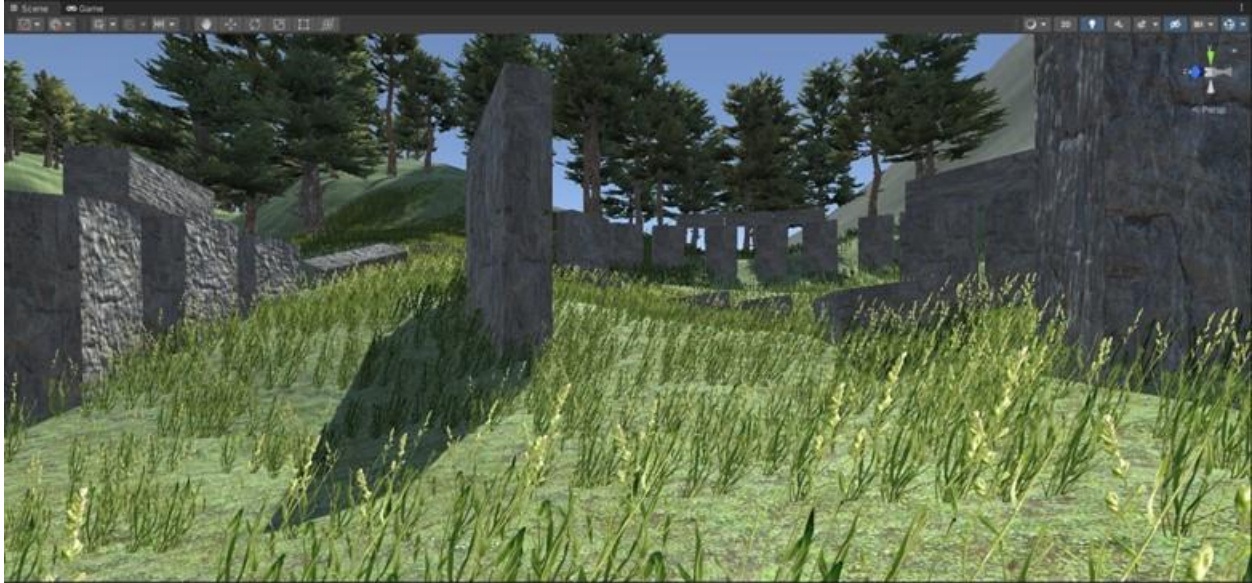
Next, we need to set the density of the grass painting tool. There is no need to cover the area with unnecessarily dense grass, so you can adjust the settings of the brush to avoid this. A decent starting point can be found by using the default brush (*builtin_brush_1*) and changing **Opacity** to 0.15 and **Target Strength** to 0.125.



Note: It is always better to paint less grass first, then add more after checking the performance of your game. Note that you can set the **Detail Density** setting in the **Terrain Settings** tab of your terrain component, to override the maximum density of all grass and trees on a terrain object.

The **Terrain Settings** tab has several options to help modify your terrain, such as resizing the **Terrain Height/Width/Height**. You may have also noticed that the grass doesn't appear in your Scene view until you zoom in close to your terrain (to help things run more smoothly), which can be adjusted in **Terrain Settings > Billboard Start**.

You can now safely **paint grass onto your terrain**, just as you did with textures. If you want to remove grass, press the **Shift** key while painting.



If you play the game now you should see what appears to be free-standing grass on the ground. If you watch carefully while rotating the camera, you should notice that the grass always looks the same, no matter which direction you look from.

REMINDER: Save, Commit and Push to GitHub

Other Tools/Features

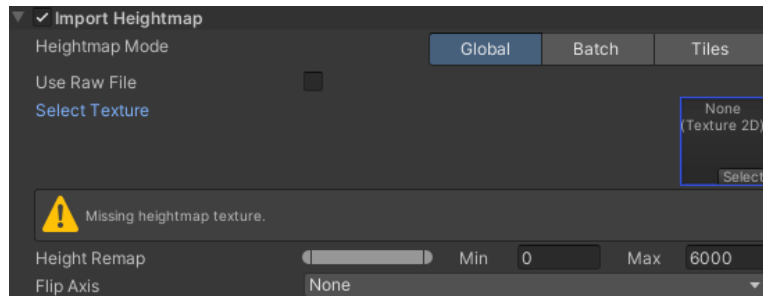
There is a raft of other tools that you might find useful when creating your terrain. To complete today's practical, implement at least **five (5)** of the features outlined below to add additional content or detail to your scene.

Adding More Terrain

Create and Sculpt a Neighbour Terrain: Create a fresh neighbour terrain, which can be designed to include some of the features below (e.g. if you made a terrain that was intended to be a forest with a walkway, maybe include a neighbour terrain that can be used to hold a lake).

Create a Terrain Backdrop/Create a New Terrain using a Heightmap: Unity also allows you to create terrains using premade Heightmaps. You can find examples of heightmaps online or on the Unity Asset Store (in some cases, using examples of real locations -- a good tool for referencing real life heightmaps is [Heightmapper](https://github.com/Tangram-Games/HeightMapper), by Tangrams).

To make a new terrain using a heightmap, follow the same procedure for creating a terrain but tick the **Import Heightmap** checkbox in **Terrain Toolbox > Create New Terrain** and click the dropdown triangle.



From here, click the Select Texture field and select one of the Heightmaps (the black and red images) from **Assets/TerrainSampleAssets/Textures/Heightmaps**, or use one of your own. Once you have done this, you may get a warning regarding the heightmap resolution, change the **Heightmap Resolution** to whatever Unity recommends. From here click **Create** and tweak your new terrain however you see fit.

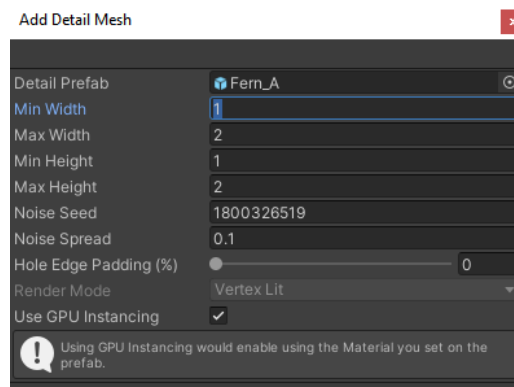
You may have to experiment with the terrain width/length/height values of heightmapped terrain to make it appear more realistic and get rid of staircasing. If you are setting up the new terrain as a background then check the perspective from within your scene to make sure it works appropriately, or if you want it to connect it to your old terrain you might have to move it to be adjacent and adjust the intersections between the two terrain objects to make them seamless.

Adding Natural Details

New Trees/Grass: You might have noticed that there are several types of grass and a few more trees included in the recommended packages. Try incorporating some of these into your scene to add some more variety in the vegetation. If you're adding several different types of grass, try experimenting by turning **Billboarding** off on certain types of grass (**Paint Details > (Select a texture) > Edit Details > Billboarding**).

In addition to holding **Shift** and **Left click** to delete all grass, you can use **Ctrl/Cmd** and **Left click** to delete the selected type of grass

Additionally, if you are on a powerful device, you might like some of the **Detail Mesh** (3D) assets provided in the 'Terrain Sample Asset Pack'. You can add these by selecting **Paint Details > Edit Details... > Add Detail Mesh**.



Water: There are lots of different ways to represent water depending on the effect you want. Making realistic water is complicated as well as computationally expensive, but often a simple approximation will be good enough.

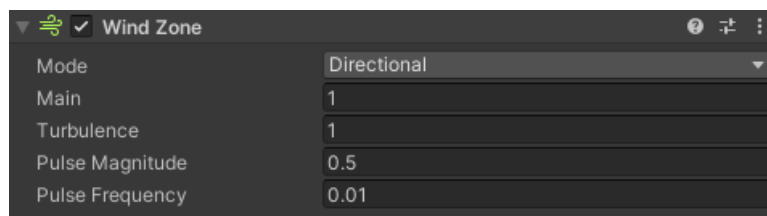
There are a few different types of water prefabs supplied for you in the 'Environment Kit', each with varying performance costs, features and appearances. For my scene, I found 'WaterProDaytime' to be suitable (Found in **Water/Water_FlatTransparent/Prefabs**). This water has mild performance costs (due to the reflections), and is also a bit flat, but I felt that this water felt quite calm and suitable for my alpine environment. Again, pick which water suits your scene best.

Once you've decided upon your water, place it in your scene. Move and resize it until it fills the area you want to contain the water. Make sure the water is large enough, so its edges intersect with the terrain.



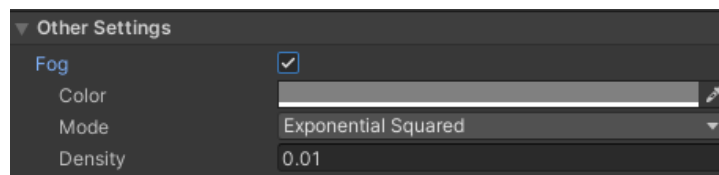
Note that these implementations of water are just flat planes. Moving the camera underwater makes the water seem to disappear. For water the player can swim in and under, you will need to find a different implementation (we look at this in more detail in a future prac). Also, if you don't want the player to fall below the water, try adding a **Mesh Collider** to it.

Wind Zones: Trees can be brought to life by adding a little wind. This is achieved by adding a **Wind Zone** object to the scene (**GameObject > 3D Objects > Wind Zone**). A 'WindZone' has two types, **Directional** and **Spherical**. 'Directional' wind blows in the same direction across the whole scene (like 'directional lights'). 'Spherical' winds blow outwards from a point. Directional winds are good for natural wind. Spherical winds are good for explosion effects.



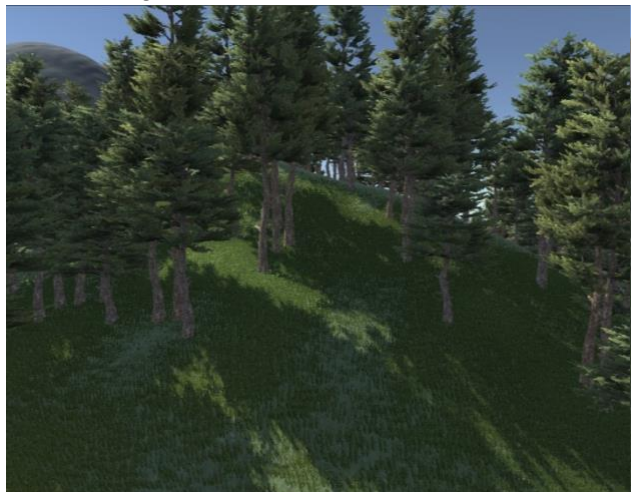
'Wind Zones' also have parameters to change the strength of the wind and implement random "pulses" (or gusts) in the wind. Adjust these variables until the trees move the way you want. Note that the Conifer trees are fairly sensitive to these Wind Zones, so should look better if you tune the wind effects down.

Fog: In a large open scene, it can seem a little odd that distant objects are as clear and crisp as close objects. A standard solution is to add a small amount of fog to make distant objects hazy. This kind of fog can be enabled in the **Lighting** panel. To view the Lighting panel go **Window > Rendering > Lighting Settings**. Navigate to the **Environment** tab, and in **Other Settings**, tick the checkbox next to Fog to turn it on, revealing three extra options.

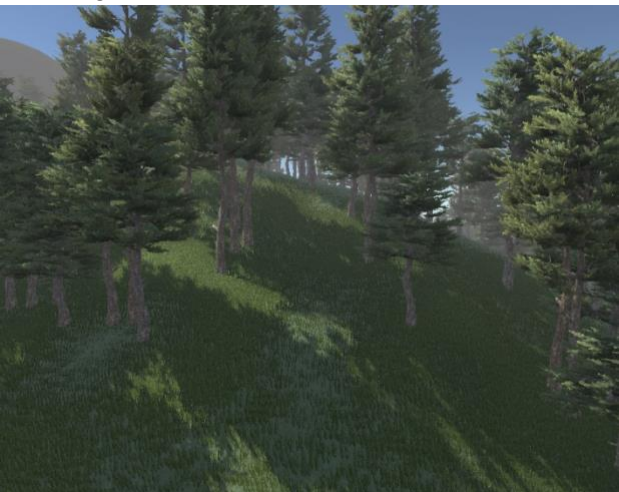


Toggle fog on and off and check the Scene view to see what effect it has. Experiment with different settings to see what they do. This kind of fog isn't very sophisticated. If you want to implement realistic or volumetric fog/mist in your game, you'll need a more complex effect. There are several packages available in the Asset Store which could do this for you.

Without fog:

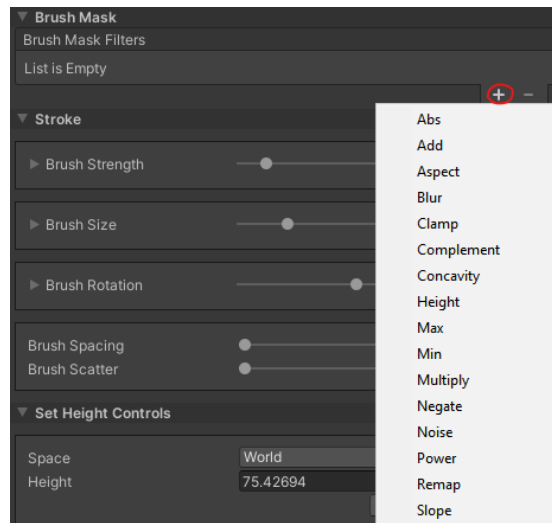


With fog:



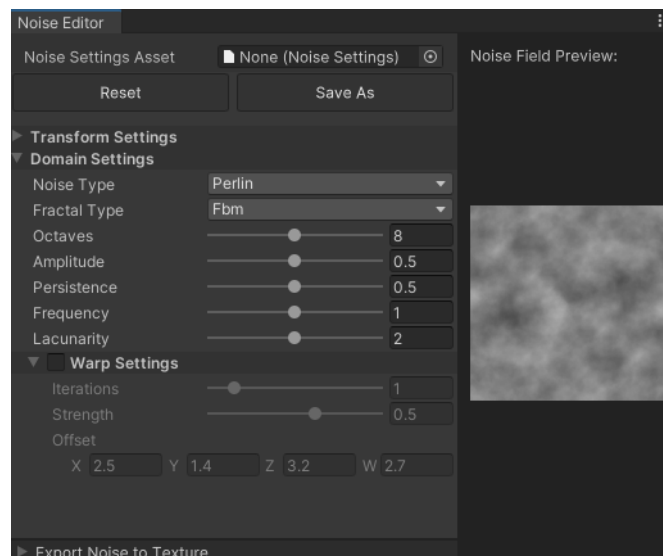
Advanced Brush Techniques

Adding Noise by Using Brush Mask Filters: By default, all brush assets have a 'Mask Texture', which defines the shape and strength of the brush (in combination with your chosen parameters under **Stroke**). To modify this brush even further (without creating your own), you can use a **Brush Mask Filter**.



Most of these are fairly straight forward, for example 'Add' simply adds to the pixel value in the brush mask (e.g. an 'Add' filter with a value of 0.1 will add 10% depth to all pixels in your brush mask). These are worth experimenting with, but by far the most in-depth and complicated filter is the **Noise** mask filter.

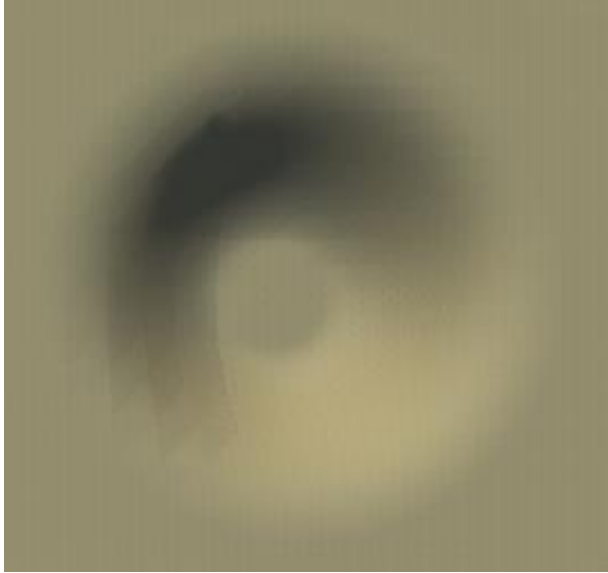
A **Noise** mask filter adds “randomness” to your existing brush mask by overlaying a “Noise Field”. To create some **Noise**, add a Noise brush mask filter from the drop down list as shown above, then click the **Edit** button in the **Brush Mask Filters** list. Note that you can also access the general 'Noise Editor' window by navigating to **Window > Terrain > Noise Height** in Unity's top menu bar (since noise assets can be used with many different functions in Unity).



As you tweak the settings in the noise editor it will be visualised as a texture in the Noise Field Preview on the right. Noise fields are fairly complicated and explaining all their settings here would be a bit out of scope, so for now simply tweak its parameters until your noise field preview

is appropriately distorted/messy. Once you're finished, click the **Save As** button, and you should be able to set your noise mask as a filter (in **Noise Editor > Noise Settings Asset**). Additionally, setting the brush mask filter to **Local** will produce the most obvious results.

No Noise:

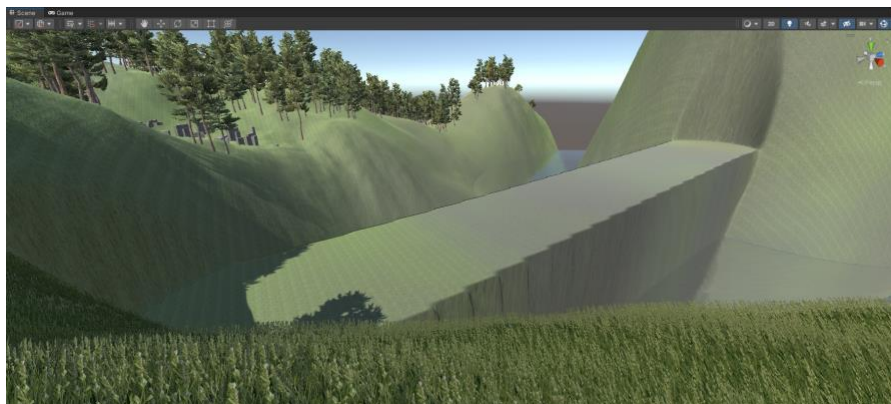


Same settings/brush, but with noise:



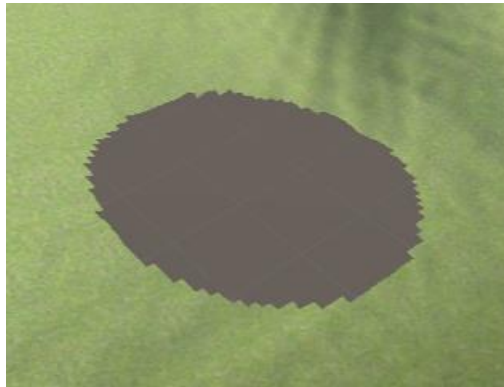
Modifying Terrain

Creating Bridges: There is also a tool that allows you to create rudimentary bridges out of the terrain, which you can find by selecting **Paint Terrain > Sculpt/Bridge**. Before making a bridge, it is worth highlighting that the width of your bridge is dependent on your **brush size**, so make sure you configure that first. To make a bridge, hold **Ctrl/Cmd** and **left click** to select the start point for your bridge on your terrain. Left click again (without holding **Ctrl/Cmd**) to select where the bridge ends.

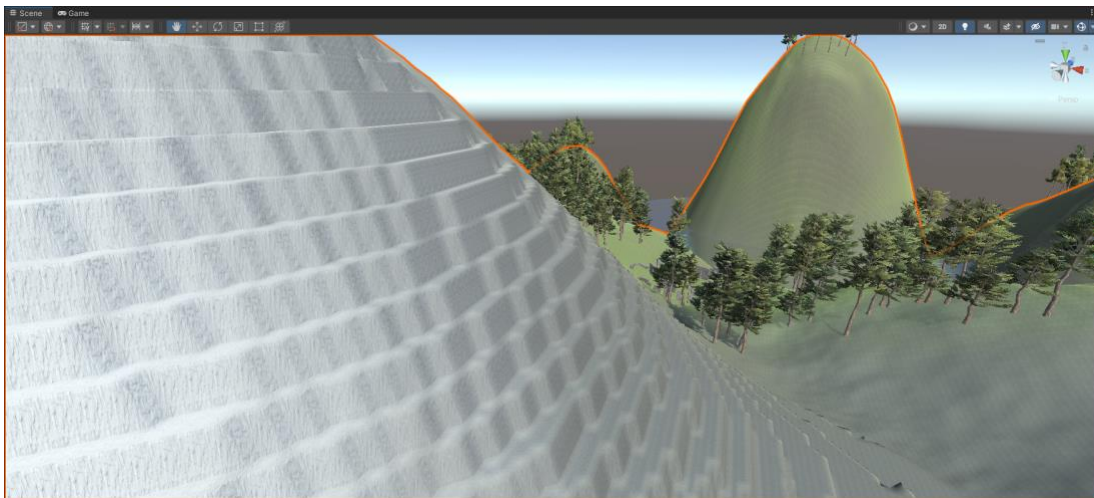


Painting Holes: If you wanted to have a separate geometric structure built into your terrain (like a hidden bunker or a tunnel underneath the ground connecting two different areas) then you'd need to cut a hole in your terrain to let players access it. Thankfully, there is an option for this.

Use **Paint Terrain > Paint Holes**, and then draw on your terrain using a brush with high opacity. This will create holes in your terrain that players can see and fall through.

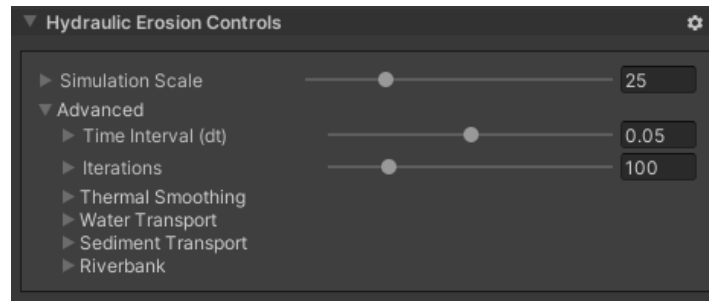


Terracing: You can “terrace” parts of your terrain to also help you create a sense of wind erosion, creating interesting flat “steps”, or simply flat surfaces. You could use this feature to recreate terraced erosion as seen on some desert mountains or features such as terraced rice paddies in the mountains of Vietnam. You can find this tool by selecting **Paint Terrain > Sculpt Terrace**.

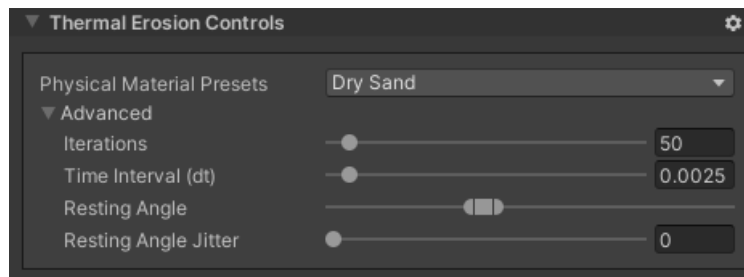


Additional Erosion Effects

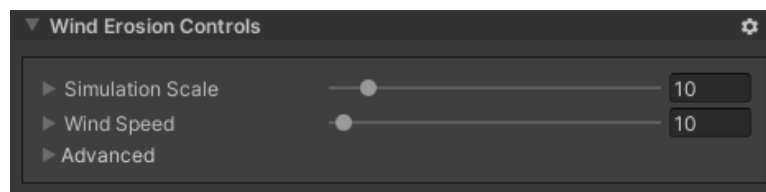
Hydraulic Erosion: A variation of the **raise or lower terrain** tool, **Paint Texture > Erosion/Hydraulic** smooths and modifies the terrain shape to simulate the effect of water transporting and redistributing sediment. This could be used to create some natural edges to a body of water, as well as some realistic underwater terrain.



Thermal Erosion: A variation of the **raise or lower terrain** tool, **Paint Texture > Erosion/Thermal** smooths and modifies the terrain shape to simulate several effects, such as weathering or temperature erosion. Presets for these simulations can be selected from the **Physics Material Presets** drop-down menu.



Wind Erosion: A variation of the **raise or lower terrain** tool, **Paint Texture > Erosion/Wind** smooths and modifies the terrain shape to simulate wind transporting and redistributing sediment. This is great for creating some natural looking deformations for high or steep cliffs, as well as sand dunes.



Show your demonstrator:

To demonstrate your understanding of this week's content to your prac demonstrator you might show:

- Your finished terrain, sculpted and dressed with textures, grass and trees
- Your terrain texture pallet, containing at least four (4) terrain layers
- Your Stonehenge and Player prefabs integrated into your scene
- At least five (5) implementations of "Other tools/features" to add additional content or detail to your scene
- Your frequent commit history in your GitHub repository