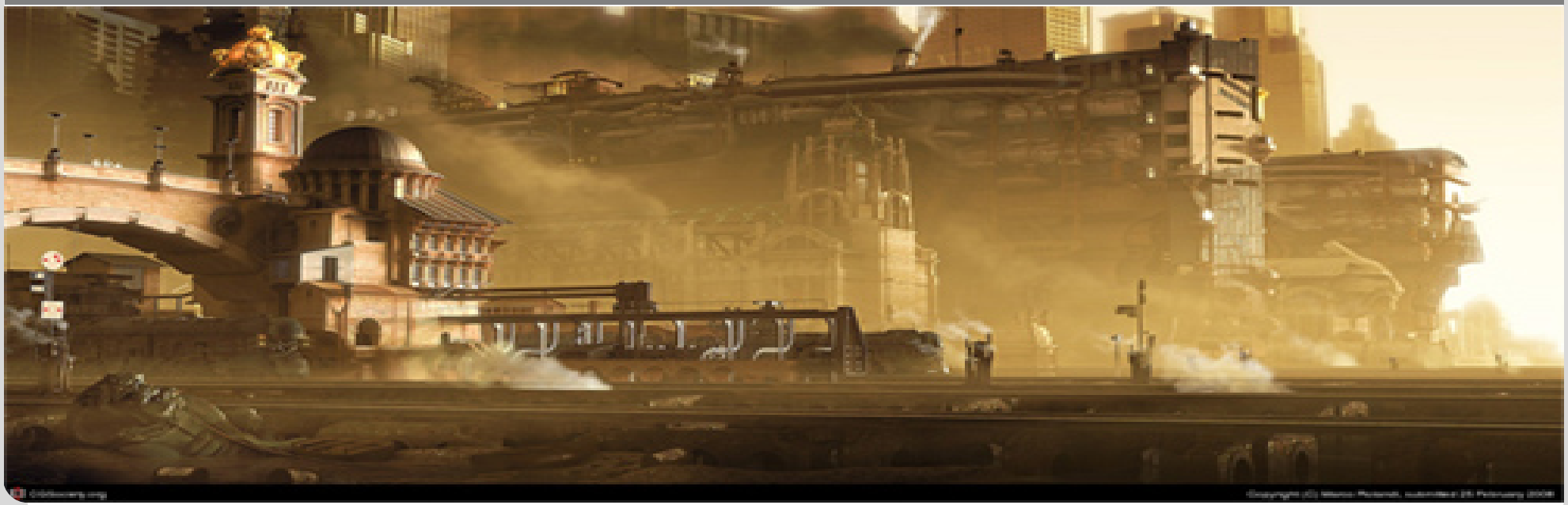
Terrain

Computer Graphics - SET08116

EDINBURGH NAPIER UNIVERSITY



Outline

1. Review
2. What is Terrain Rendering?
3. Summary

Review

- Texturing is the process of applying image data to a surface
  - Wall can be defined using simple geometry
  - Image of a brick wall can be added to make it a brick wall
  - Image of wooden wall can be added to make it a wooden wall
- Texturing is the third part of what can be considered core 3D rendering
  - Geometry, Lighting, Texturing

Review

- Shaders are small programs that run on the GPU
- Shaders allow us to implement effects that we may wish for in our rendered scene
  - Lighting, texturing being the most basic
- Three types of shader
  - Vertex
  - Geometry
  - Fragment (or pixel)

What is Terrain Rendering?

- Terrain rendering is the process of rendering realistic world surfaces
  - Can be real-world or alien world
- The goal of terrain rendering is to provide an outdoor environment for your game
  - Even cityscapes can have underlying terrain
- Terrain rendering generally focuses on the development of hills, valleys, mountains, etc.
  - These can be procedurally generated

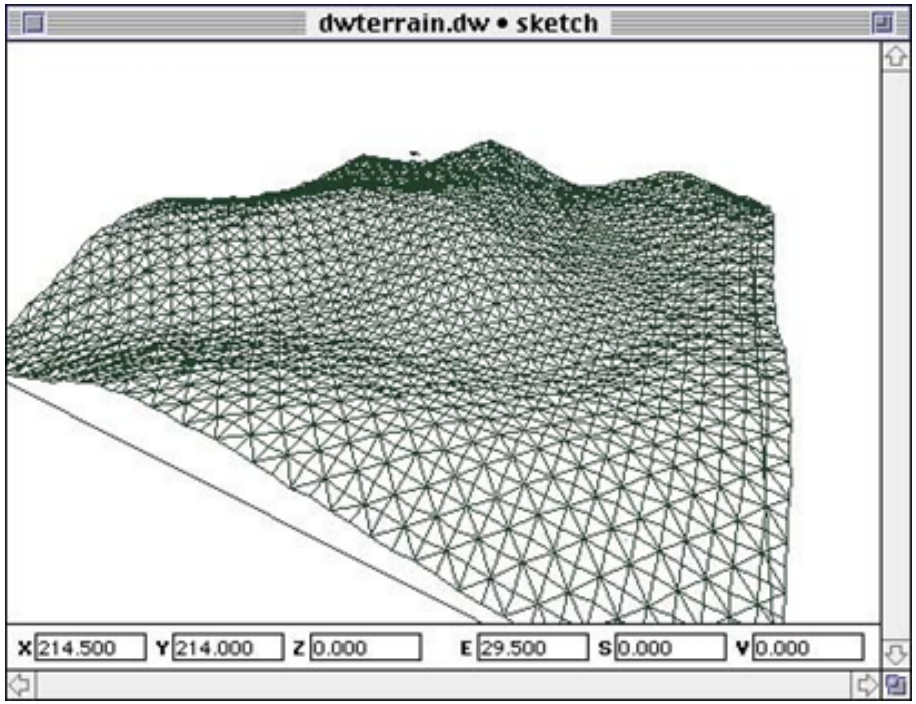
Example

How Terrain Rendering Works

- Terrain rendering utilises base triangle meshes, manipulated by a height map, with relevant lighting and texturing added
- Four key concepts
  - Generation of a triangle grid
  - Reading of texture data to set vertex heights
  - Generation of normals for each vertex
  - Texturing / colouring of vertices based on height / change in height

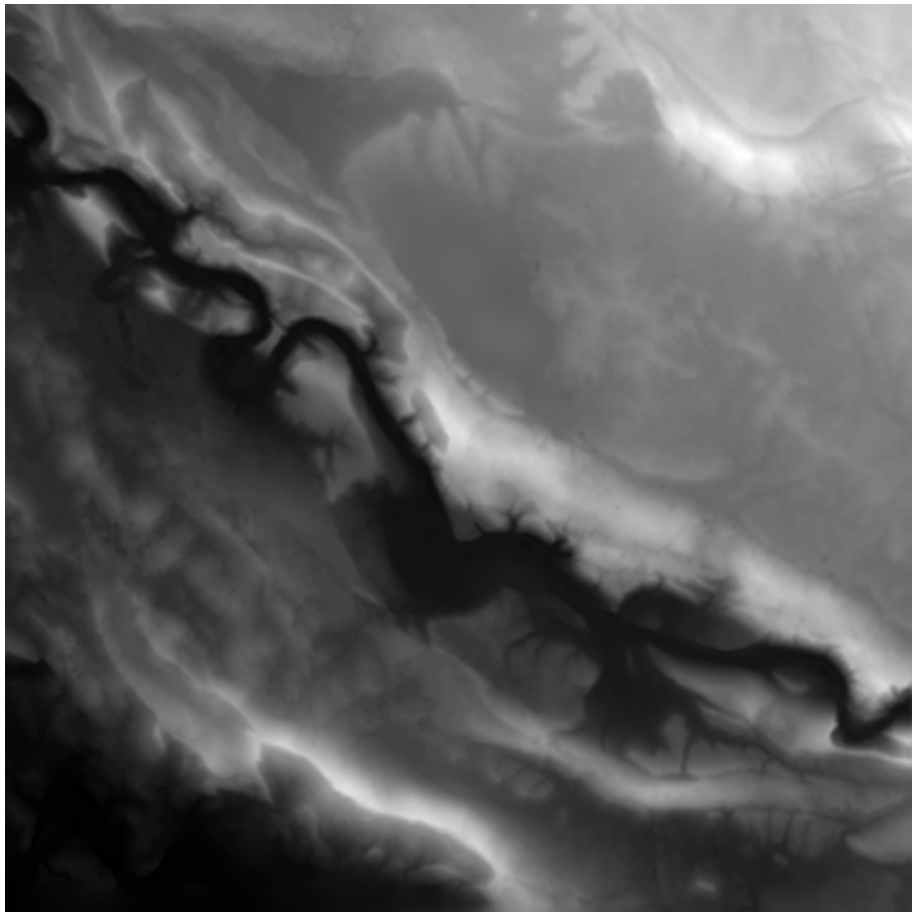
Triangle Grid

- The basic part of a terrain is a large, flat grid of triangles
  - A 2D array of 3D vertices
  - Each vertex initially has a height of 0
- The heights of each vertex are manipulated, leading to a terrain like effect



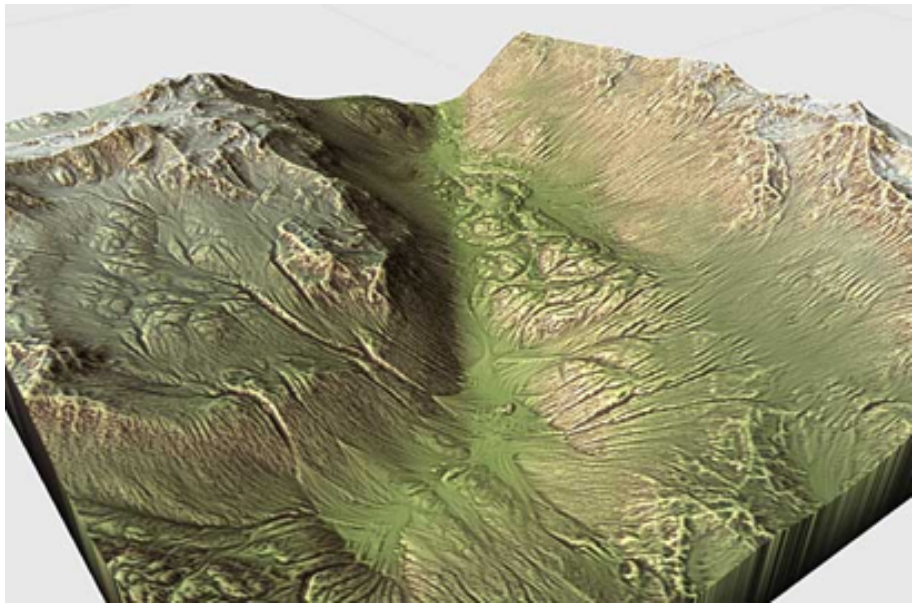
Height Map

- To get the heights of an individual vertex, we use a height ma
- Each pixel of the texture represents a height value of the terrain
  - Dark low, light high

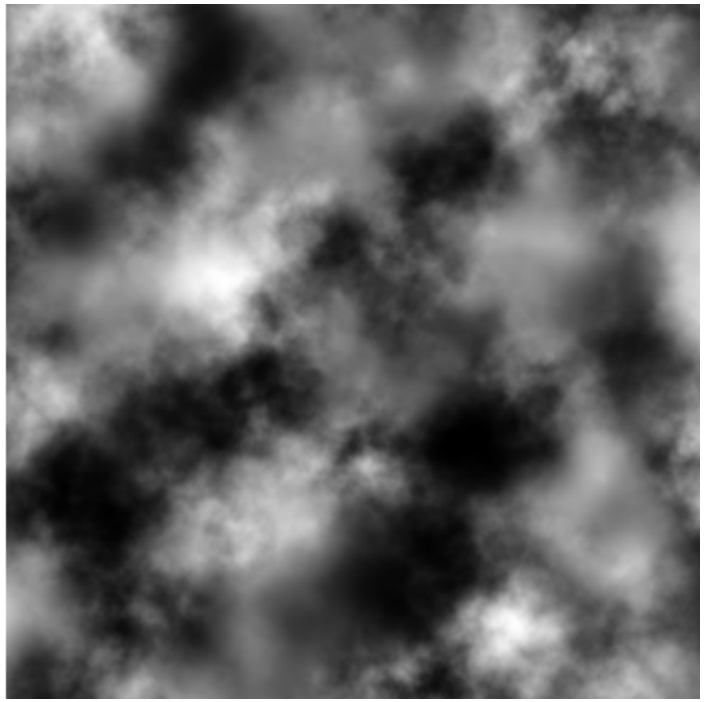


Texturing and Lighting

- Finally, we add texturing and lighting to the terrain to get the effect we want
  - Normals have to be generated (or read from an image)
  - Textures can be blended

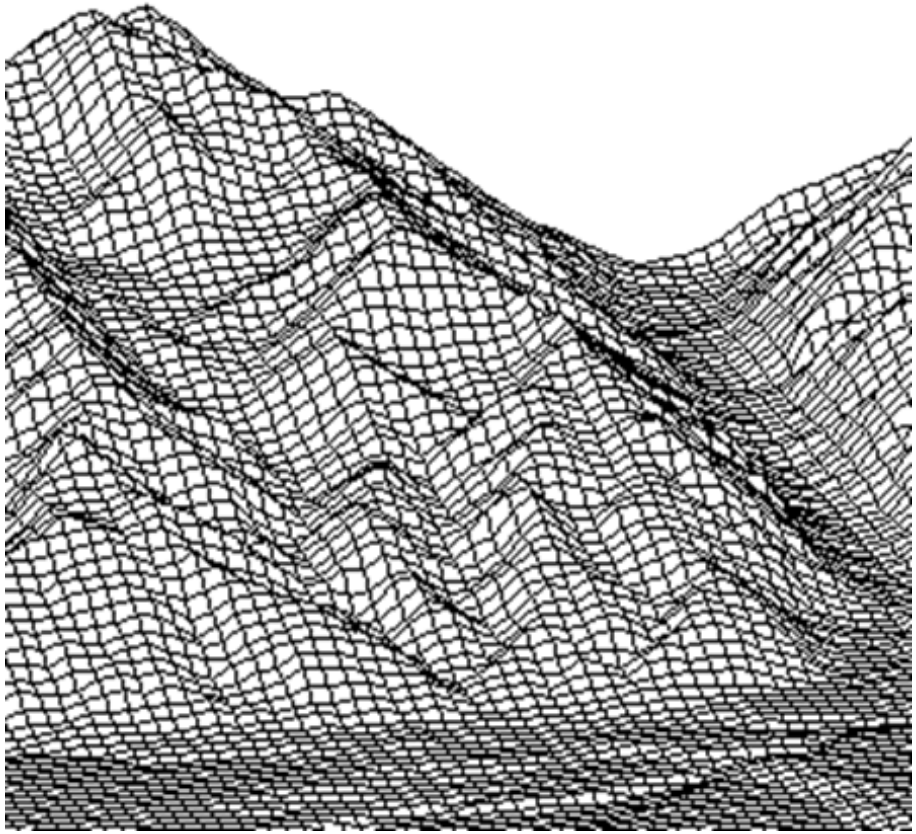


Height Map Examples



Terrain Mesh

- The goal is to create a triangle mesh that represents the terr
  rendered
- Each vertex has a height based on the colour of the height map
  - Some smoothing may be required



Texture Data

- As you know, a texture is just a 2D array of pixels
- Each pixel has an RGB(A) value associated with it when it is read in
- These pixels can be analysed to determine there colour
  - We can use the intensity calculation from last week
- This value can then be used to set the individual y components of our vertices

Creating a Height Map

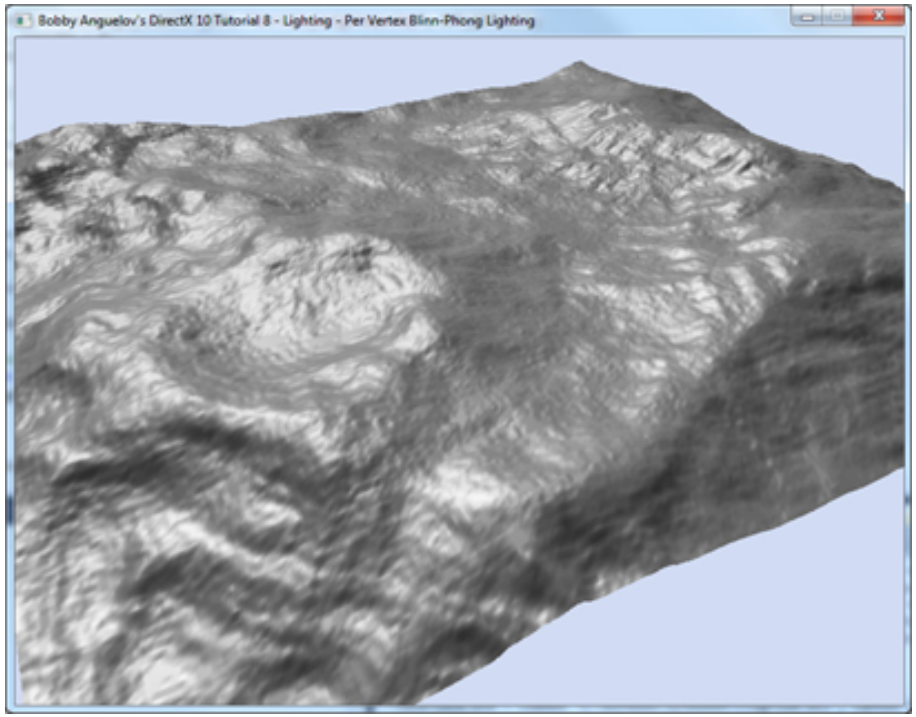
- Sometimes you will want to create a height map from real world data
  - You can normally acquire height data from somewhere
- Sometimes you may want your artist to generate your height map
  - Think Sim City or similar
- You can also generate this map yourself
  - Procedural

Filtering and Smoothing

- Depending on how you get your height data, you might have a limited range
  - For example, if you used the red component, you only have 255 unique values
- Technique is to take these values as large steps, and then filter based on the neighbouring vertex heights
  - Take the average of the vertex and its eight neighbours

Generating Normals

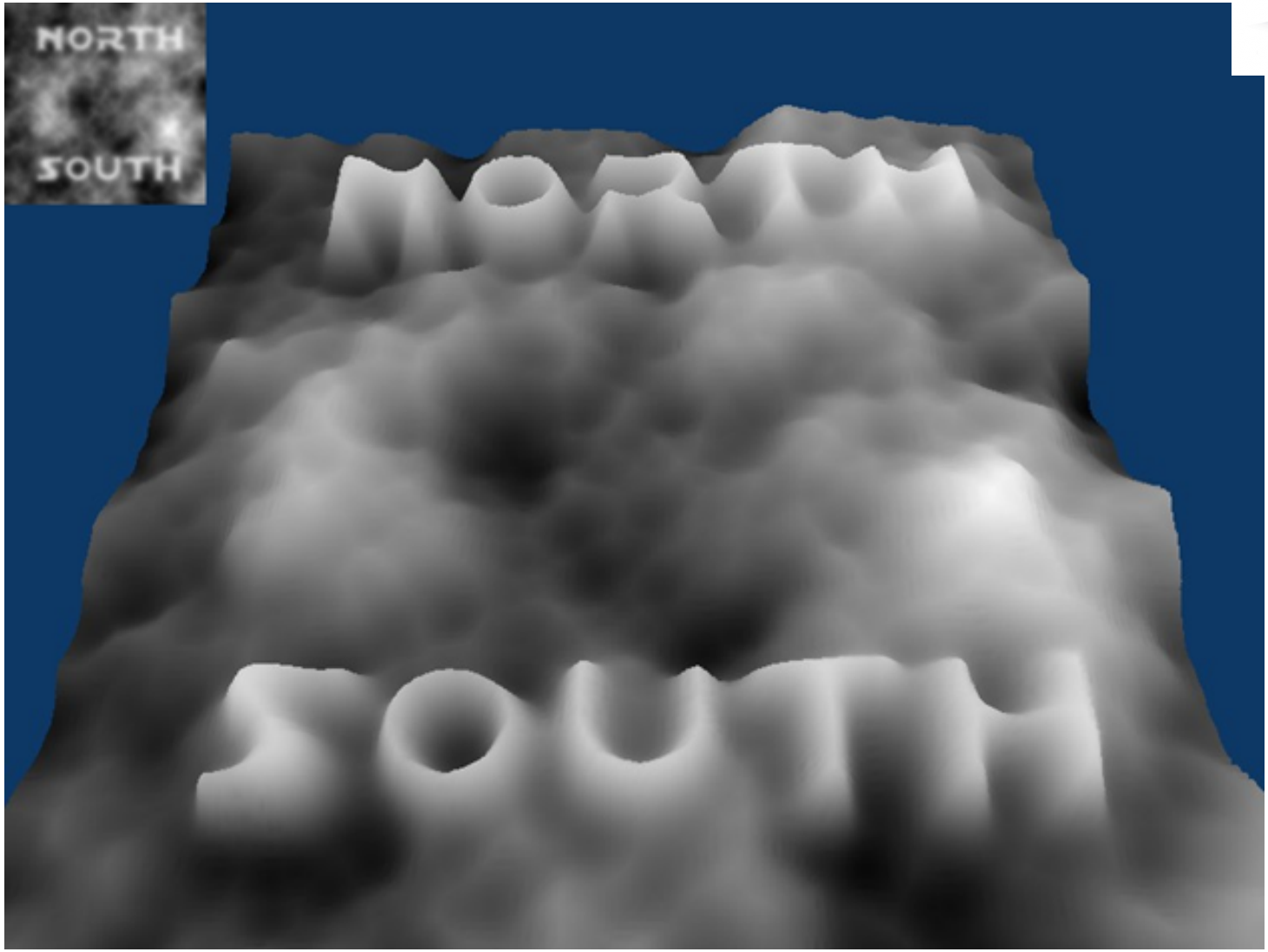
- For each vertex, we have to also generate the normals
- Generally, we calculate based on the neighbouring vertices
  - Use cross product to generate
- Can also use normal maps



Lighting a Terrain

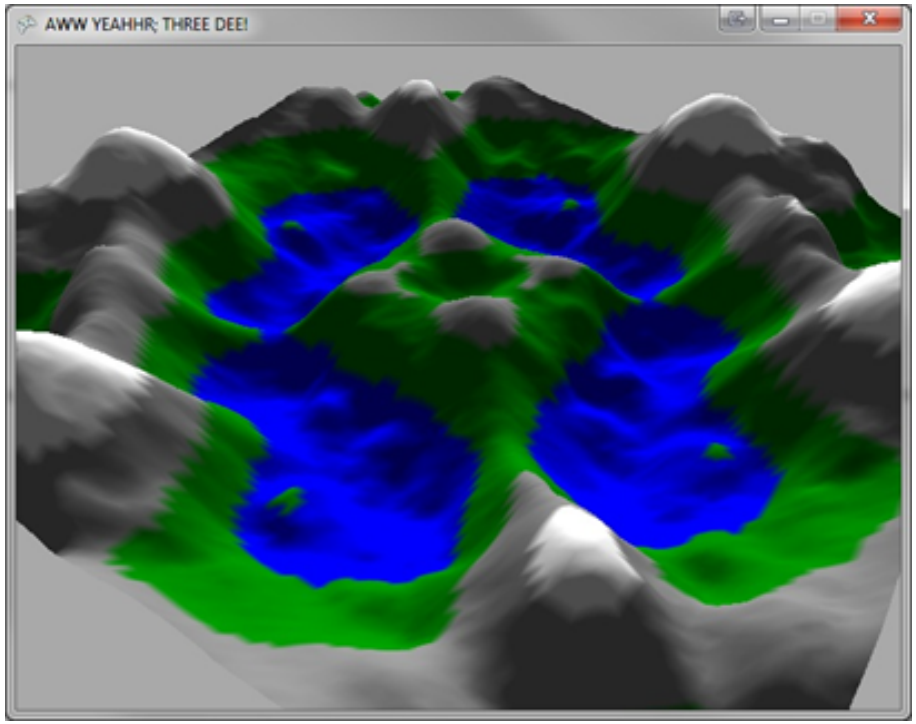
- Once we have a terrain and the normals, we can go about lighting it
- Use the same techniques as before
  - Ambient light
  - Diffuse light
  - Specular light (depends on the terrain)
- We can also use occlusion maps and other light maps to help provide more realistic lighting
  - Longer to generate

Example



Colouring by Height

- Simplest method to increase believability of the terrain is to colour vertices based on height
  - Easily done in the shader
- Choose a number of colours and based on height use that colour

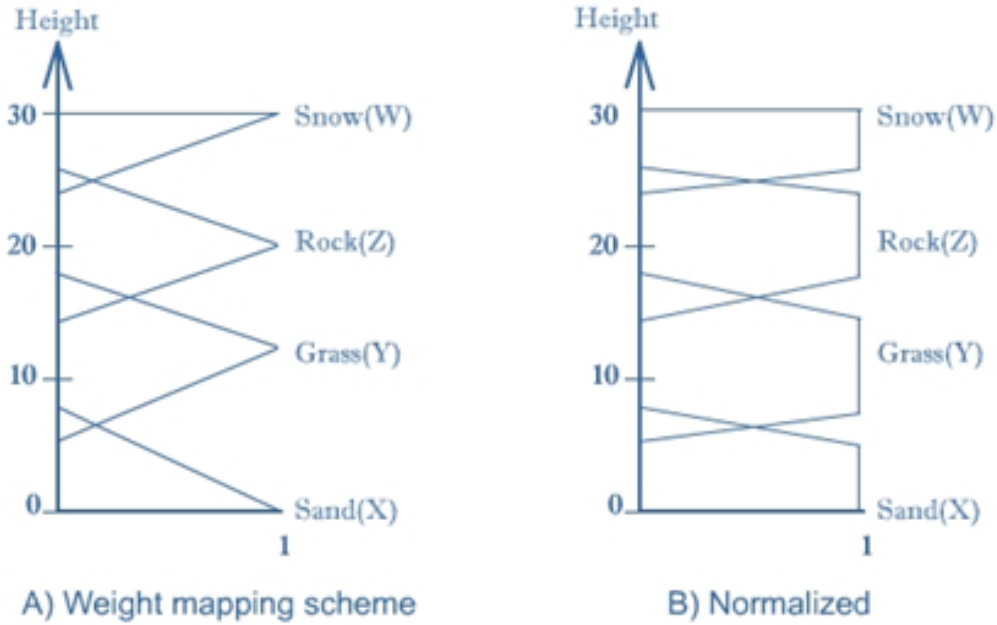


Colouring by Height Change

- We can also colour based on the change of height
  - Steeper slopes have different colours
- Usually we want to combine this with colouring by height to provide a more realistic terrain
  - Height used to determine base colour
  - Steepness used to determine colour to blend with

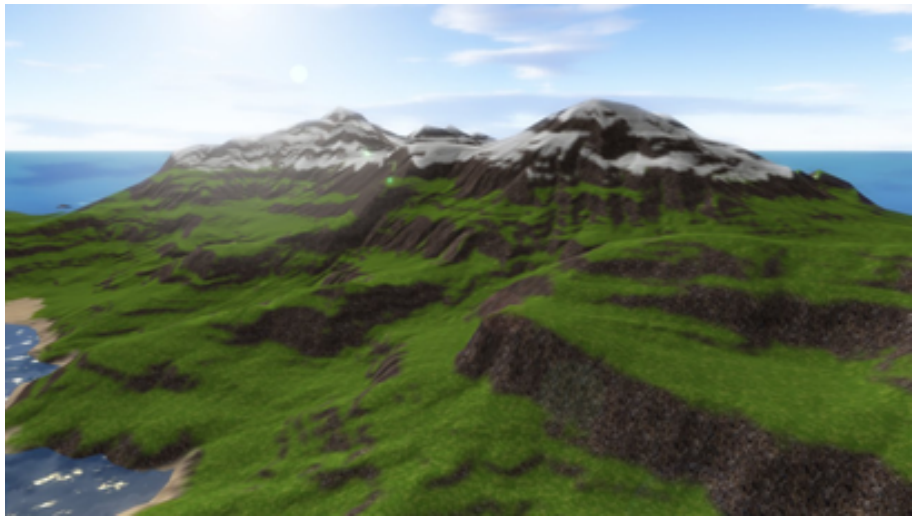
Texturing by Height

- We can perform the same operation as colouring using textures
- Normally we want to blend textures between heights
  - Looks too artificial otherwise



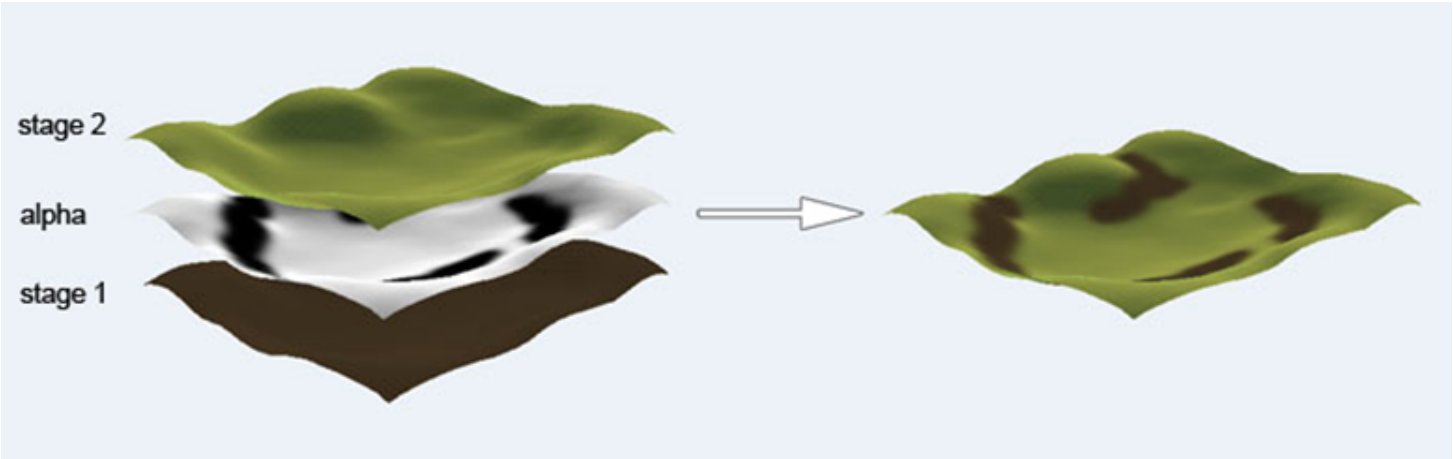
Texturing by Height Change

- We can also take into account the change in height as a blend factor for the terrain
  - Provide cliff faces
- This will provide the most realistic terrain using standard techniques



Using Multiple Terrain Maps

- Sometimes we want to use multiple terrain maps to provide other features
  - For example a road
- Blending textures allows us to create realistic features in our terrain



Using Multiple Terrain Maps

- One final step in generating terrain is adding vegetation
  - Trees
  - Grass
- Common technique for adding grass is billboarding
- Common technique for adding trees is instancing

Billboarding

- Billboarding involves rendering sprites in world space
- As sprites are 2D, they always face the camera
  - Gives a weird effect when moving
- Billboarding is often used in particle effects

Billboarding Example

Summary

- Terrain rendering is one of the commonest techniques used in 3D games
  - Realistic outdoor environments being the goal
- Generating terrain is actually quite easy
  - Height map
  - Triangle mesh
  - Terrain textures
- You normally want to optimise as well
  - Sub grids of the terrain for culling