

CE318: High-level Games Development

Lecture 5: Terrains

Diego Perez

dperez@essex.ac.uk
Office 3A.527

2016/17

Outline

1 Terrains

2 Lab Session 5

Outline

1 Terrains

2 Lab Session 5

Terrains

Almost all video games make use of terrains. This could be terrain that can be actively **explored** (e.g., first person shooter) or terrain in the background to create the illusion of **open-ended** worlds (e.g., racing game).



Terrain can be generated automatically on the fly to create **truly** open-ended worlds. Alternatively, another common usage of mountain ranges is to **limit** the world available to the gamer.

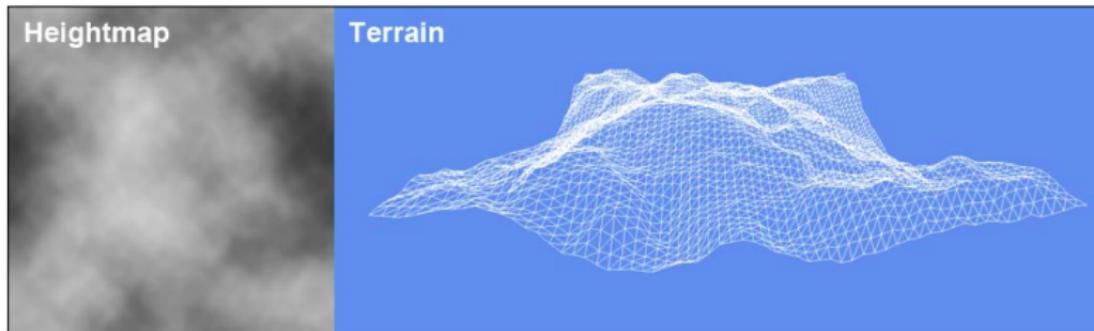
Note: remember that distant elements will suddenly come into view given the far plane of the viewing furustum. One can use elements like **fog** to ensure smooth transitions without the need to draw too many items in the distance.

Terrain generation

A common way to generate terrain is to build a mesh (using triangles) from a **heightmap** and to apply a texture to that mesh.

A heightmap is a 2D image in gray scale:

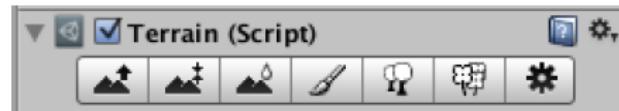
- Each pixel corresponds to a vertex.
- The colour value of the pixel determines its height



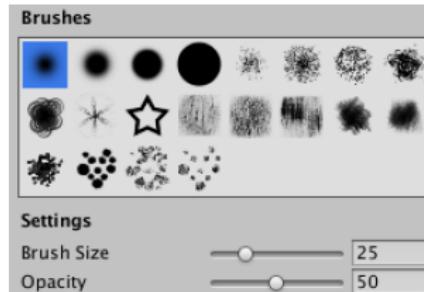
A heightmap may be generated using special tools or simply using an image editor such as Photoshop or Paint.NET.

Creating and Editing Terrains

In Unity, terrains can be added as game objects from *Create* → *3D Object* → *Terrain*, in the *Hierarchy View*. This creates a completely flat large plane, and the surface can be modified by using the Height Tools available in the *Terrain (Script)* component.



From left to right, these tools are: *Raise/Lower Terrain*, *Paint Height*, *Smooth Height*, *Paint Texture*, *Place Trees*, *Paint Details* and *Terrain Settings*. With the exception of the tree placement tool and the settings panel, all the tools on the toolbar provide a set of “brushes” and settings for brush size and opacity.



Height Tools *

The Height Tools (the first three tools: *Raise/Lower Terrain*, *Paint Height* and *Smooth Height*) are used to **paint changes** in height onto the terrain. They essentially work by holding the mouse in one place, similar to the effect of the airbrush tools in image editors, what elevates the terrain in the area clicked. If the *Shift* key is pressed simultaneously, the height of the terrain at that point will be lowered. The *Brush Size* and *Opacity* options vary the area of the brush and the strength of its effect respectively.

The *Paint Height* tool has an additional property (*Height*), that works as a **target height** value. Higher points in the terrain will be lowered (and lower points will be elevated) when clicking using this tool. Using the *Shift* key allows to sample the height of a particular point. Next to the Height property is a *Flatten* button that simply **levels** the whole terrain to the chosen height.

Finally, *Smooth Height* does not substantially raise or lower the terrain height but rather **averages out nearby** areas, softening the landscape and reducing the appearance of abrupt changes.

You can import and export your terrain height in RAW file formats (*Terrain Settings* → *Import Raw* and *Export Raw* buttons respectively).

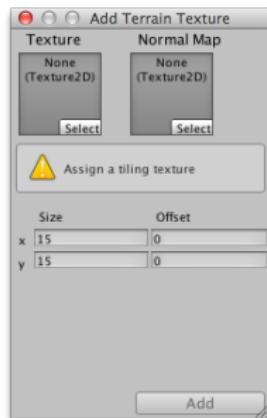
Textures (1/2)

Textures can be added to the surface of a terrain.

- Since terrains are such large objects, it is standard practice to use a texture that repeats seamlessly and tile it over the surface.
- One texture will serve as the background image over the landscape but you can also paint areas of different textures to simulate different ground surfaces such as grass, desert and snow.
- The painted textures can be applied with variable transparency so you can have a gradual transition between grassy countryside and a sandy beach, for example.

Initially, the terrain will have no textures assigned for painting. If you click the Edit Textures button and select Add Texture from the menu, a window will be shown to add textures.

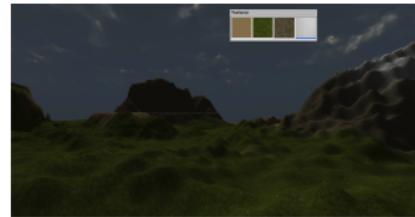
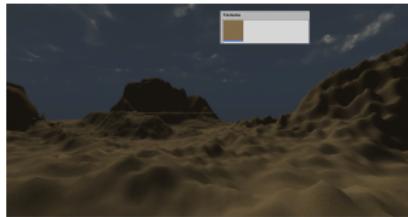
The standard assets supplied with Unity include a number of terrain textures that are useful for simple effects.



Textures (2/2) *

The first texture added will work as a background, painting the whole terrain. The rest of the textures added will be available for painting using the brush tools, as seen with the Height tools.

Properties available are *Brush Size* and *Opacity*, which work as in the other tools, and a third setting called *Target Strength*. This last setting sets the maximum opacity value that the brush will build up even if it passes over the same point repeatedly (useful for patches within a single terrain to break the monotony of a large homogeneous area).

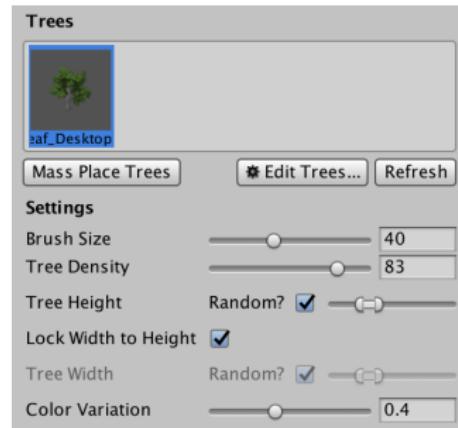


Trees (1/3)

Unity terrains can be furnished with trees. Patches of trees can be painted onto a terrain in much the same way that heightmaps and textures are painted but the trees are solid 3D objects that grow from the surface. Unity uses optimisations (billboarding for distant trees) to maintain good rendering performance, so you can have dense forests with thousands of trees and still keep an acceptable framerate.

As with textures, initially there are no trees available. Clicking on the *Edit Trees* button allows to add trees to the terrain.

With a tree selected, you can paint onto the landscape in the same way you paint textures or heightmaps. You can remove trees from an area by holding the *Shift* key while you paint and remove just the currently selected tree type by holding down the *Control* key.



Trees (2/3) *

Brush Size and *Tree Density* (which *replaces* “*Opacity*”) are used to paint trees as used for painting textures and changing heights. *Color Variation* adds some noise in the color of the trees planted. Height and Width of the tree can also be determined, with noise factor as well.

The *Mass Place Trees* button is a very useful way to create an overall covering of trees without painting over the whole landscape.



Trees (3/3)

Colliders with Trees: You can add a Capsule Collider to a new tree asset by instantiating it in the scene (from the prefab), adding a Capsule Collider and saving the prefab for the modified tree object.

Note that the physics engine has a limit of 65536 colliders in a single scene. This is normally not an issue but when trees are used with colliders, you should make sure you don't exceed this limit.

Making trees bend in the wind: When a *Wind Zone* is available (see later), it is possible to make the trees bend with the wind. The trees need to be specifically set for this, in *Edit Trees* → *Edit tree*. Setting the *bend* value to 1 will cause the trees to bend with the wind (notice that you must also adjust the wind strength to determine the overall effect).

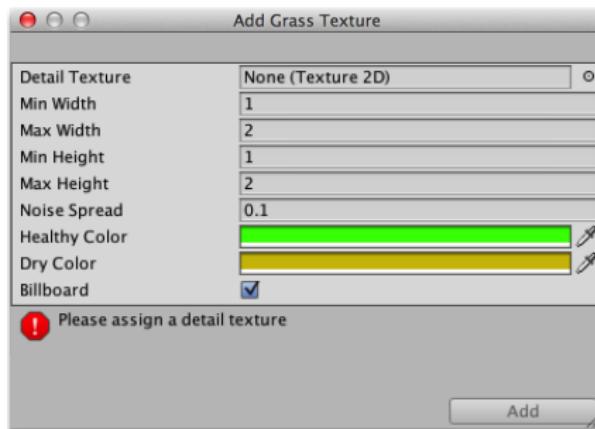
Unity has its own **Tree Creator** (*Assets* → *Import Package* → *Tree Creator*). Trees can be created as a game object (*3D Object* → *Tree*), and you can specify settings as branches, leaves and materials, and also how they are affected by wind zones. Check the manual for Tree Creation at:

<http://docs.unity3d.com/Manual/class-Tree.html>

Grass and Other Details (1/3)

The *Paint Details* button on the toolbar enables grass/detail painting. You must click on the *Edit Details* button to enable these.

Add Grass Texture: This allows to specify a texture for the grass. The texture is simply a small image with alpha set to zero for the empty areas (note that “grass” is rather generic: you could create flowers or other objects with different textures):



The *Min Width*, *Min Height*, *Max Width* and *Max Height* values specify the upper and lower limits of the size of the clumps of grass that are generated.

Grass and Other Details (2/3) *

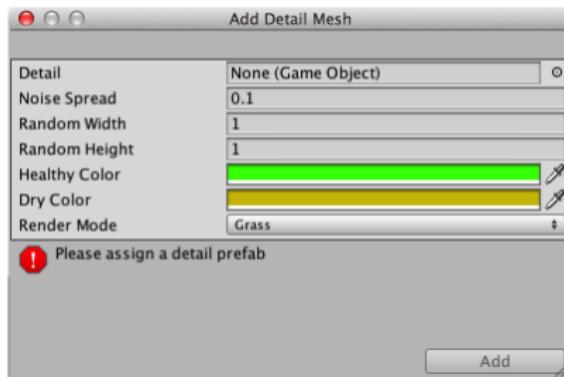
The *Noise Spread* value controls the approximate size of the alternating patches, with higher values indicating more variation within a given area. The alternating patches of grass are considered more “healthy” at the centres than at the edges and the *Healthy/Dry Color* settings show the health of grass clumps by their color.

If the *Billboard* option is enabled, the grass images will rotate so that they always face the camera.



Grass and Other Details (3/3)

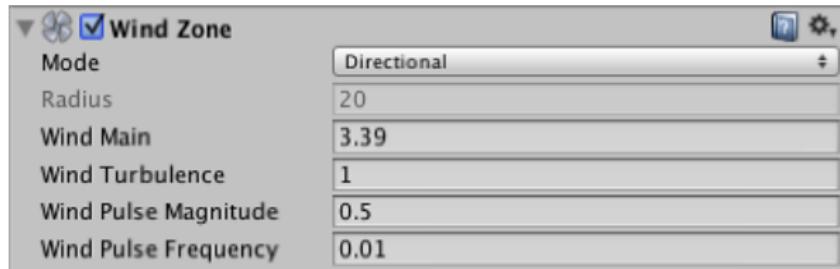
Add Detail Mesh: This allows to scatter copies of a mesh around the terrain. When you select the *Add Detail Mesh* option, you will be prompted with the following panel:



The *Detail* property is used to select a prefab from your project which will be scaled by the *Random Width* and *Random Height* values for individual instances. The *Noise Spread* and *Healthy/Dry Color* values work the same as they do for grass. The *Render Mode* can be set to *Grass* (the instances of detail objects in the scene will be flattened into 2D images - like grass textures) or *Vertex Lit* (details will be rendered as solid, vertex lit objects in the scene).

Wind Zones *

You can create the effect of wind on your terrain by adding one or more objects with **Wind Zone** components:

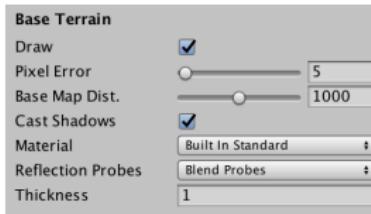


The *Mode* can be set to *Directional* (affects the whole terrain) or *Spherical* (blows outwards within a sphere defined by the *Radius* property). Directional winds are more useful for creating natural movement of the trees while spherical winds are more suitable for special effects like explosions.

The *Wind Main* property determines the overall strength of the wind but this can be given a little random variation using *Wind Turbulence*. As mentioned above, the wind blows over the trees in pulses to create a more natural effect. The strength of the pulses and the time interval between them can be controlled using the *Wind Pulse Magnitude* and *Wind Pulse Frequency* properties.

Terrain Settings (1/4)

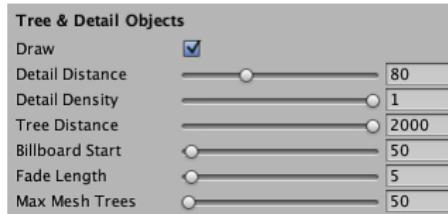
Base Terrain:



- **Show:** If terrain should be shown or not.
- **Pixel Error:** Higher values for lower accuracy (relation height/textures and terrain) but lower rendering overhead.
- **Base Map Distance:** The maximum distance at which terrain textures will be displayed at full resolution. Beyond this distance, a lower resolution composite image will be used for efficiency.
- **Cast Shadows:** Does the terrain cast shadows?
- **Material:** The material used to render the terrain.
- **Reflection Probes:** How reflection probes are used on terrain.
- **Thickness:** How much the terrain collision volume should extend along the negative Y-axis. (to prevent high-speed moving objects from penetrating into the terrain).

Terrain Settings (2/4)

Tree and Detail Objects:

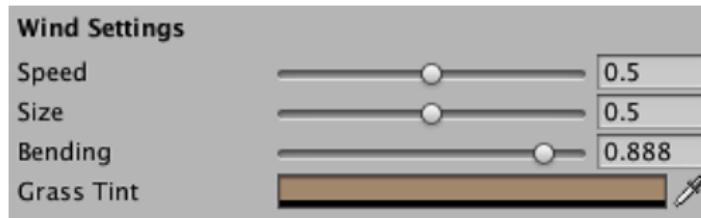


- **Draw:** Should trees, grass and details be drawn?
- **Detail Distance:** The distance* beyond which details will be culled.
- **Detail Density:** The number of detail/grass objects in a given unit of area. The value can be set lower to reduce rendering overhead.
- **Tree Distance:** The distance* beyond which trees will be culled.
- **Billboard Start:** The distance* at which 3D tree objects will be replaced by billboard images.
- **Fade length:** Distance* over which trees will transition between 3D objects and billboards.
- **Max Mesh Trees:** The maximum number of visible trees as solid 3D meshes. Beyond this limit, trees will be replaced with billboards.

* Distances indicated from the camera.

Terrain Settings (3/4)

Wind Settings:



- **Speed:** The speed of the wind as it blows grass.
- **Size:** The size of the ripples on grassy areas as the wind blows over them.
- **Bending:** The degree to which grass objects are bent over by the wind.
- **Grass Tint:** Overall color tint applied to grass objects.

Terrain Settings (4/4)

Resolution:

Resolution
Terrain Width
2000
Terrain Length
2000
Terrain Height
600
Heightmap Resolution
513
Detail Resolution
1024
Detail Resolution Per Patch
8
Control Texture Resolution
512
Base Texture Resolution
1024

* Modifying the resolution will clear the heightmap!

- **Terrain Width/Length:** Size of the terrain object in its X/Z axis.
- **Terrain Height:** Difference in Y coordinate between the lowest possible heightmap value and the highest (in world units).
- **Heightmap Resolution:** Pixel resolution of the heightmap.
- **Detail Resolution:** Resolution of the map for the separate patches of details/grass. Higher resolution gives smaller and more detailed patches.
- **Detail Resolution Per Patch:** Length/width of the square of patches rendered with a single draw call.
- **Control Texture Resolution:** Resolution of the splatmap that controls the blending of the different terrain textures.
- **Base Texture Resolution:** Resolution of the composite texture used on the terrain when viewed from a distance greater than *Basemap Distance*.

Terrain Collider *

Terrain Collider: The Terrain Collider takes a Terrain and builds its Collider based on that terrain.



- Material: The Physic Material used for the terrain surface to specify its friction and bounce.
- Terrain Data: The terrain data.
- Create Tree Colliders: When selected Tree Colliders will be created.

Outline

1 Terrains

2 Lab Session 5

Lab Preview

In this week's lab you'll create a terrain with all its features, including also audio and lighting.

Next lecture: GUIs, Menus and Data, Particle Systems.