CHAPTER 4

Terrain Generation: Creating a Test Environment

One thing common with many game engines is a terrain editor. Not only can you sculpt your idea into being, but you can populate it with foliage from grass to bushes to trees via paint textures.

In Unity's full-fledged terrain editor, many things are handled for you, such as LOD (level of detail), distance culling, and animations. While this makes it extremely easy to get up and running around a terrain, as with anything that does a lot of the work for you, it also comes with limitations. For more in-depth information on the tree generator, go to the Reference Manual ➤ Terrain Engine Guide.

In this chapter, you will create a basic terrain to test more functionality than is strictly necessary for a classic point and click adventure game. During the process of developing the functionality and logic sequences for your game, you will quickly discover the need to keep everything in a fairly small area. Having to travel halfway across the terrain to find an object will get old fast. For that reason, you only need a smallish area to work in until everything not related directly to the topography is up and running.

Note You may import terrains created in DCC (digital content creation applications such as 3ds Max, Maya, Blender, etc.) programs to use as the ground, but you will not be able to paint trees, grasses, and other detail meshes that respond to wind and scene lighting. A compromise would be to create a height map from your external terrain, and then load it into a Unity terrain instead of painting the topography.

As long as you are still in test mode, go ahead and create a new scene from the Files menu.

1. Open Unity if you have not already done so.

It should open the BookProject project by default unless you have opened other projects since the last time you saved it. If it does not open the BookProject, from the File menu, choose Open Project and select it from the Open Project tab. You may also use the version available in the Chapter 4 folder from the book's downloadable package.

2. If the project opened with TestScene1 loaded, go to File ➤ New Scene.
3. From File ➤ Save Scene, save it as TerrainTest.
4. In the Project view, create a new folder and name it Scenes.

■ **Tip** During the project you will be creating and importing many assets. You will soon be buried under folder upon folder, so it is good practice to stay organized. Even those of us who are inclined to be organizationally impaired will soon be forced to admit that it is an important requirement of making even smaller games. As Unity's file system is based on dependencies, you must always remember to implement your housekeeping chores inside the Project view and never out in the OS's Finder or Explorer. Drag both the TestScene1 and TerrainTest scenes into the new folder.

5. From the Terrain menu, select Create Terrain, as shown in Figure 4-1.

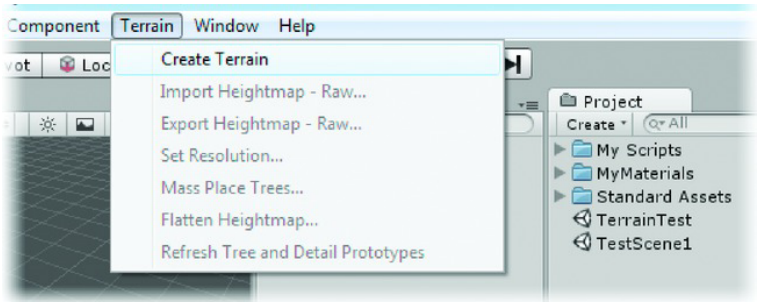


Figure 4-1. *Create Terrain in the Terrain menu*

You will see that a Terrain object has been added to the scene in both the Hierarchy and the Project views with one of its corners at 0,0,0, as shown in Figure 4-2.

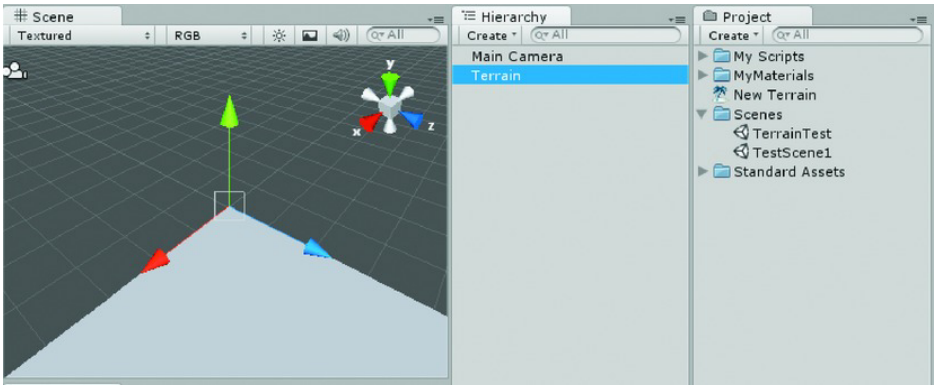


Figure 4-2. *The new terrain object with its corner at 0,0,0*

In the Inspector, with the Terrain object selected, you will see its properties and the tools available for refining it, as shown in Figure 4-3.

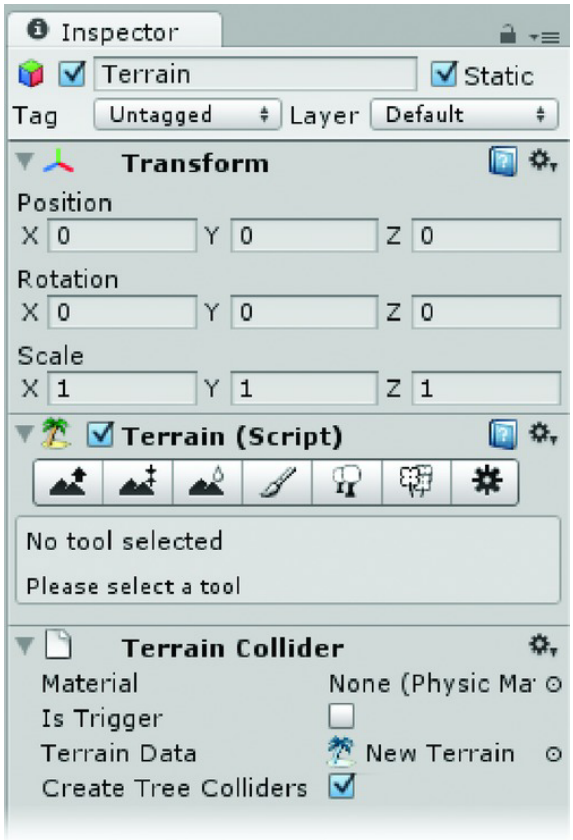


Figure 4-3. *The Terrain Tools in the Inspector*

6. Make sure you are using Default rather than Scene lighting in the Scene window.

■ **Tip** Scene lighting can be toggled off and on using the  button at the top of the scene view. It's a sure bet that you will often find yourself toggling this critter off and on throughout the project depending on the objects with which you are working.

Before experimenting with the tools, you need to set the other parameters of the terrain.

1. From the Terrain menu, select Set Heightmap resolution, as shown in Figure 4-4.

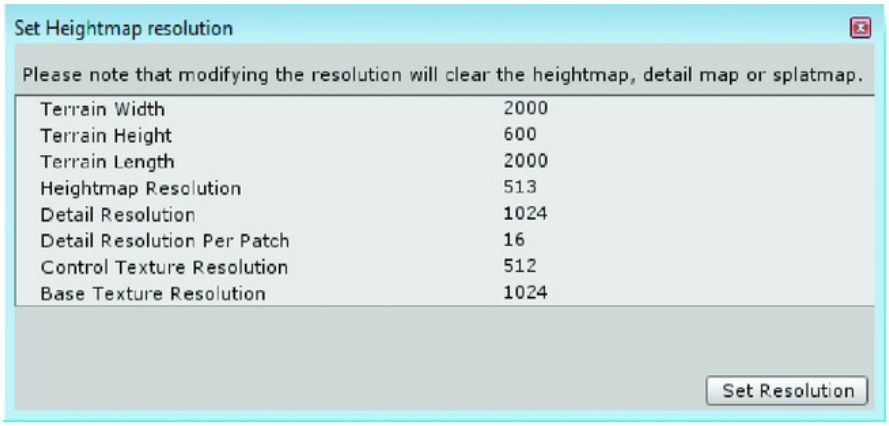


Figure 4-4. *The Set Heightmap Resolution dialog*

2. Take heed of the "Modifying the resolution will clear the height map, detail map or splat map" warning.

If you start creating your terrain, then decide to change its resolution, you will lose much of your modifications.

■ **Tip** A *splat map* is another name for the Terrain texture map/mask. Once you have started painting your terrain, you will be able to see that it is basically a gray-scale mask on drugs; it uses RGBA, and each color channel is used to mask a different texture.

---

3. Set the Terrain Width and Length to 500 meters and the Height to 200 meters.

This size is still overkill for what you will need during the development process, but it will allow for lots of practice with the terrain tools. Don't stress over making it perfect.

4. Click the Set Resolution button.

Because you can sculpt the terrain up or down, you will need to set the base height to allow for maximum depth where you may want to paint the terrain lower (see Figure 4-5).

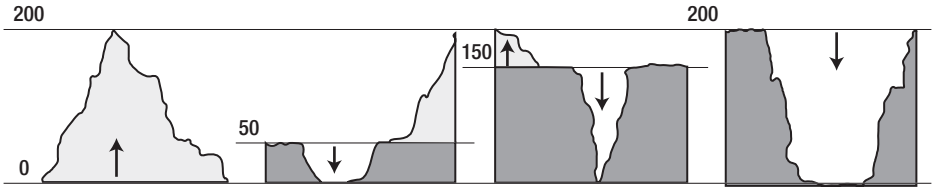


Figure 4-5. *Terrain base height shifted to allow painting of depressions for lakes, crevasses, etc.*

5. From the Terrain menu, select Flatten Heightmap.
6. Set the Height to 50 to give you 25 percent of your terrain's total height below sea level.
7. Click Flatten.

The terrain drops out of sight in the Scene window.

Before you sculpt, take a few minutes to experiment with the navigation tools; there are subtle differences in how they work with terrains because of the scale involved.

1. Select the Terrain in the Hierarchy view.
2. Position the cursor so it is *not* over the terrain in the Scene window and press the F key.

The viewport is zoomed all the way out so you can see the extents of the terrain object, as shown in Figure 4-6.

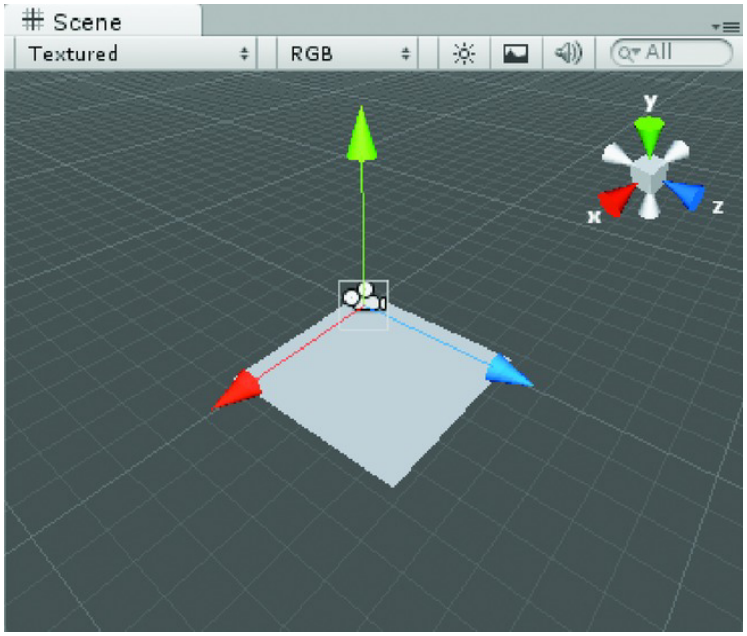


Figure 4-6. *The full terrain found or focused in the Scene view*

3. Position the cursor *over* the terrain near the right hand edge and press the F key

The view is zoomed to a more specific location, as you can see in Figure 4-7.

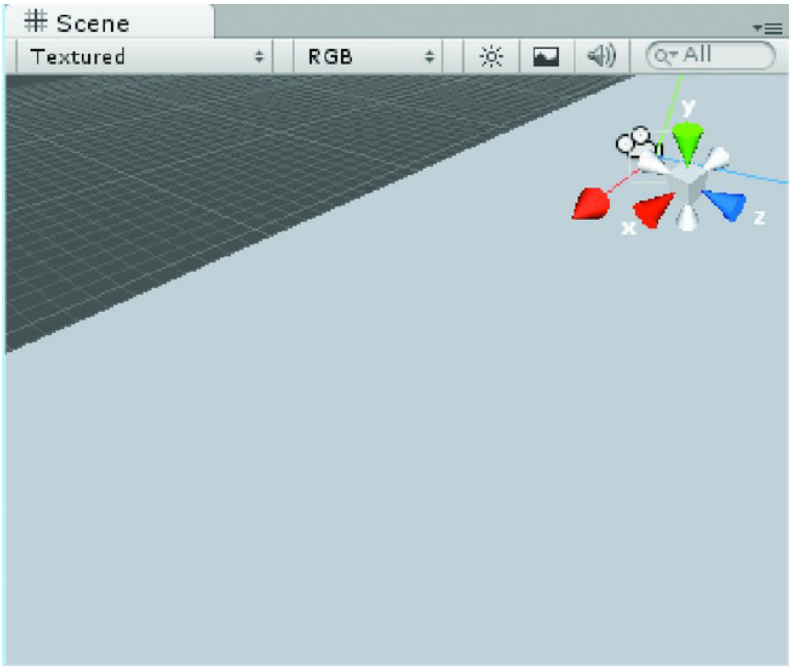


Figure 4-7. *Zooming to a specific area on the terrain*

Flythrough Scene Navigation

You can also do flythrough navigation. This becomes useful when creating terrains because it allows you to travel through the scene in the direction you are looking as if you were a bird or even a caped superhero.

1. Click and hold the right mouse button.
2. Use the WASD keys to move left/right forward/backward.
3. Use the Q and E keys to move up and down.
4. Hold the shift key down to move faster.

Topography

Time to do some damage to your terrain!

1. Zoom out so more of the terrain is visible.
2. In the Inspector, choose the first tool: Raise/Lower Terrain  .

Note the use instructions immediately beneath the Terrain Tools toolbar when you select a tool; it may be cryptic at first, but once you get underway it will start making sense.

3. Select a Brush.
4. Click and drag or use your favorite pressure pen to create mountains around the terrain, as shown in Figure 4-8.

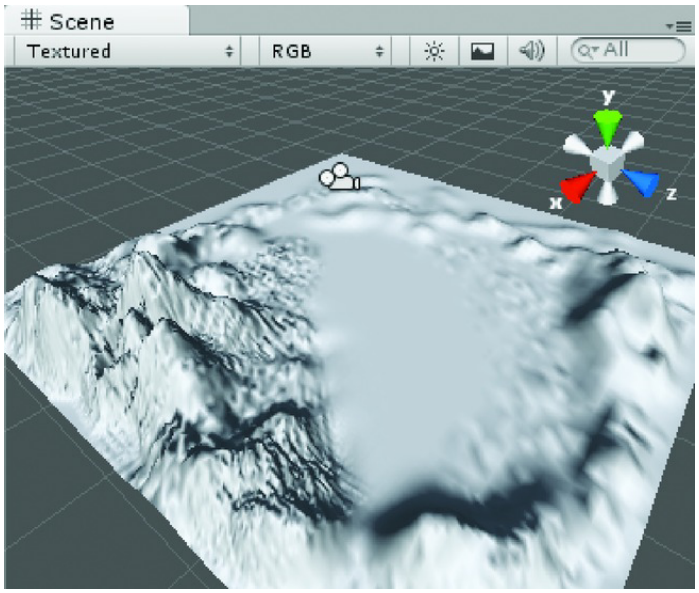


Figure 4-8. *Mountains created on the terrain*

■ **Tip** You can undo the last action with Ctrl+Z. Unity allows a large number of undos, but as with any program, you should not rely on going back too far. Inevitably, the action you want to get rid of will be one more than the limit.

Note that as you paint, if you have zoomed out to see the entire terrain, you will see the terrain at its highest resolution in an area around the cursor. One of the great benefits of using Unity's Terrain system is that it has a built-in LOD (Level of Detail) mechanism. As soon as you let go, the terrain reverts to its LOD resolution for the far distance (see Figure 4-9). While this can be frustrating, remember that it is just terrain, and the scale is such that once in game, it is likely not going to be crucial. You will be able to adjust the LOD distance later should you wish in the Terrain Settings Detail Distance.

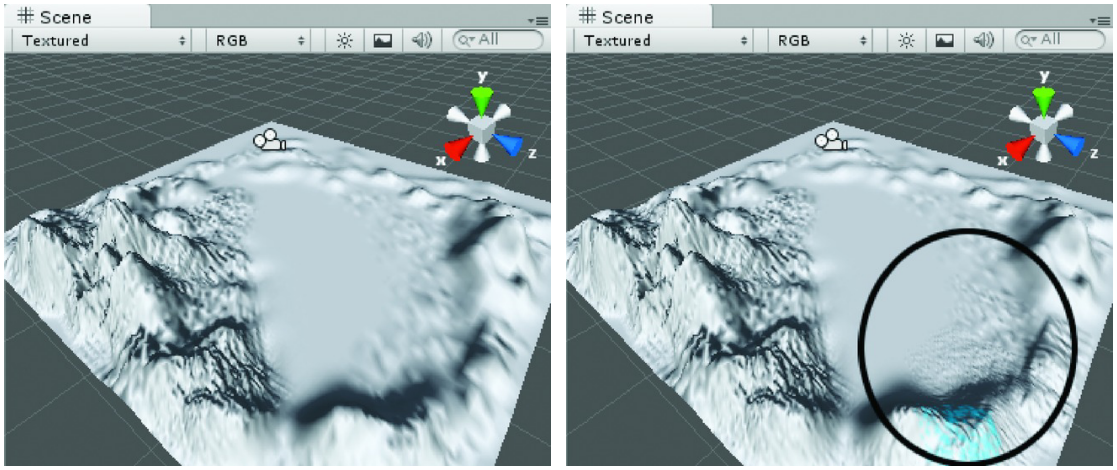


Figure 4-9. *True LOD resolution on the left and the temporary higher resolution or detail where you are painting on the right*

For now, just experiment with the Brush, Brush Size, and Opacity (strength). The brush size will be depicted as blue in the Scene window.

■ **Tip** With this tool, you may paint over existing topography to increase it up to the maximum set in the Set Resolution dialog (in your case, 200).

---

5. Hold the shift key down while painting to push the ground down, as shown in Figure 4-10.

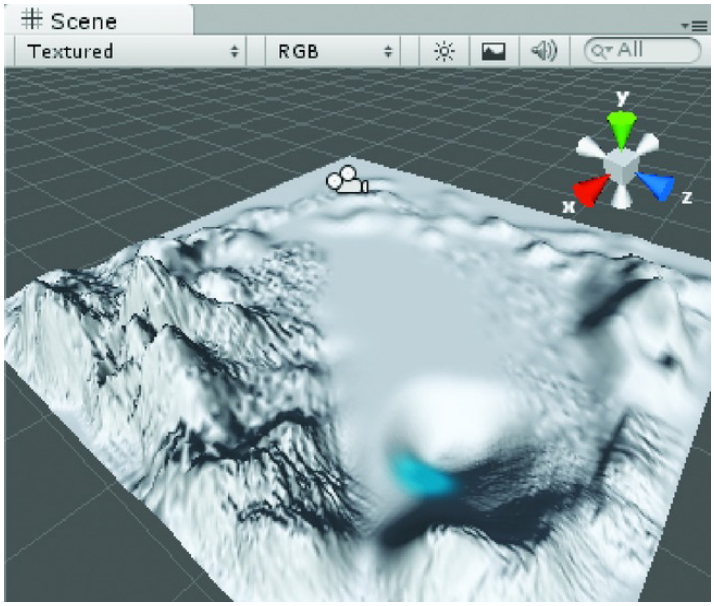


Figure 4-10. *A depression made in the terrain by holding the shift key down while painting*

Because you set the Flatten Height to 50, that is as deep as it will go down when you paint. You will deal with the hard edge later. You may also use the shift key to lower mountain peaks that are too high, but there is a better tool for that coming up.

The next tool to try is the Paint Height tool. This tool *specifies* a height to work to so the amount you set dictates whether you will be painting up or down. The most important thing about this tool is that it allows you to create flat areas at specific heights such as building pads, plateaus, mesas, sink holes, or anything else where the surface is superficially flat.

Remember that your default height is set to 50, so the height you set here will determine whether the painting adds or subtracts to the terrain.

6. Select the Paint Height tool  .
7. Set the Height to 40 and paint in a default flat area until the hole bottoms out at 40.
8. Now set the Height to 60 and paint an area so the height tops out at 60 (see Figure 4-11).

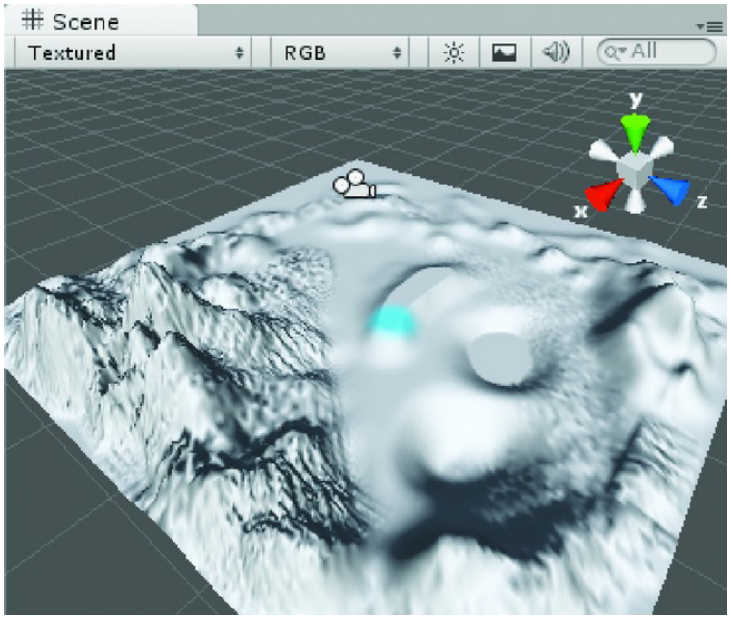


Figure 4-11. *The Paint Height results from 40 (right) and 60 (left)*

If you want to be extremely clever, write down the height of areas you plan to place structures upon at some point. This way, instead of manually moving the object up and down until it looks sort of okay, you can type in the Y position in the Inspector and move on. As anyone who has worked with 3D for any length of time can tell you, deciding when an object is "on the ground" can be challenging. If you are using Unity Pro with dynamic shadows, the task is somewhat easier, but shadows are not a silver bullet. Numbers are still your friend.

To get rid of the nasty looking sharp edges where the tools have bottomed out or caused a crease, you can use the next tool in the collection. Smooth Height relaxes the terrain to give it a more natural look or adds a bit of weathering to your jagged mountains.

9. Zoom in to the area where you painted the Height.
10. Select the Smooth Height tool  .
11. Smooth the hard edges left over from the Paint Height tool (see Figure 4-12).

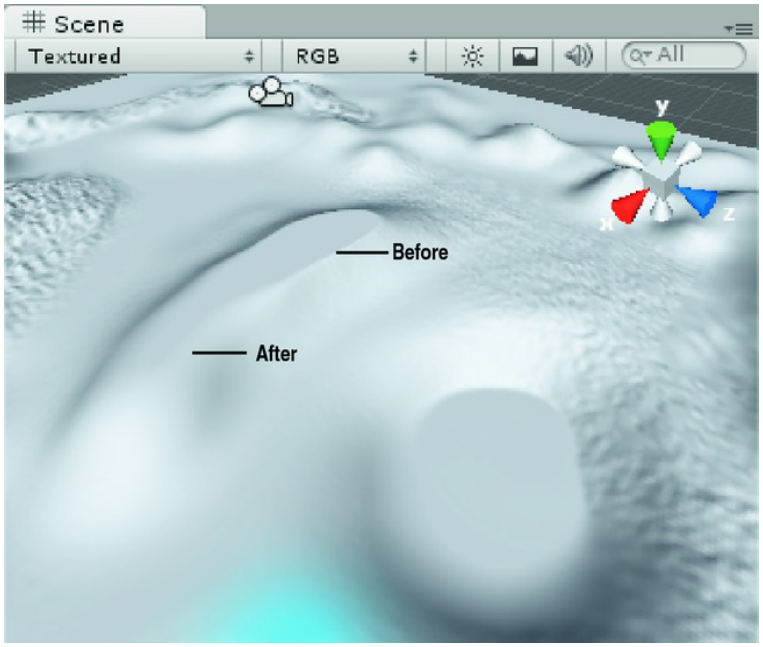


Figure 4-12. *The half-smoothed topology*

12. Experiment with the three Height tools until you are comfortable with your efforts. Remember that this is just a temporary terrain.
13. Save your scene and project.

Paint Texture

With the topography in place, it's time to put some textures on the terrain so it looks less like something from a government survey site (see Figure 4-13).

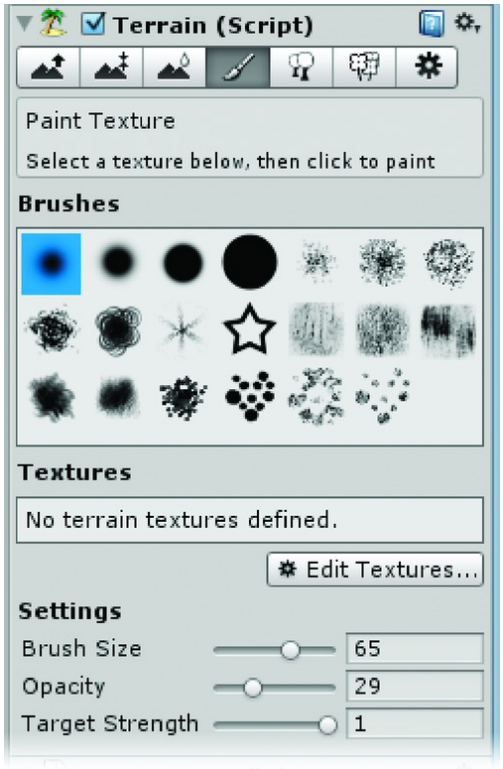


Figure 4-13. *Paint Texture and Edit Textures buttons*

1. Click the Paint Texture button  .

Before you can start painting the terrain, you will need to set up some textures. The first texture you load will fill the entire terrain, so choose a texture that will make the most sense as a base texture.

2. To load textures for painting, click the Edit Textures button and choose Add Texture (see Figure 4-14).

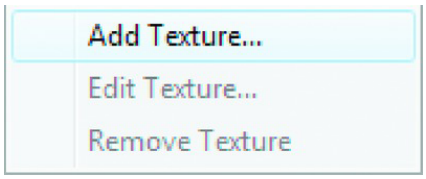


Figure 4-14. *Add Texture*

A dialog box appears and you are prompted to select a texture. You can do this by either clicking the browser icon to the right of Splat and choosing from the asset list (double-click it), or you can drag and drop a texture into it directly from the Project view. To finish the addition, you must pick the Add button at the bottom of the dialog (see Figure 4-15 and 4-16).

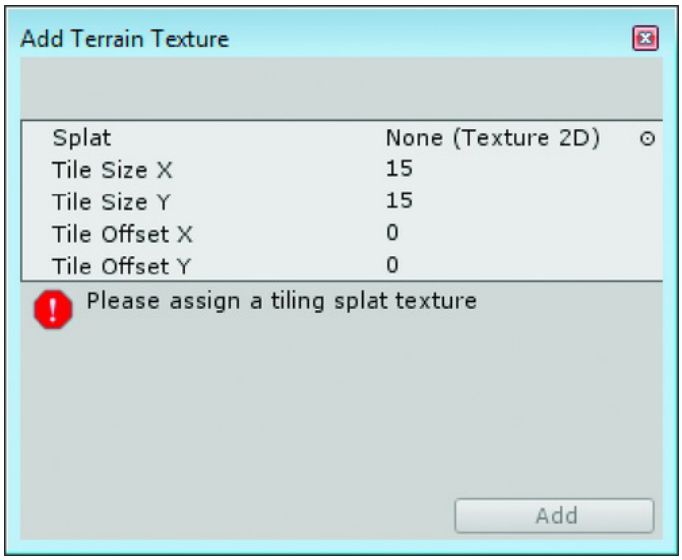


Figure 4-15. *Add Terrain Texture dialog*

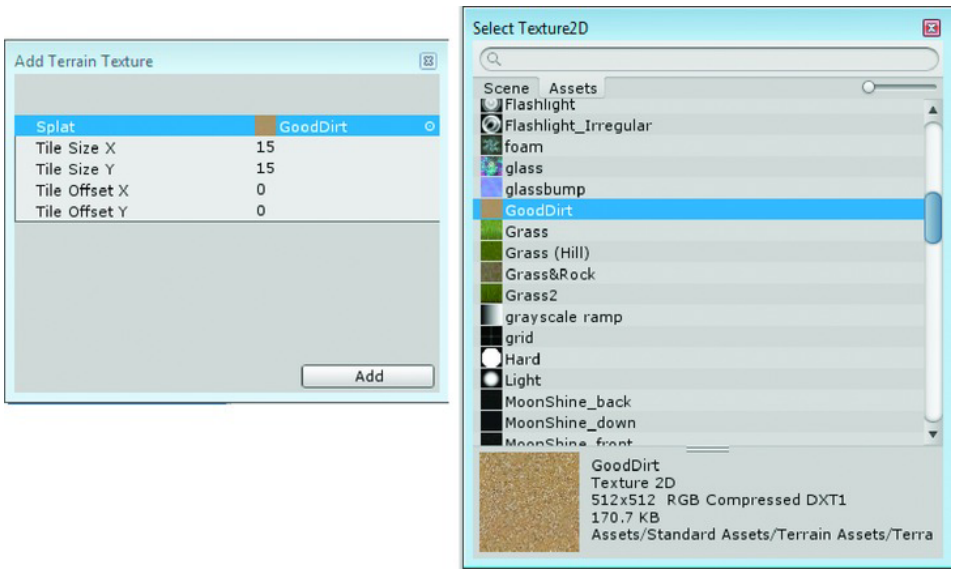


Figure 4-16. *Adding a texture*

Tiling parameters indicate the size of the tiling, not the number of tiles. Smaller numbers will increase the number of tiles while larger numbers will decrease the tiling. At this point, you probably have no idea of the scale, so you may adjust the tiling parameters later by clicking Edit Texture then selecting Edit Textures.

3. Add a couple more textures to your palette. The active texture has a blue highlight, as shown in Figure 4-17.

Figure 4-17. *More textures for the terrain: GoodDirt, Grass(Hill), Grass&Rock*

4. Using the various Brushes, paint the textures onto the terrain, as shown in Figure 4-18.

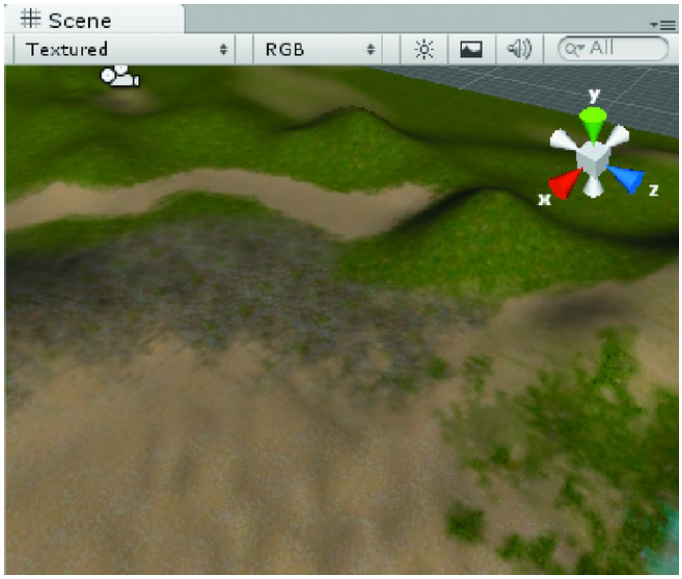


Figure 4-18. *The terrain painted*

As with any of the other terrain tools, you can access and adjust or change any of your terrain elements whenever you wish.

■ **Tip** Remember you can zoom to an area by moving the cursor to the spot you want, then pressing the F key.

To view the splat map created while you painted the various textures onto the terrain, you can look at the Terrain asset in the Project view.

5. Save the scene to make the Splat map show in the Project view.
6. Open the Terrain asset in the Project view.
7. Select the SplatAlpha 0 object.

The splat map shows in the inspector (see Figure 4-19). Note that it uses the Red, Green, Blue, and Alpha channels as a mask for the various textures.

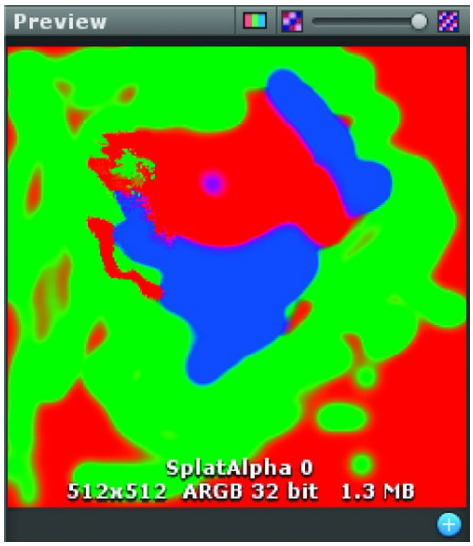


Figure 4-19. *The terrain's SplatMap in the Inspector*

Trees

An important part of the terrain editor is the Place Trees tool. While you could create and place your own trees, using the terrain editor to populate your scene provides several advantages. As you might expect, it also has some limitations.

Pros: Trees generated by the terrain editor may be set to animate to sway in the breeze. They also have automatic LOD; at a certain distance the full mesh tree is

swapped out for a billboarded plane with the tree's image, and at a far distance, the billboard object is not drawn at all, or *distance culled*. These distances can be set in the Inspector for the Place Trees tool, but more importantly, the engine will automatically reduce the distances to insure acceptable frame rate on slower machines. *Shadowing* is automatically generated for the trees. If the tree is on the shadowed side of a mountain, the trunk and leaves will be darkened.

Cons: Trees used for the Place Trees tool should not be over 2,000 polygons. Anything larger may not load properly in the terrain editor. Trees can't use more than two materials. The materials used are limited to two specialty shaders. If you don't use the Nature-Ambient Occlusion shaders in their materials, the lighting and the billboard will not be generated. In sparse areas, the transition between LOD models may be too noticeable.

■ **Rules** Imported trees used for the terrain editor should be no more than 2,000 polygons, can have no more than two materials, and those materials must use the Nature-Soft Occlusion shaders.

---

1. Select the Place Trees tool  shown in Figure 4-20.

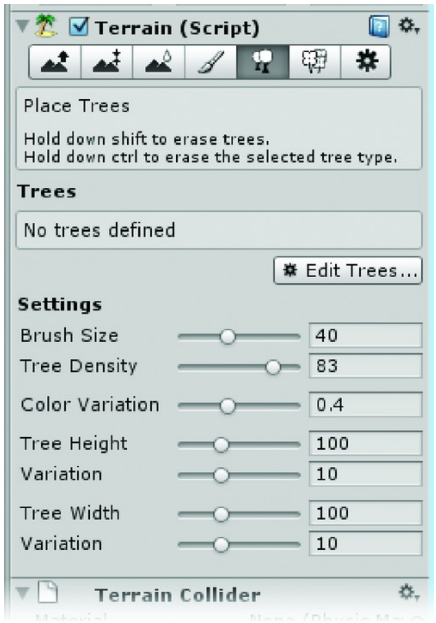


Figure 4-20. *The Place Trees tool*

Trees must be loaded before they can be used, just as textures must be loaded as well.

2. Click the Edit Trees button and choose Add Tree (see Figure 4-21).

Figure 4-21. *The Add Tree dialog*

3. In the dialog, click the Browser icon and select the Palm tree asset (see Figure 4-22) but do not click Add yet.

Figure 4-22. *The Palm tree asset*

Bend Factor is the parameter that animates the trees in the environment. Be aware that a little goes a long way. To see your trees bend at runtime you will need to add a WindZone object. You will be using WindZones later in the project, but feel free to experiment with one now.

4. Set the Bend Factor to 0.5.
5. Click Add.
6. Paint some trees on the terrain, experimenting with the various parameters, as shown in Figure 4-23.

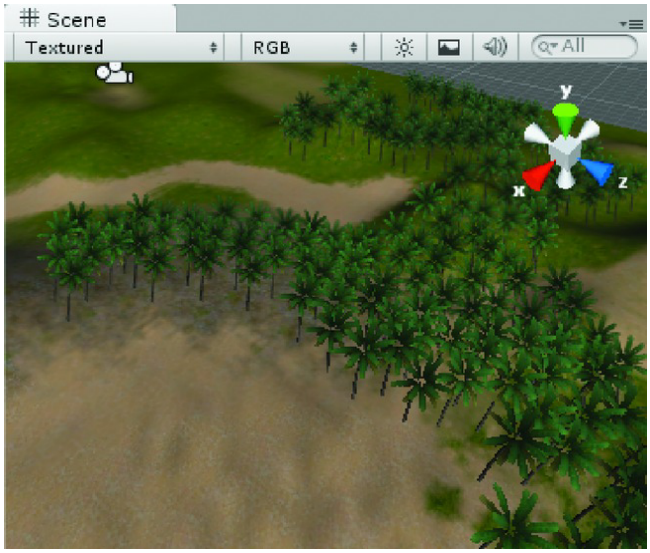


Figure 4-23. *The palms trees added to the terrain*

7. Navigate the viewport until you are close enough to see the mesh trees rather than just their billboarded LOD versions, as shown in Figure 4-24.

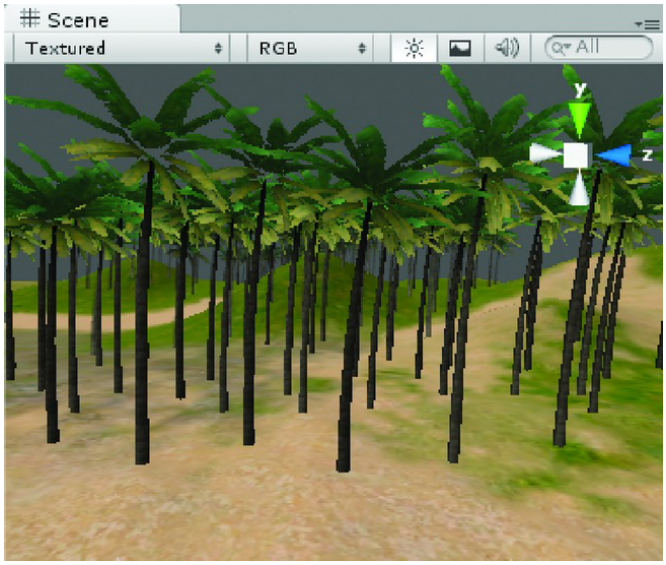


Figure 4-24. *A closer view of the palm trees*

To get a better view of the trees in the Game window, you need to do a few more things. Just as with the first cube experiments, you need to match the camera to the view you want, then you need to add a directional light. Additionally, because this is a real outdoor environment, you ought to add a sky.

1. Select the Main Camera.
2. From the GameObject menu, choose Align with View.

The camera matches the Scene view and the view appears in the Game window.

3. From the GameObject menu ➤ Create Other, create a Directional light.
4. Rotate the light until the palm trees are well lit, as shown in Figure 4-25.

Figure 4-25. *The trees lit and seen in the Game window*

5. Toggle the coordinate system to Global rather than Local and move the light up out of the way.

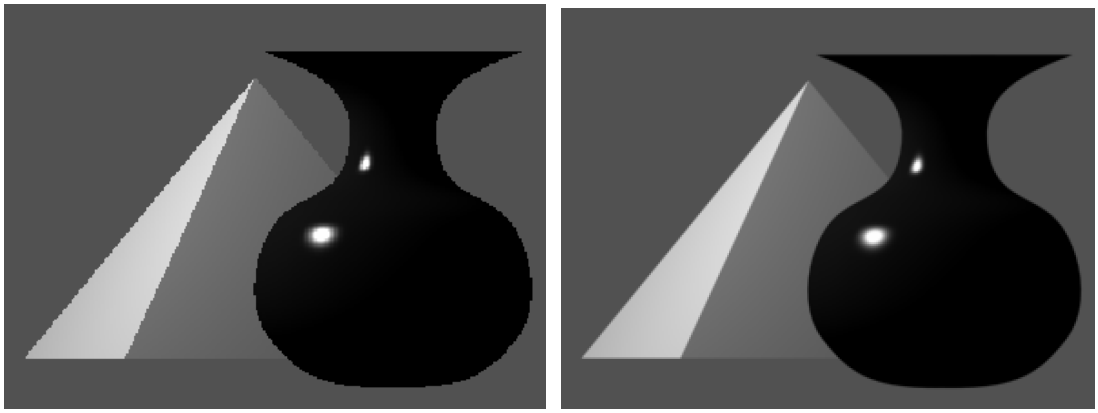
■ **Tip** You will find that it will be necessary to toggle the Scene lighting off and on throughout the project, depending what you are doing.

Now that you can see the trees more clearly in the Game window, you may notice that the trees are very rough looking. In the default settings of Good Editor Quality, anti-aliasing is turned off.

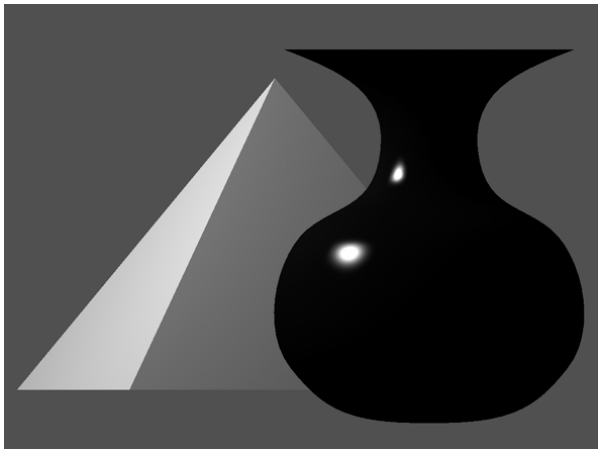
ANTI-ALIASING

When an object is drawn against an existing background or other objects, its edges will appear jagged. In render-type DCC applications, anti-aliasing is done on an object basis with a variety of anti-aliasing algorithms depending on the desired effect and style.

In real time, however, where speed is of the utmost importance, that type of anti-aliasing is too slow. Instead, it is performed by what is known as super-sampling where the scene is rendered out 2, 4, 6, or 8 times (depending on the capabilities of the graphics card) the size of the render window, then sized down to the final dimensions. The scaling produces a reasonable softening of the edges in a fraction of the time of traditional anti-aliasing.



Non-anti-aliased edges (the "jaggies") on the left and traditional anti-aliasing (fuzzy) on the right



Super-sampling(sharp but smooth)

6. From the Edit menu ➤ Project Settings, select Quality.

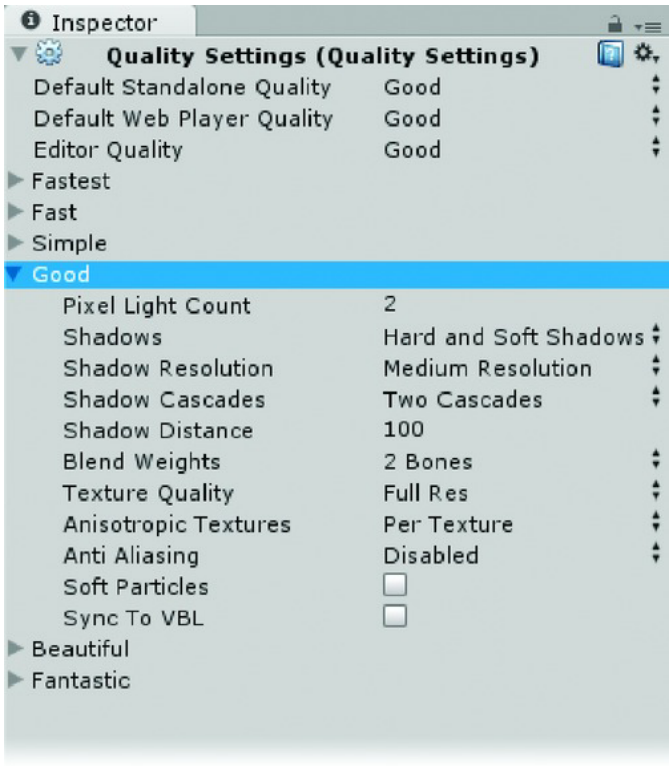


Figure 4-26. *The Quality settings*

7. Open the Good preset, as shown in Figure 4-26.

Note that anti-aliasing is turned off in this preset.

8. If your graphics card is fairly good, change the editor quality to Beautiful or Fantastic to see the difference, as shown in Figure 4-27.

■ **Tip** You can set the Default Standalone Qualities here, but the player will have the option to change them when first starting the game. Unity will automatically drop down to a setting that is supported by the player's graphics card.

Figure 4-27. *The scene with Fantastic quality turned on in the editor*

Sky

As long as you are taking a brief side track to improve the overall look of things, let's add a sky to the scene. Skies can be added to regular geometry as in the case of a skydome or, in Unity, you can employ a shader that uses a six-sided cubic map to do the job, as shown in Figure 4-28.

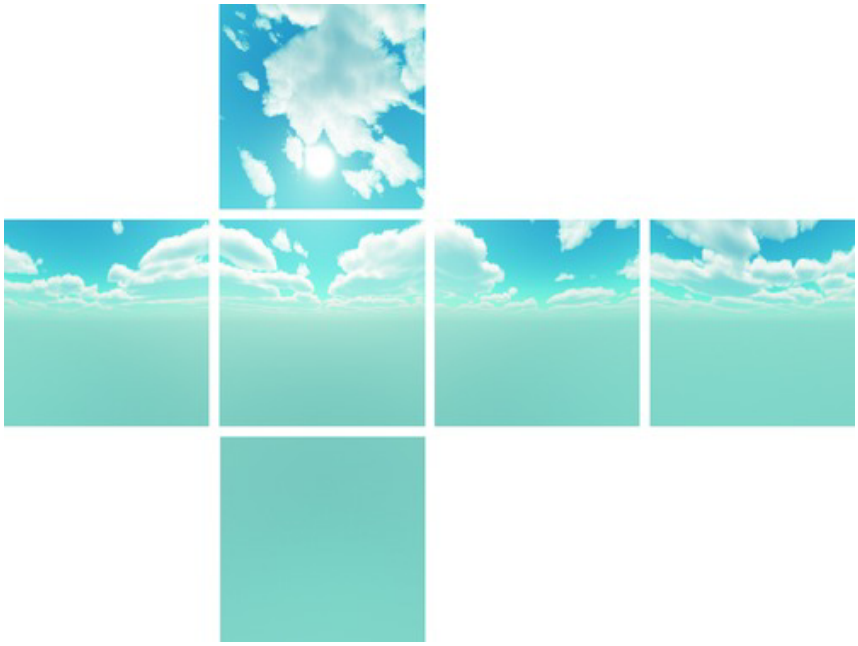


Figure 4-28. *The six images used for a cubic sky map*

A traditional skydome would have the advantage of working on systems that do not have much shader support, but because it generally uses simple spherical mapping, it's prone to distortion at the top. Take care to use an image that avoids the problem as much as possible. Skydomes also have the disadvantage of finite geometry. If they are too close, you will see where they intersect the regular scene geometry. If they are too far, they extend the camera clipping plane and increase the possibility of Z order fighting where two surfaces are too close to each other.

Because you are creating your game to play on traditional computers, you will go ahead and use the built-in shader type sky for your scene. Unity has two shader options for skies: one uses six separate maps and one that is able to use some preformatted cube maps. If you are using an external cube map, it may need to be broken into the six images using the RenderFX/Skybox. The advantage of component parts means you can make adjustments to size and quality via the import settings.

1. In the Project view, open the Standard Assets folder and open the Skyboxes folder.
2. Select the skybox info text asset and quickly read through it.

It gives information about skyboxes in Unity.

3. Click on each of the materials (sphere icons) to look at the choices.
4. Expand the Textures folder and open the Sunny3 folder.

This is where the images used to generate the materials reside and where you would go to re-import them using different settings.

5. Select the Sunny3 Skybox material, as shown in Figure 4-29.

Each of its component images are shown.

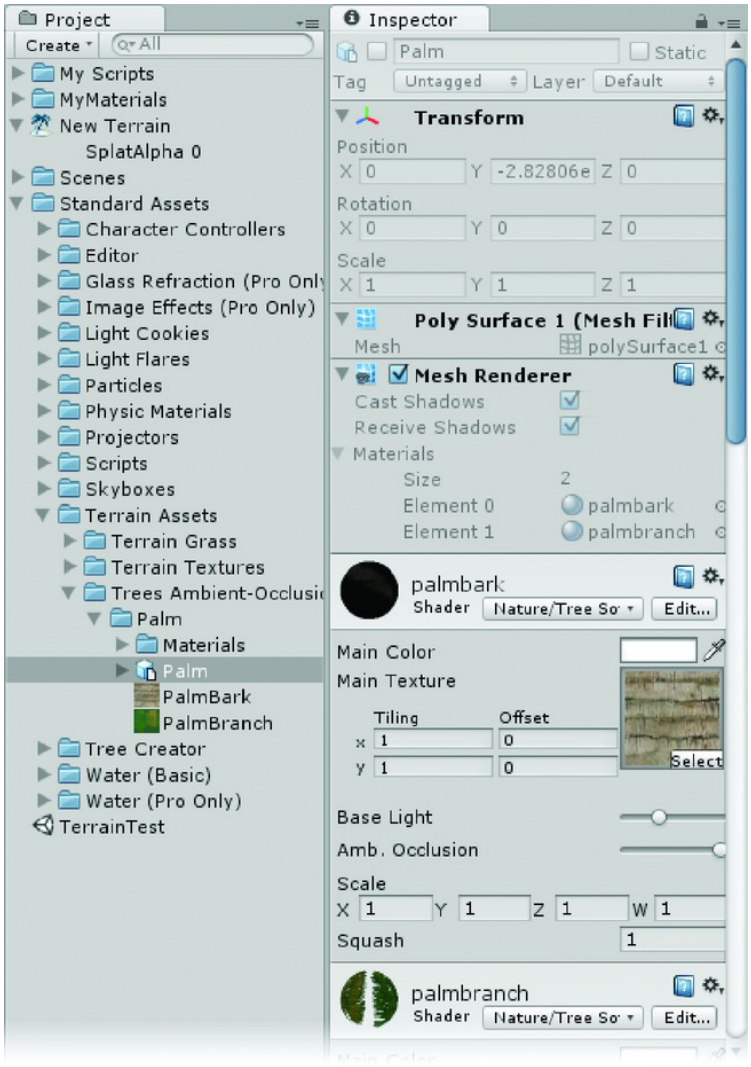


Figure 4-29. *The Sunny3 Skybox material in the Inspector*

To load a skybox into your scene, you will access Render Settings from the Edit menu.

■ **Tip** You can also load a skybox into a camera directly in the Component/Rendering menu.

6. From the Edit menu, select Render Settings.
7. Drag the Sunny3 Skybox texture into the Skybox Material slot, or pick the asset browser icon to the far right and select it from the list (see Figure 4-30).

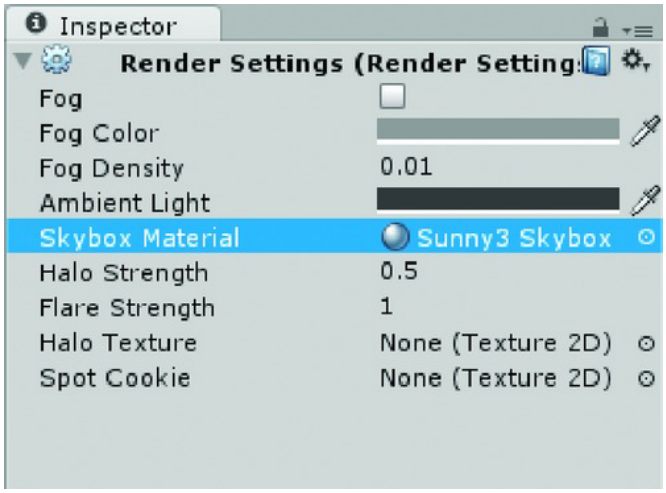


Figure 4-30. *The Sunny3 Skybox loaded into the Render Settings's Skybox Material setting*

Figure 4-31. *The skybox in the Game view*

■ **Tip** You can turn the skybox on in the Scene view by toggling on the Game Overlay  icon next to the Scene Lighting toggle. It will also turn on GUI objects, so it is usually left off once the scene has more content.

Back to the Trees

Now that your environment is looking better, you can continue with the Terrain editor.

1. Select the Terrain object in the Hierarchy view once again.
2. In the Inspector, select the Place Tree tool again.
3. Select the Palm tree and choose Edit Trees to change the Bend Factor.

As with the Edit Texture, edits affect parameters on all existing elements.

■ **Tip** Changes made to the Terrain editor during runtime will not be lost when you stop playback, so you can freely adjust parameters animation while getting instant feedback. Parameters such as scale will need to be changed in Edit mode to see the effects.

Once loaded, the billboard version of the tree (an image with an Alpha Channel) will show in the editor. The currently active tree's alpha area will appear gray and the selected tree outlined in blue when it is selected for use.

4. Paint the palm tree around your test terrain and experiment with the various parameters.

■ **Tip** In case you got a bit heavy handed with the brush, you can erase trees by holding down the shift key. Any tree, regardless of the currently selected tree will be removed. Holding down the Ctrl key will selectively remove the currently selected tree.

Packages

You are probably getting tired of palm trees by now. Let's see how to go about re-using assets from another scene.

Importing trees for use with the terrain involves more than just bringing in a textured mesh. Unity makes good use of what are called *prefabs*, prefabricated game objects that can contain all kinds of information, sub-objects, scripts, and other components. In the case of trees, they will need the special Nature shaders, and, if thicker than saplings, will probably need Colliders.

You've seen Colliders used before to "catch" mouse events. The more obvious use for Colliders, of course, is to prevent the player from going through what should be solid objects. To that end, larger trees will need Collider components.

1. Select the Palm tree in the Project view, as shown in Figure 4-32.

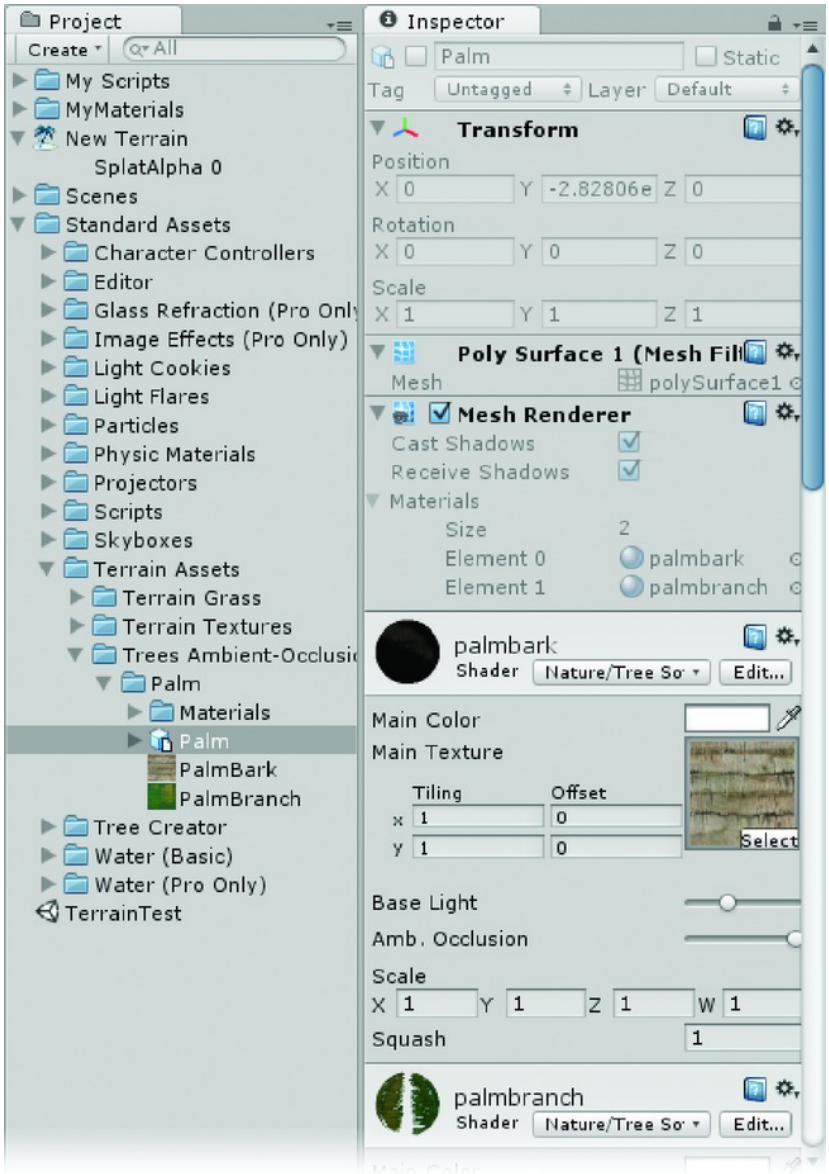


Figure 4-32. *The palm tree prefab in the Project view*

If you scroll through its components in the Inspector, you will see that it does not have a Collider. At runtime, you would be able to move through it. Testing for collision does use resources, so not all trees will or should have colliders. In the case of the palm trees, it will be less frustrating for the player if you let him go through them.

■ **Tip** An empty prefab is represented by a white cube. As soon as you add scripts or meshes to it, the cube becomes blue.

If you use Import Asset to import a tree, only the GameObject (the parent group), the mesh, and the materials and textures will come. The textures will need to be re-assigned and the shaders changed to the Nature shaders.

By copying the whole folder into the scene's Assets folder via the OS (Explorer for Windows) there will be less to repair, but you'll certainly lose some connections. The safest way to transfer assets from one scene to the other is to save them as packages.

Packages are collections of assets that, for the most part, can be imported into a project without losing important connections or dependencies. In the following section, you will be making use of an asset package created for this book. You may also wish to experiment with an asset package available for download from the Unity3D web site, TerrainAssets.unitypackage. This package contains several useful textures, trees, and other terrain-related assets.

■ **Tip** Check out `http://unity3d.com/support/resources/assets/` for Unity assets.

---

2. From the Assets menu, select Import Package.
3. Navigate to the Book Assets folder and select ExtraTerrainAssets.unitypackage.
4. Click Open.

The package is decompressed and you are presented with a dialog that allows you to select the assets you wish. They should all be checked by default, as shown in Figure 4-33.

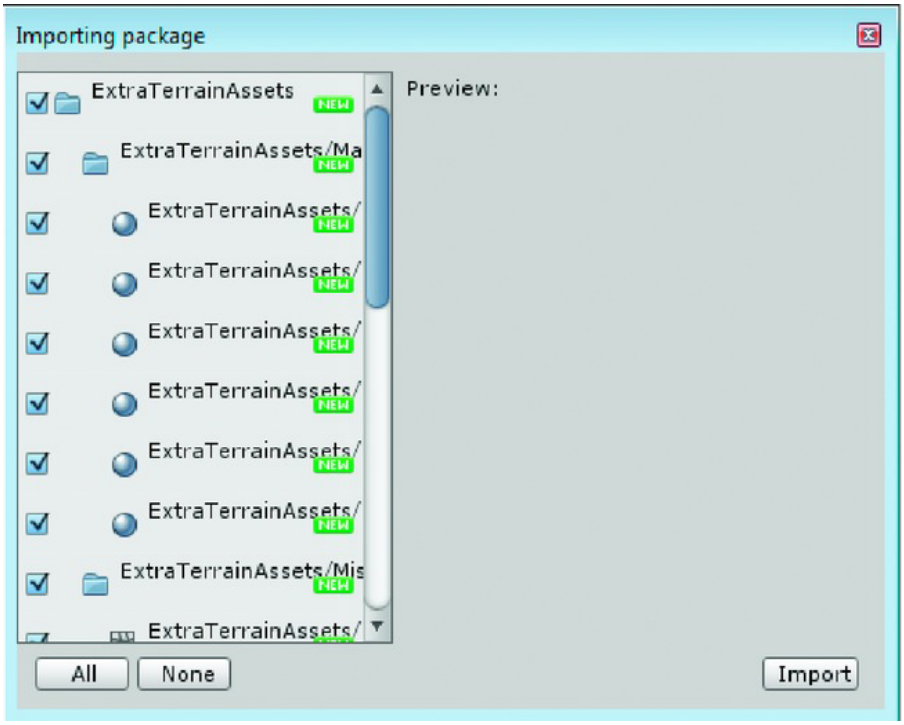


Figure 4-33. *The Importing Package dialog*

5. Click Import.
6. A new folder with the name of the package's original folder name is added to the Project view.
7. Open the ExtraTerrainAssets folder.
8. Open the Trees Ambient-Occlusion folder.

■ **Tip** In order for trees to cast and receive shadows, they must be in a folder named Trees Ambient-Occlusion and make use of the Nature shaders.

---

9. Select the BanyanOld prefab, as shown in Figure 4-34.

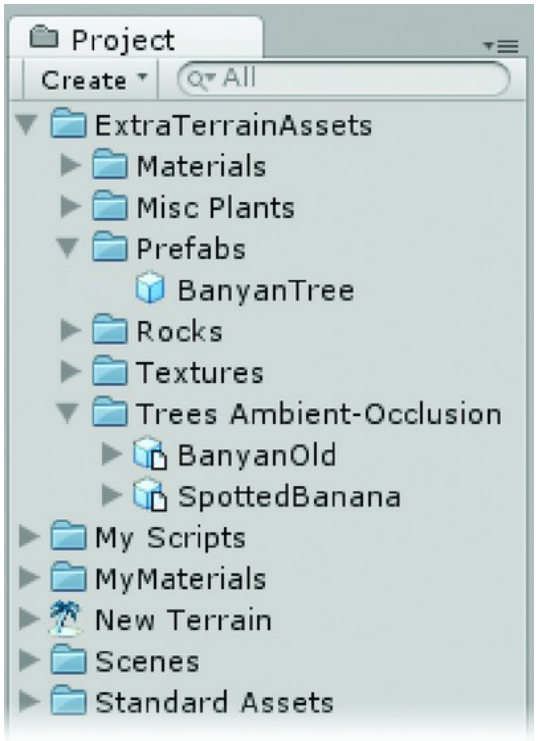


Figure 4-34. *Banyan prefab*

Note that it does not yet have a Collider. Since it is probably larger than a meter in diameter in real life, you really ought to prevent the player from going through it. Before you add the collider, you should make sure the import scale was correct. Let's load the new tree into the Terrain Editor and try it out to see how the scale looks.

1. Select the Terrain object in the Hierarchy view.
2. Activate the Place Trees tool in the Inspector.
3. Click Edit Trees and Add Tree.
4. Select the BanyanOld.
5. Choose an empty area near the palm trees (so you will be able to judge the scale) and paint some banyan trees on the terrain.
6. If you see no results, zoom in very close to the ground where you painted the trees.

As you paint the banyan, you may see tiny specks on the ground where you expected a great spreading tree (see Figure 4-35). You probably pictured them bigger!

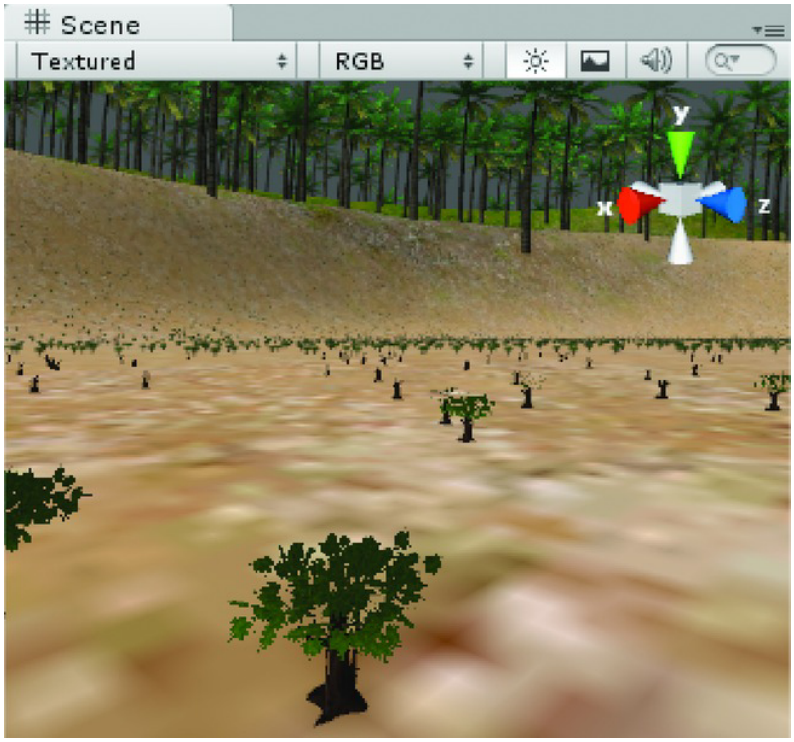


Figure 4-35. *Tiny trees*

On import, meshes are scaled to the last used Scale value and these will probably need their scale adjusted.

7. Select the Banyan Tree mesh in the Project view.

If the Scale Factor is 0.1, change its value to at least 1 and click the Apply button just above the Preview window near the bottom of the panel, as shown in Figure 4-36. The banyans grow dramatically.

■ **Tip** In a perfect world, you may be able to control the scale the imported assets are using in the original DCC program so that they always come in the same. It is probably more realistic to assume you will end up scrounging assets from several different sources and will need to adjust the Scale Factor. Fortunately, Unity is very good with scale adjustments, even skinned meshes.

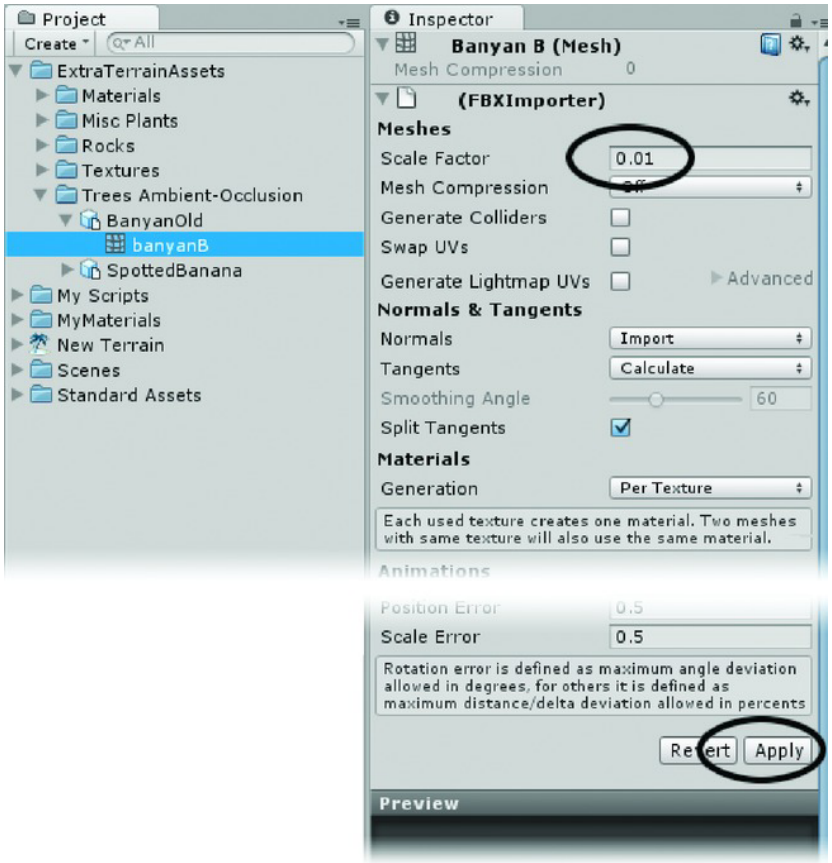


Figure 4-36. *The Apply button*

8. Rotate the view to exam the newly scaled trees.

Depending on your view, you may notice the billboarded versions are still small. Any time you make changes to assets used in the terrain editor, you need to update them to make sure the automatically generated features are updated as well.

9. In the Terrain menu, select Refresh Tree and Detail Prototypes.

■ **Tip** Refresh Tree and Detail Prototypes updates light maps, LOD billboards, materials, etc.

---

10. Use Ctrl and paint to remove the newly enlarged trees.

Before repainting the trees, you need to add the colliders. With the exception of the materials, imported assets cannot be directly affected in the Project view. To add a collider, you will need to instantiate a copy of the tree directly into the scene, add the collider, and then use it to create a prefab which can then be used in the Terrain editor.

1. Select BanyanOld in the Project view from the ExtraTerrainAssets folder's Trees Ambient-Occlusion folder.
2. Drag it into the Hierarchy view.
3. Use the F key to find it in the Scene view.
4. From the Component menu ➤ Physics, select Capsule Collider, as shown in Figure 4-37.

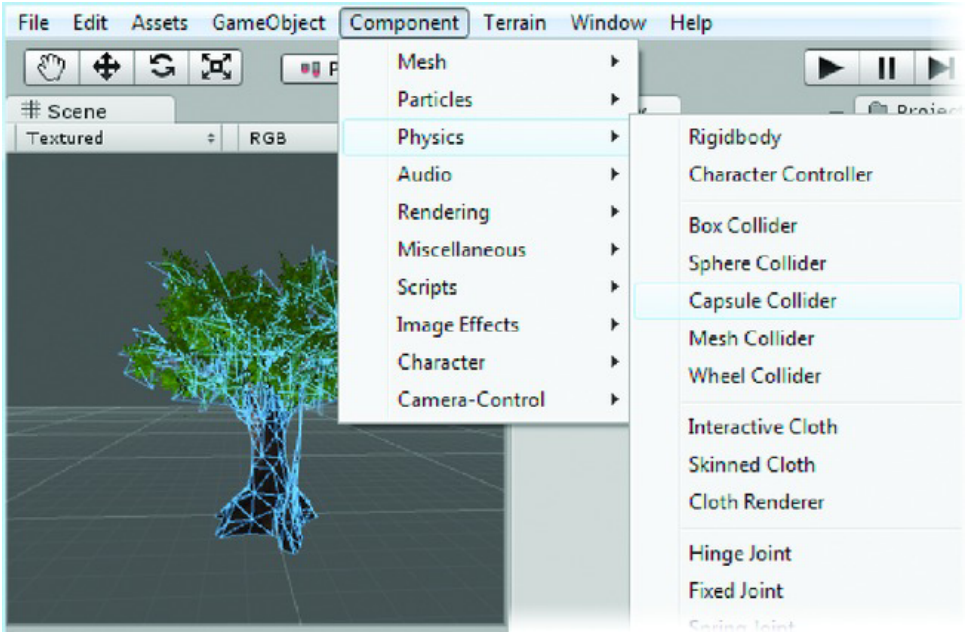


Figure 4-37. *Adding a Capsule Collider*

5. Click Add at the dialog that warns about losing the prefab, as shown in Figure 4-38.

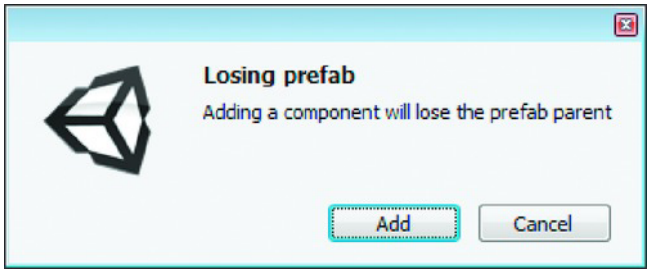


Figure 4-38. *The Losing Prefab dialog*

A collider is added to the tree encompassing the entire tree.

6. In the Inspector, set the collider's radius to about 2, and adjust the Center parameters to align the collider with the trunk and smaller trunk extension, as shown in Figure 4-39.

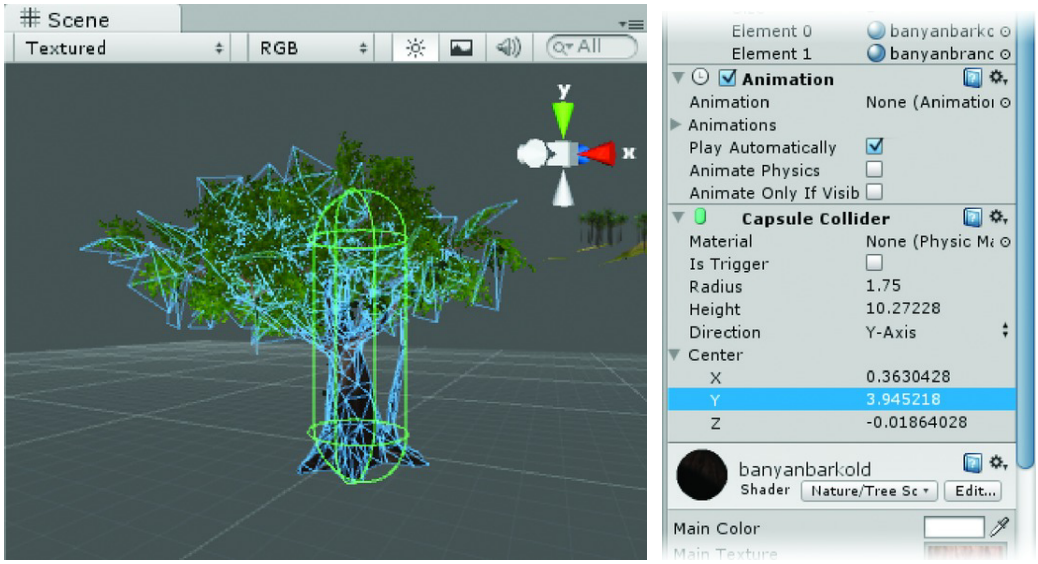


Figure 4-39. *The adjusted Collider in the Scene view and Inspector*

The tree is now ready to be "prefabbed."

7. Select the ExtraTerrainAssets folder.
8. Create a new folder in it and name it Prefabs.
9. Select the new folder.
10. Right-click folder, select Create ➤ Prefab, as shown in Figure 4-40.

■ **Tip** Alternatively, you can access the Create sub-menu from the Assets menu.

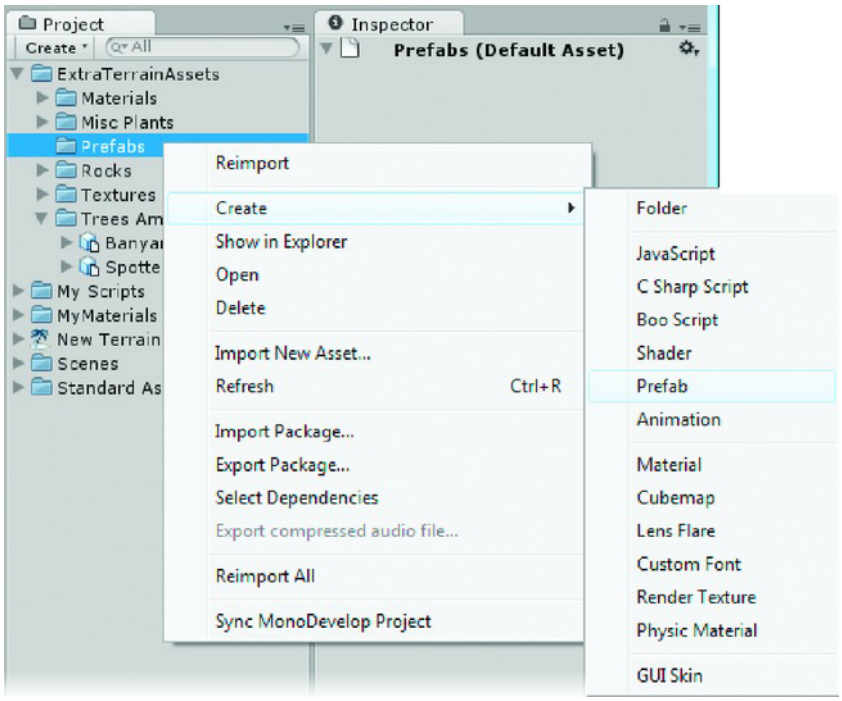


Figure 4-40. *Create Prefab*

11. Name it BanyanTree.
12. Drag the BanyanOld object from the Hierarchy into the new prefab.

Once the prefab has been created, you can delete the altered original from the Hierarchy view. The prefab does not need to be in the Trees Ambient-Occlusion folder because it references the original BanyanOld from there. The white cube icon for the prefab turns blue once an asset is added to it.

13. Select the BanyanOld in the Hierarchy view and delete it.
14. Select the Main Camera.
15. From the GameObject menu, use Align View to Selected to get back to where you were last planting trees.
16. Select the Terrain and activate the Place Tree tool in the Inspector.

Now that you have a replacement tree with a collider, you need to delete the currently loaded version of the Banyan.

17. Select the currently load banyan.
18. From the Edit Trees menu, select Remove Tree.

The previously painted banyans are removed.

19. From Edit Tree, select Add Tree and load the new BanyanTree prefab.
20. Adjust the Density and repaint them where you want them.

The bark appears quite dark on the banyans. Even considering that their dense foliage would be blocking a lot of light, you may wish to lighten the material a bit.

21. Select the BanyanBarkOld material in the Hierarchy view.
22. Experiment with the Base Light and Ambient Occlusion sliders (see Figure 4-41).
23. In the Terrain menu, select Refresh Tree and Detail Prototypes after each adjustment if you are not seeing the results.

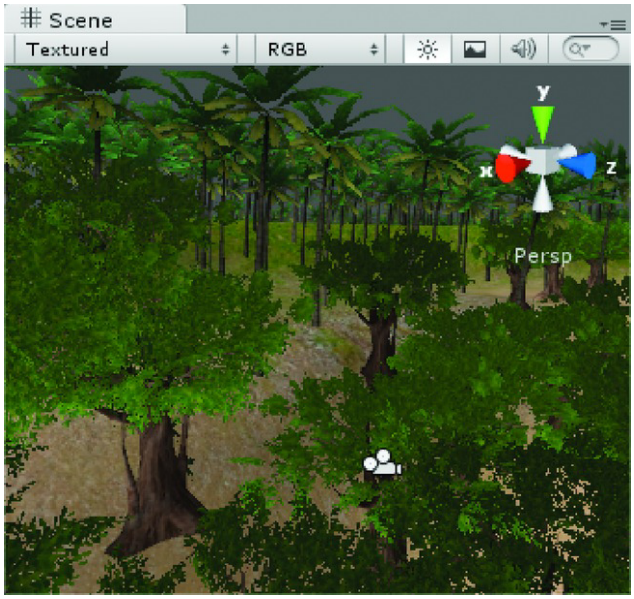


Figure 4-41. *The lightened bark*

Before you continue with the terrain tools, you will want to change the scale on the other objects in the package. As they were all created and exported with the same scale, they will all need to be set to 1.0.

24. Adjust the scale on the other imported meshes to 1.0.

Now that the tree is the correct size, you can set up the collider for it.

■ **Tip** If the trees need to be a little larger and you do not wish to adjust the import size further, you can increase the Tree Height in the Place Trees settings.

Sometimes it's quicker to remove trees from a densely populated area than to paint them manually. You can populate your entire terrain all at once to make things quicker. To mass place trees, from the Terrain menu, select Mass Place Trees. You may set the amount of trees. All trees currently loaded in the editor will be used, but you can't specify the percentages. Once populated, you can remove trees from unwanted areas by using the shift key.

Unity also has a Tree Generator that you will use later in the book to create a special tree for your final scene.

Paint Details

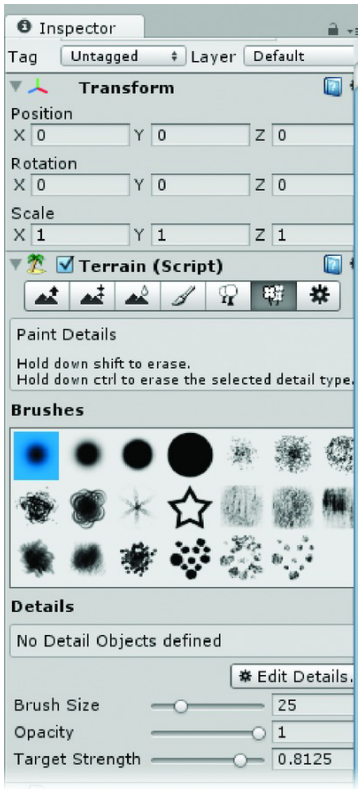


Figure 4-42. *Paint details*

This section lets you paint grasses, scrubs, rocks and other details (see Figure 4-42). There are two main types of details: grasses and meshes. Grasses are simple planes that are billboarded (they always face the camera) and can be set to animate to sway in the breeze. When you set up grasses, you will select only an alpha channel-using texture.

1. Select the Paint Details tool  .
2. Pick Edit Details and Add Grass Texture (Figure 4-43).

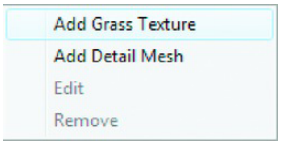


Figure 4-43. *Add Texture*

3. Select Grass, as shown in Figure 4-44.

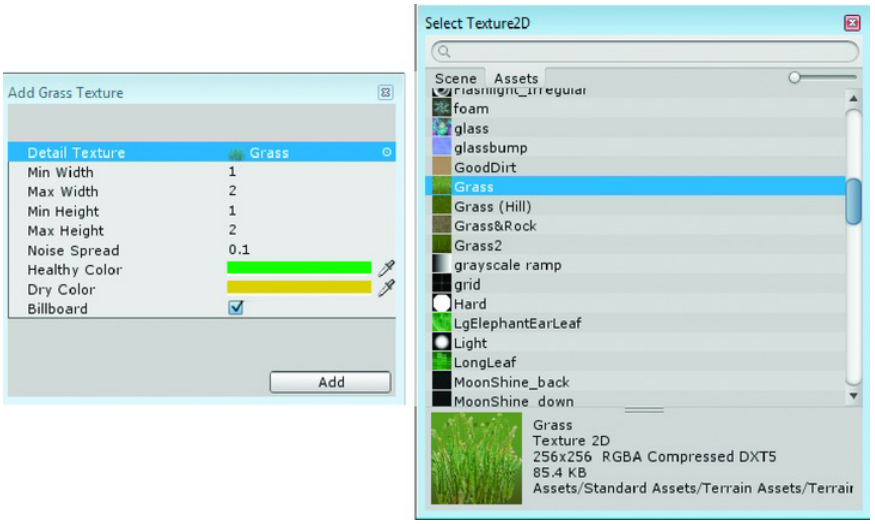


Figure 4-44. *The Grass asset*

4. Paint the grass around the scene, as shown in Figure 4-45.

Figure 4-45. *Grass added to the terrain*

Note the color variation provided by the Dry Color and Healthy Color textures.

■ **Tip** Take care when painting grasses and detail meshes. If you are too far away when you are painting, the grass will automatically be distant culled so you may not see the results of your work until you zoom in closer, at which point you may find you have a lot more than you planned.

Unlike trees that have a Bend Factor, it is assumed that grass will bend and sway in the wind. Bend Factor isn't what makes the trees sway back and forth, it just dictates how stiff or bendy the tree is. Wind speed, along with several other parameters, can be found in the Terrain Settings tool.

The remaining Detail type is a mesh object. This is where you will add 3D meshes such as rocks and mesh plants. The detail meshes are like grasses in that they have color ranges with Healthy and Dry. They can also be animated to move in the wind with Render Type. Grass will let them be affected by wind, Vertex Lit will keep them static. Noise Spread is the size variation.

1. Add the SpottedBanana as a detail mesh and set it to Grass Render Mode so it will sway in the wind.
2. Try adjusting the Height to positive and negative numbers.
3. Remove some of the saturation from the Healthy Color texture.
4. Turn the Opacity and Strength down.
5. Be sure to select the Spotted Banana in the Details palette before you start to paint.
6. Carefully click to place the plants, as shown in Figure 4-46.

You will probably need to adjust the material's Base Light and Ambient Occlusion. Remember to Refresh Tree and Detail Prototypes after each adjustment to see the results.

Figure 4-46. *A few more plants*

Next, add the rock as a static mesh in Details. Remember to change its Scale Factor to 1 from the FBXImporter section before you add the rock to the terrain editor.

1. Click Edit Details and select Add Detail Mesh this time (Figure 4-47).
2. Add the Rock by dragging it from the Rocks folder or selecting it from the browser.

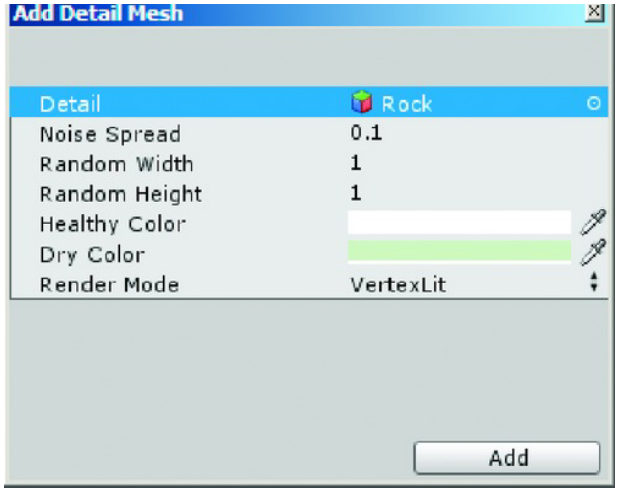


Figure 4-47. *The Rock asset*

3. Change the Healthy and Dry Colors to something more subtle.
4. Since rocks shouldn't sway in the wind, set the Render Type to Vertex Lit.
5. Adjust the Noise Spread to get a nice size variation on the rocks.

You may be tempted to populate you terrain with boulders using detail meshes.

6. Try increasing the height of your rocks, then zooming in and out (Figure 4-48).

Unlike the trees, detail meshes are not replaced with lower poly replacements at distance; they are merely distance culled.

7. Remove the boulders.
8. Edit the Noise Spread to something smaller. As with the other terrain assets, you will need to select the icon in the Inspector, select the Edit button, and choose Edit from the list of options.

Figure 4-48. *A good size for rocks that will be distance culled*

For large plants, you may be better off adding the mesh to the tree section. Do be aware that only one billboard tree image is made from the mesh, and trees all face the same direction, so asymmetrical meshes will not work well in sparsely populated areas. Also, detail meshes may not have colliders. As with all painted terrain assets, they will disappear at the culling distance so think carefully when deciding what to place manually and what to paint.

Creating Your Own Terrain Assets

Assuming you may want to try your hand at creating your own terrain assets, there are a few things to keep in mind. As mentioned before, trees and plants require the Nature shaders, Tree Soft Occlusion Bark, and Tree Soft Occlusion Leaves to receive shadows and color tints. The shadowing color on trees is created with vertex color, as are the Healthy/Dry variations, and is added to the color set by the texture and shader Main Color.

Besides keeping the tree under 2,000 faces, it must also use no more than two materials. Careful unwrapping and texturing can do much to make up for this limitation. The banyan tree, using a derivation from the TerrainAssets package, has a texture with full leaves, a bare branch, and a vine of some sort all using the alpha channel.

If you create your assets in 3DS Max or another application that uses Z as up rather than Y, you will need to orient the objects so they appear up in the top viewport (see Figure 4-49).

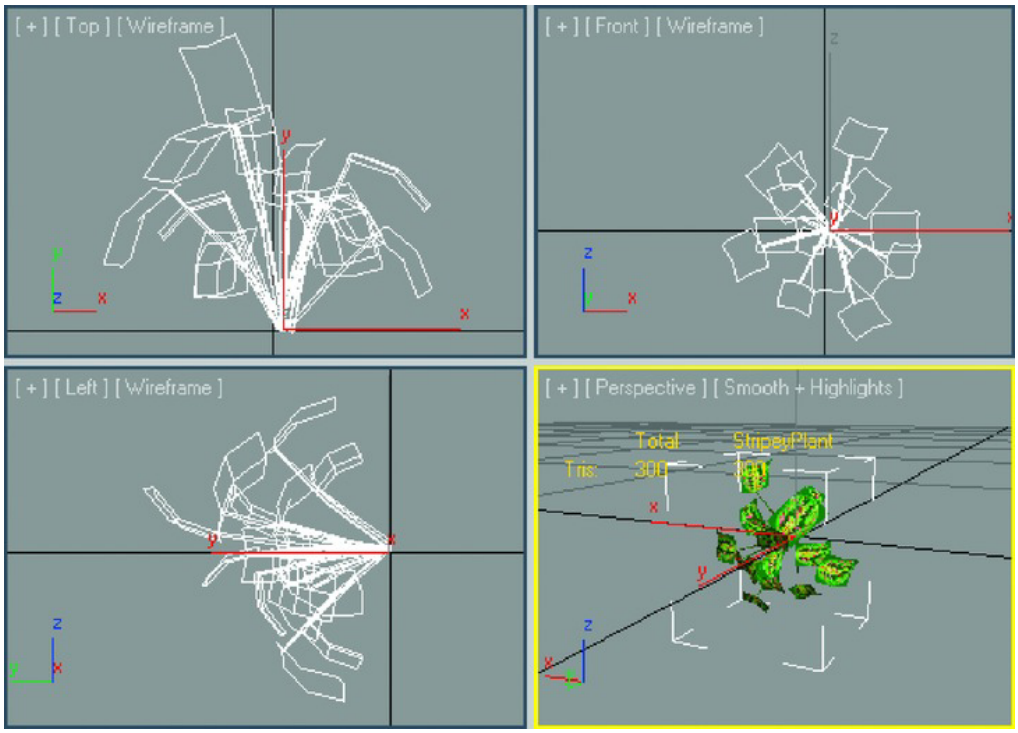


Figure 4-49. *Orientation in 3DS Max*

Unlike the regular imports that are able to convert their native orientation, terrain assets must be correct before import. To insure the transform is baked in, create a plane in the top viewport, attach (from the Edit Mesh modifier) the terrain asset to it, then remove the plane with sub-object element, as shown in Figure 4-50.

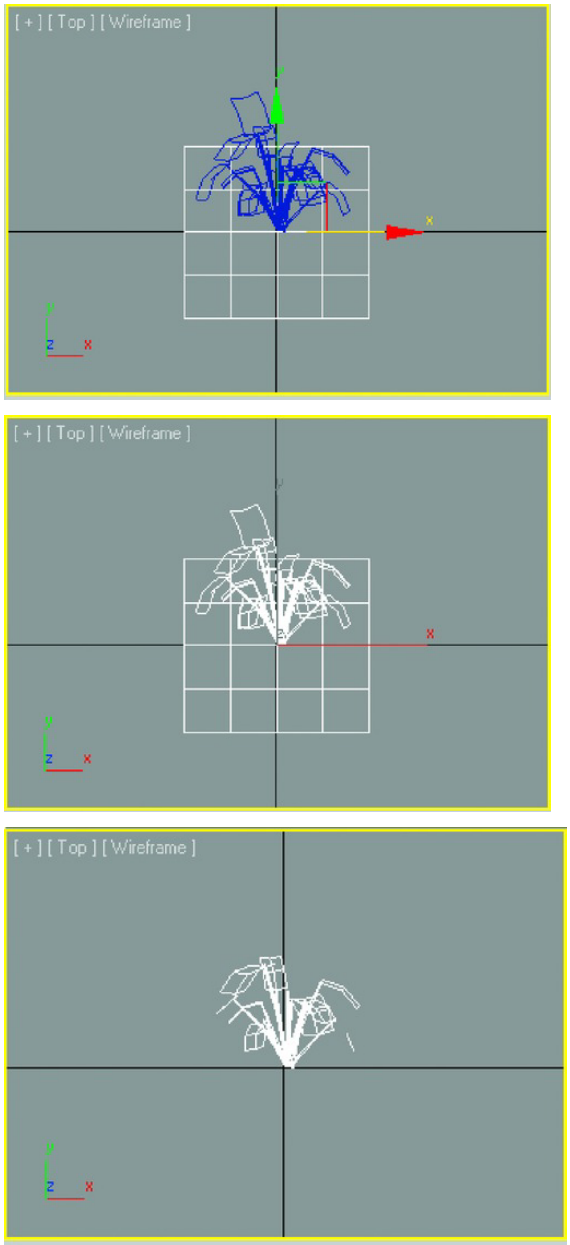


Figure 4-50. *Getting a clean transform matrix*

1. Create a plane in the Top viewport.

2. Collapse the plane to an editable mesh and attach the plant to it.
3. In sub-object element, select the plane and delete it.
4. Name the object and export.

■ **Tip** Unity will create a material for the mesh on import using the name of the Diffuse texture. If no texture is used, it will create a generic material using the name of the file and the object name.

Export vs. Use Native Format

If you are not using a versioning software such as SVN or Unity's Asset Server, you may prefer to export as FBX format so you always have access to earlier versions of your files and can use the FBX file on any machine without having the original application installed.

Versioning software allows you to save directly to the Unity project, but keeps previous versions in case you need to revert to an earlier version.

Terrain Settings

The last icon on the Terrain Editor deals with Terrain settings. This the place where you can change distances for LOD and distance culling for trees and details as well as adjust wind strength.

1. Select the Terrain object.
2. Select the Terrain Settings tool in the Inspector  .
3. Click Play.
4. Adjust the wind settings, speed, size, and bending until it looks good.

You may wish to change some of the larger plants to Vertex Lit instead of Grass to prevent them from bending.

■ **Caution!** Detail Meshes that are Vertex Lit will *not* be able to use their alpha channels!

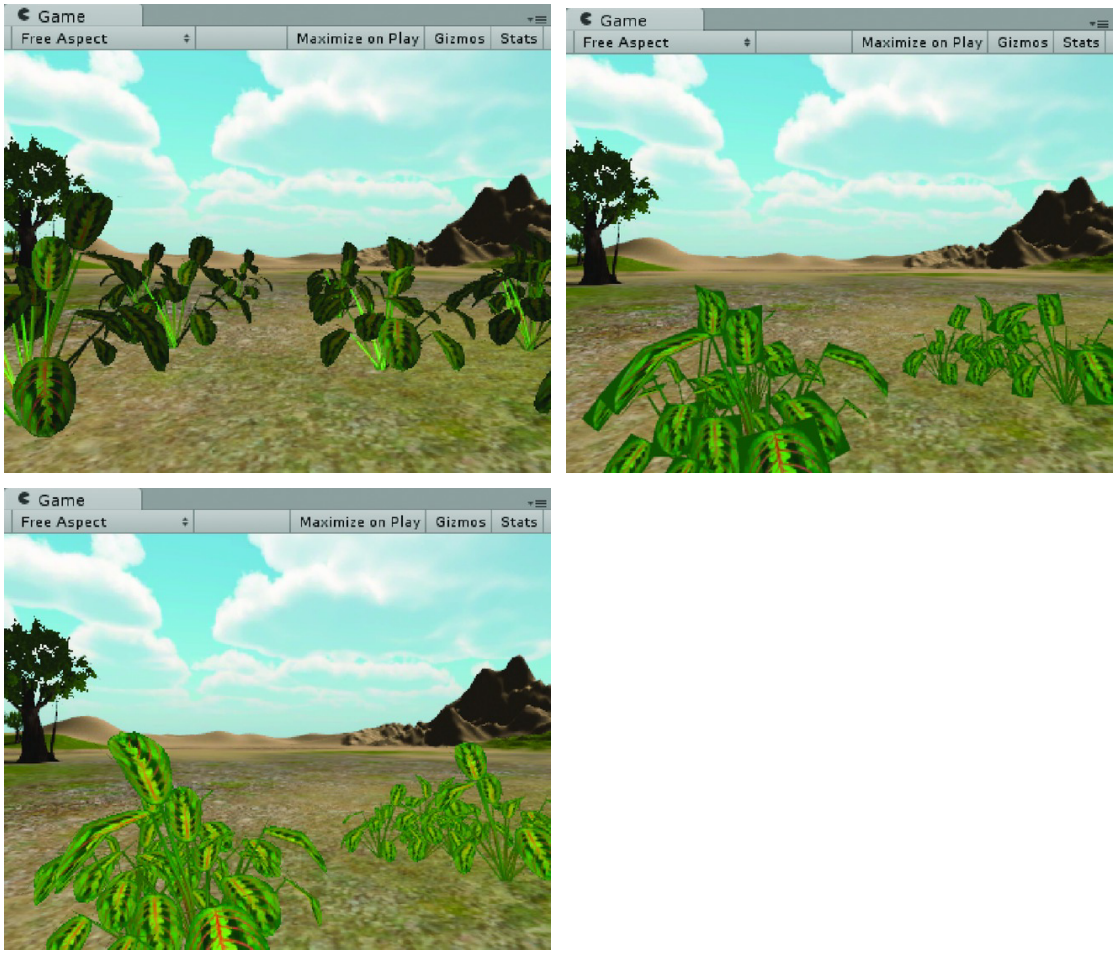


Figure 4-51. *The same plant as a tree (upper left), a Vertex Lit Detail Mesh (upper right), and a Grass Detail Mesh (lower)*

In Figure 4-51, the image in the upper left shows the Tree version being affected by scene lighting and with its own Bend settings. As you can see in the image in the upper right, the Vertex Lit Detail Mesh will not use alpha channel opacity. It is not affected by scene lighting except as a generalized vertex tinting: if the plant is in shadow, the vertex tint is darkened. Finally, the image in the lower middle shows the Grass Detail Mesh. It is not affected by scene lighting except as a generalized vertex tinting; if the plant is in shadow, the vertex tint is darkened.

Tree Bending is set individually for each loaded tree type. Detail Mesh Grass bending is controlled generically in the Terrain Settings.

■ **Tip** Terrain adjustment made during Play mode will be retained. You will need to use undo to remove unwanted adjustments may during Play mode.

Shadows

Next, let's bake some shadows into the terrain. Since terrains can be quite large, generating lightmaps for them can take a lot of time and use a lot of texture memory. *Baked* shadows can add a lot of definition to your scene but don't allow for objects that are dynamic or will animate. Dynamic shadows will track animated objects but lack the subtlety that can be obtained with large numbers of carefully set lights and global illumination (where light bounces are calculated).

If you are using Unity Pro, you can have the best of both. You can bake the shadows into the terrain and have dynamic shadows as well with the dual lightmap system. Two lightmaps are created; one with all shadows included, Far, and one with only the indirect lighting shadows, Near. Within a specified range, the Shadow Distance, the Near map is used and realtime shadows calculated. Beyond that range, the Far shadow map is used (see Figure 4-52).

Figure 4-52. *No shadows (left) and dynamic, realtime shadows (right)*

In your test scene, you have only one light, the Directional light, and its Lightmapping parameter is set to Auto, so you should be good to go. Let's start with a brief look at the Lightmapping dialog.

1. Select the Terrain object.
2. From the Window menu, select Lightmapping, as shown in Figure 4-53.

Figure 4-53. *Beast startup with the Terrain object selected*

Figure 4-54. *The Lightmap Display in the lower right corner of the Scene window; the default Shadow distance will vary according to your Quality settings.*

The first thing of note is the Static check box. Because this is a terrain and will not be animating, it is automatically set to Static. Later, when you import assets that are not used with the Terrain editor, you will need to remember to set the objects as Static or not, depending on whether they will be animating. Only Static objects will be included in lightmapping

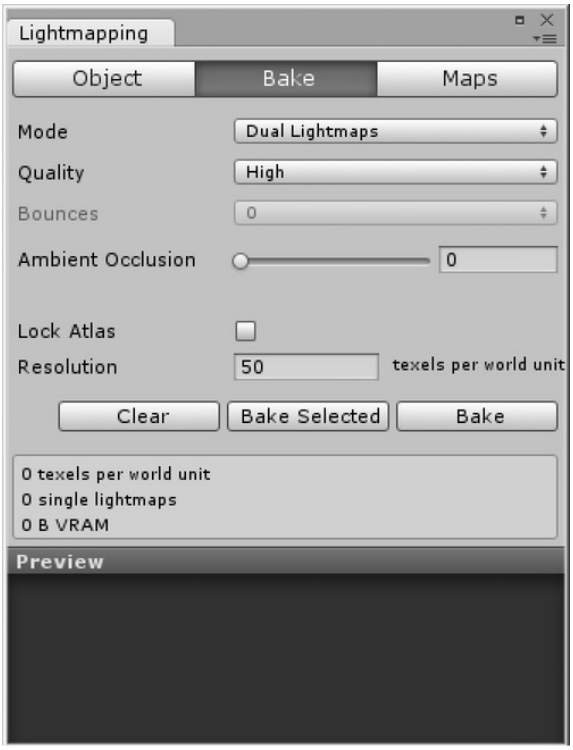


Figure 4-55. *Beast in the free version of Unity. No Bounces or Global Illumination options*

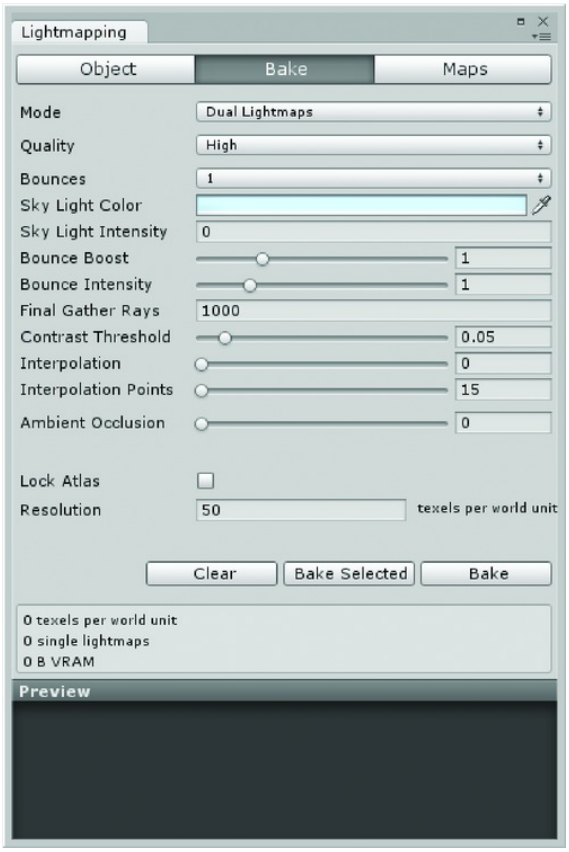


Figure 4-56. *Beas in Unity Pro. More settings to use Global Illumination*

Let's start by baking out a couple of different sized lightmaps just to see the difference. Since you will be adding objects to the scene throughout the development of a game, the lightmap will need regular updating, so there is no need to spend much time on it at this stage.

1. If you are using Unity Pro, set the Bounces to 0.
2. Set Mode to Single Lightmaps if you are using Unity Indie.
3. With the Terrain selected, pick the Bake Selected button.

The status line, bottom, far right, will show the progress as the light is baked. A thumbnail of the resulting map (or maps) is shown in the Preview window.

4. Note the time it took to render the map[s] and the memory it[they] will use, as shown in Figure 4-57.

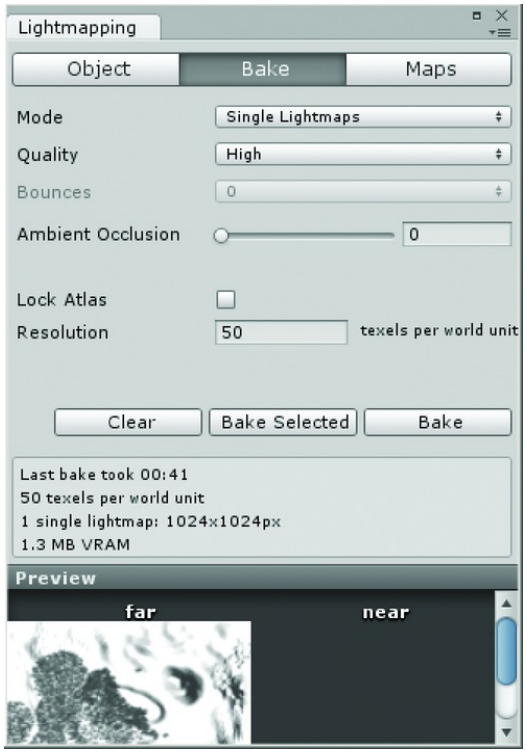


Figure 4-57. *The first bake of the Terrain*

For comparison, you will bake the same settings to a smaller map size. The default size for the Terrain is 1024; try one at 256 to see the difference.

1. From the Object tab, change the Lightmap Size to 256.
2. Press Bake Selected.

Figure 4-58. *The results of the two map sizes, 256 (left) and 1024 (right)*

As you can see in Figure 4-58, the smaller map produces soft indistinct shadows. Note that the smaller map takes up 85.3KB instead of 1.3MB.

3. If you are using Unity Pro, set the Bounces back to the default of 1 Bounce and render the 1024 map size again.

There may be a slight lightening of the shadows from light that could bounce back off the tree trunks, but at this point, it is not worth the extra render time (see Figure 4-59). Bounced light will become more important with the addition of structures. For this book, since it is not assumed that everyone will have Unity Pro, the structures will contain pre-baked lightmaps.

Figure 4-59. *Terrain with baked shadows*

You may be tempted to try the Ambient Occlusion settings. Ambient Occlusion, located in the Bake tab, is calculated apart from any scene lights. It mimics the affect of nooks and crannies being occluded from light (see Figure 4-60). Generally, only objects with convex or occluded areas will benefit greatly from this option and it adds considerable time to calculate. Unity terrains rarely fall under this category, so skip it for now.

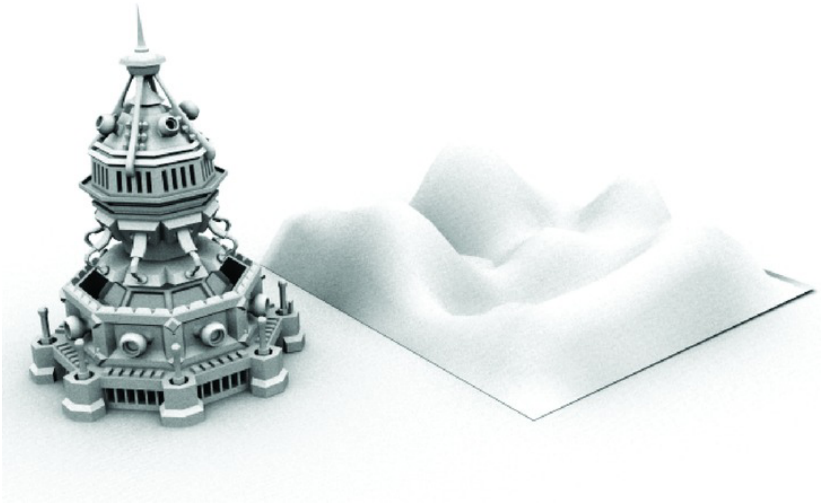


Figure 4-60. *An object that will benefit from ambient occlusion (left) and one that will not (right).*

Fog

As a final touch for your test scene, you can add a bit of fog for some environmental perspective. Fog is fairly cheap and will help promote the feel of a moist tropical jungle.

1. From the Edit menu, select Render Settings.
2. Check Fog, as shown in Figure 4-61.
3. Change the fog color to a greenish blue to simulate light filtering through the forest with high humidity, as shown in Figure 4-62.

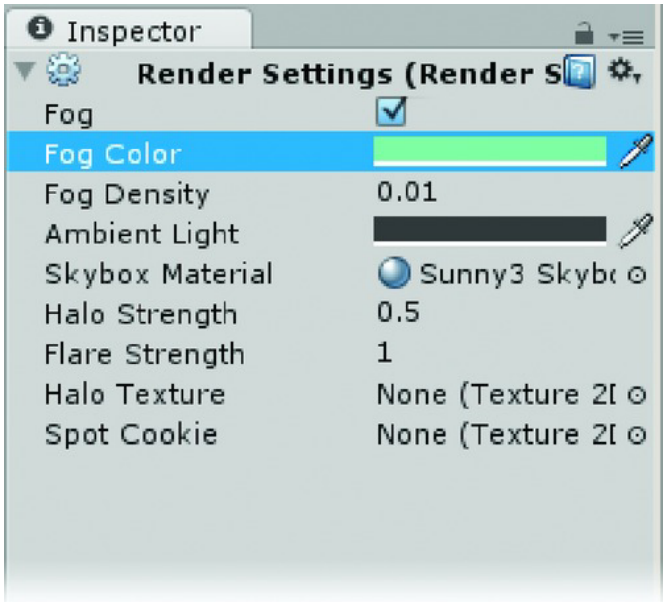


Figure 4-61. *Render Settings with Fog activated*

Figure 4-62. *The fog in the Game window*

Before you move on, you will need to create a large flat area somewhere in the environment for your game's future structures.

1. Set the Paint Height tool's Height setting to 50.
2. Create an area approximately 50 units/meters in diameter somewhere on the terrain.
3. Clear most of the trees from the area. Use the shift key to remove the trees in Place Tree mode.

Chapter Summary

In this chapter, you learned how to use Unity's built-in terrain editor to create the basis for your outdoor environment. Using the Raise/Lower Height, Paint Height, and Smooth Height tools, you were able to sculpt mountains, lakes, and valleys, as well as create building pads and other areas of interest. You learned that focusing on a terrain object is different than regular objects, but that the Flythrough mode of scene navigation is quite useful.

You were able to paint multiple textures on the terrain using the variety of brushes available in the editor. You also learned how to add grass and mesh objects with the option to sway in the wind. Trees were added the same way, but you discovered that to prevent the player from going right through the trunks, you needed to create a prefab containing a Collider. You discovered how to import packages of

prepared assets and learned a few tips for creating your own terrain assets. Finally, you learned how to bake shadows into the terrain at a few different settings and then added a bit of fog for some environmental perspective.

In the next chapter, you will learn how to navigate through your environment and start customizing the existing scripts to fit the needs of your adventure game.

■ **Tip** You can find more on terrain creation with Unity's Terrain Toolkit. This a free tool provided by Unity that you can use to rapidly create terrains. A video tutorial provides a brief but helpful introduction on how to use it. Go to `http://unity3d.com/support/resources/unity-extensions/terrain-toolkit`.
