Terrain Part 1

Wayne Rippin (UoD)

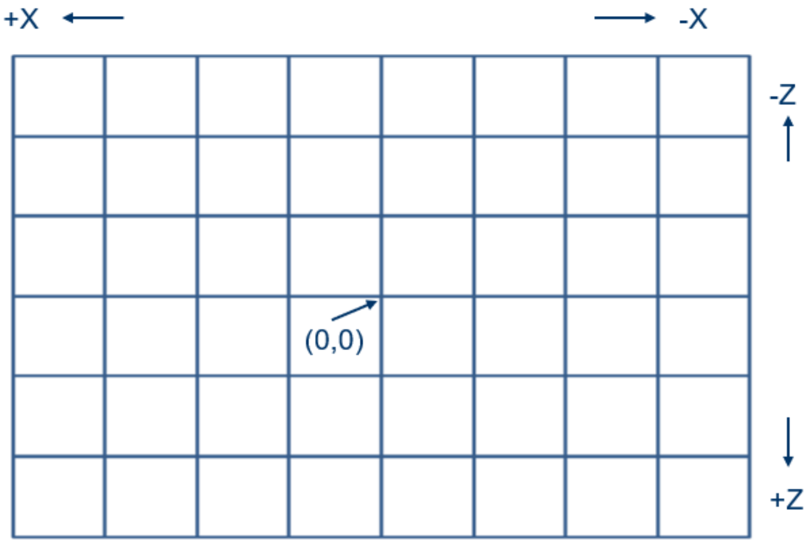
Dr Panagiotis Perakis (MC)

What is Terrain?

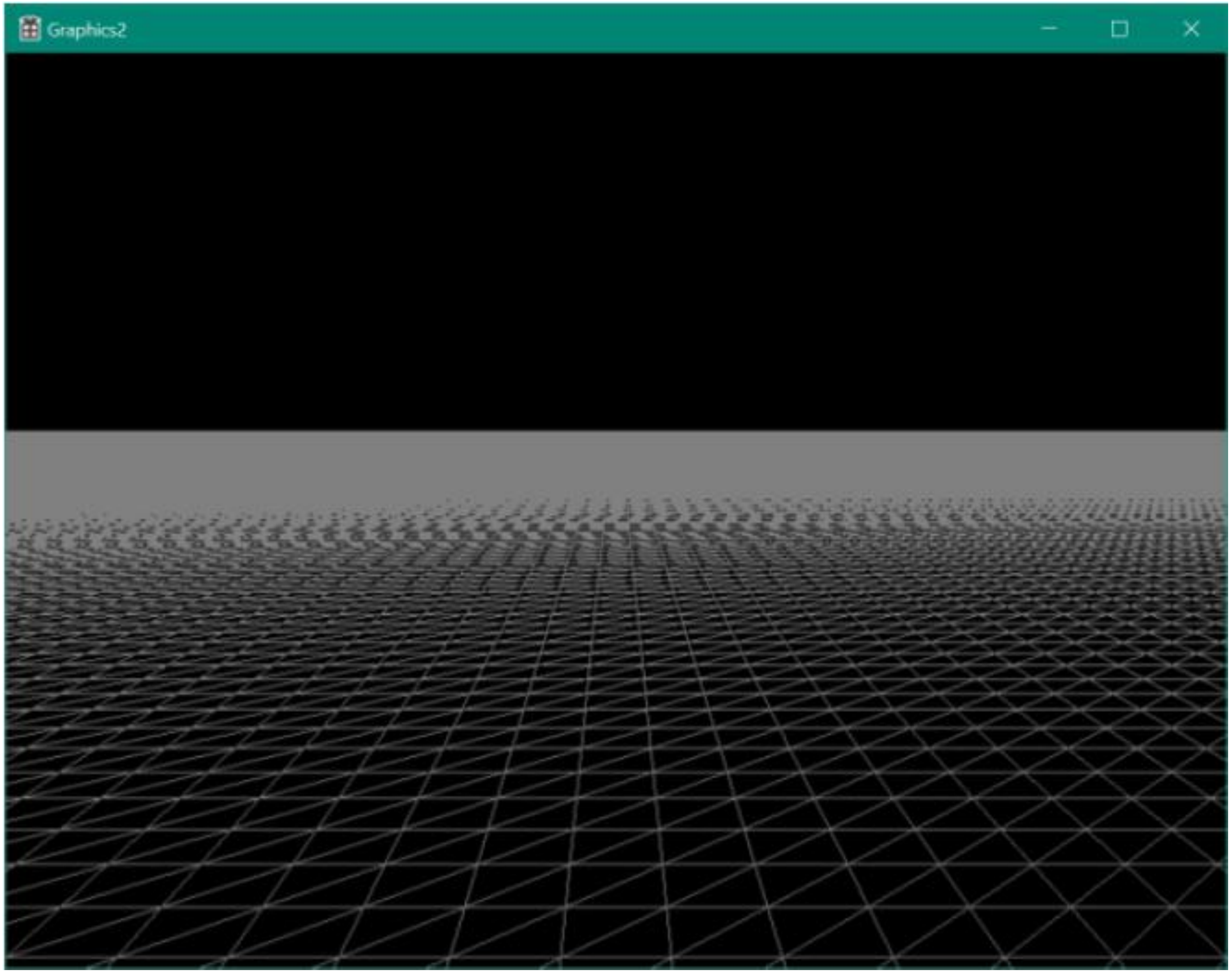
- Terrain makes up the environment that the player moves around in, for example: ground, trees, rocks, water, etc.

Starting Point

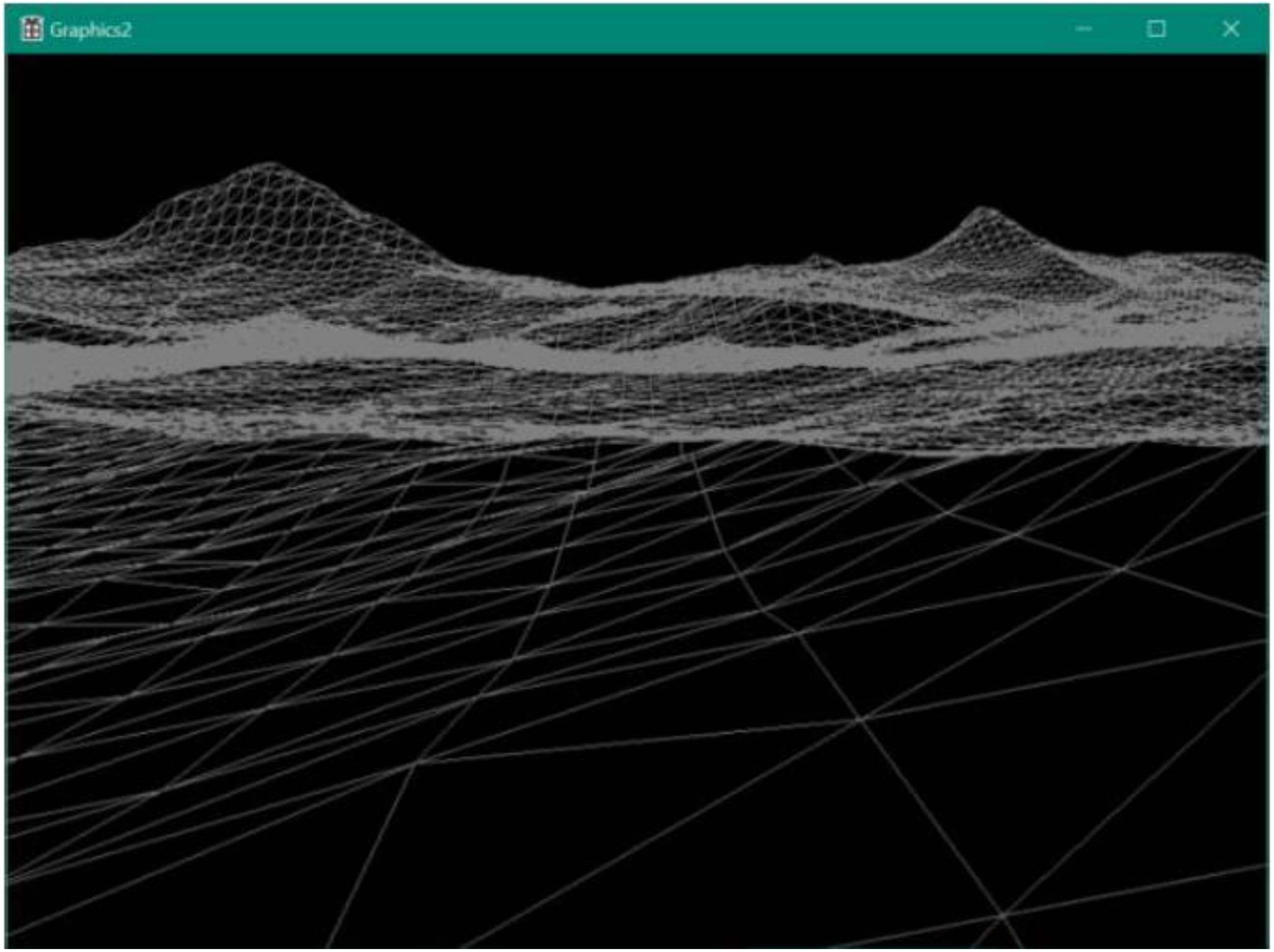
- We start with a grid of squares, each split into two triangles.
- The grid needs to be large enough to cover the area we are interested in
- This is generated by code.
- Each square consists off our separate vertices (i.e. they do not share vertices with other squares). The reason for this is that we will need to provide texture coordinates to the vertices later.



Starting Point

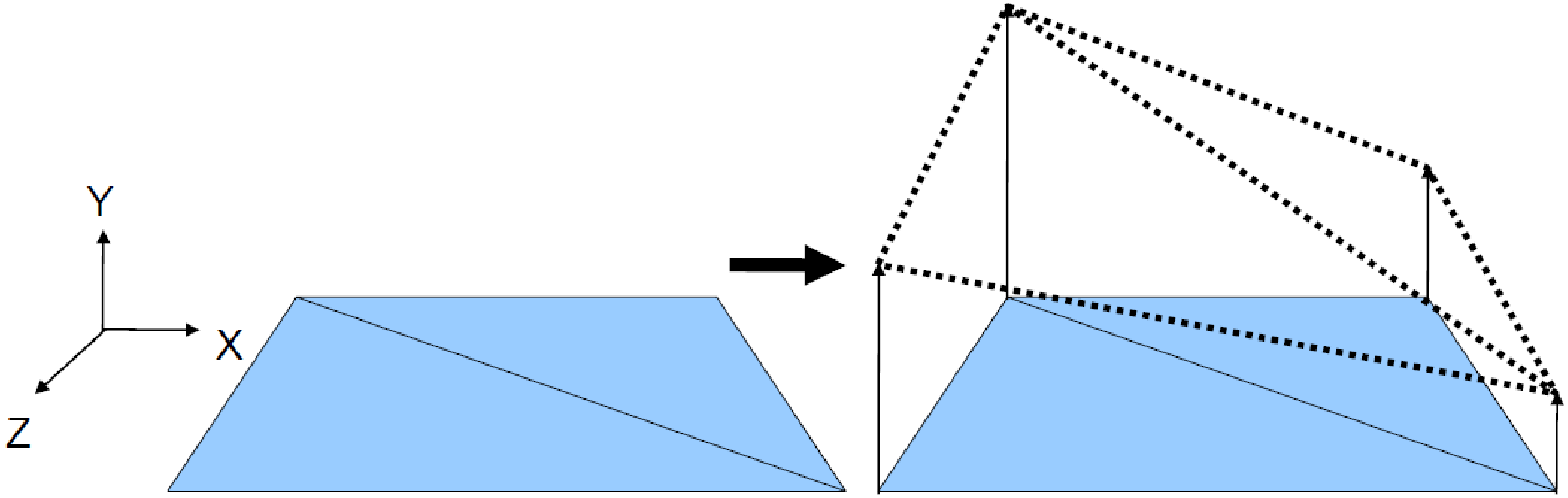


What we want



Generating Terrain

- The grid starts off with the Y co-ordinate of all triangles as 0
- We need to generate Y co-ordinates to produce height



Generating Terrain

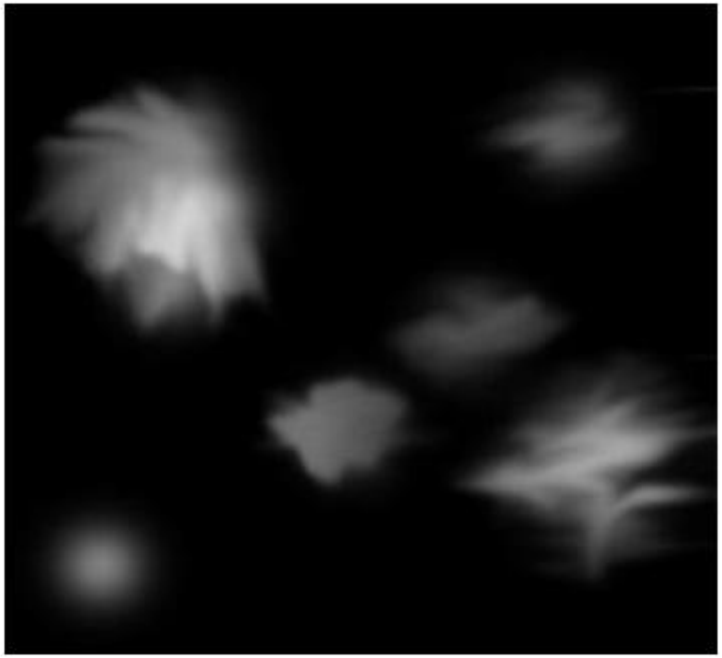
- There are two ways we can generate terrain for this grid:
  - Offline or static generation to create height maps
  - Real-time generation using code at run-time.

Offline or Static Generation

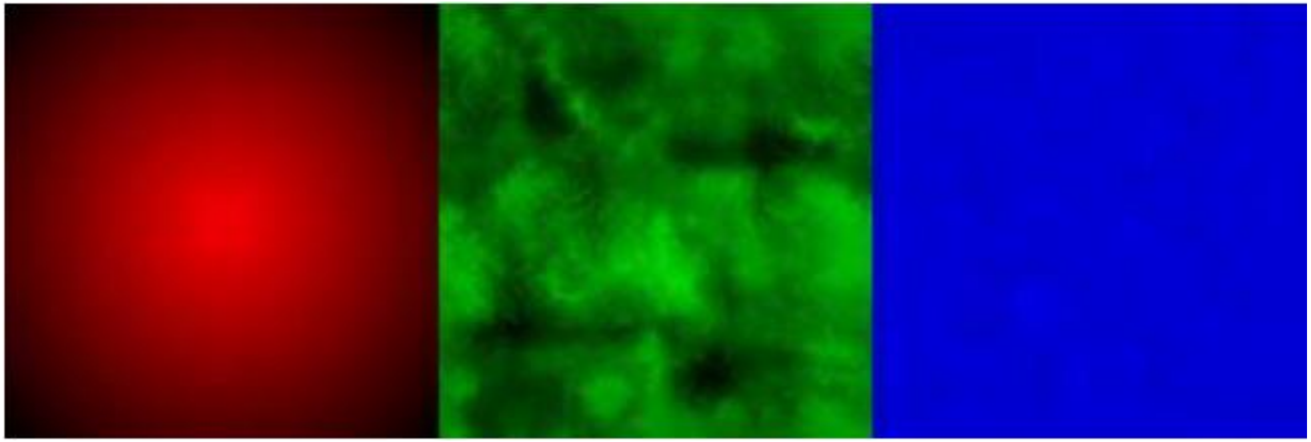
- Makes use of image files, usually RAW format (i.e. no header information)
- Each pixel in the image represents the height at that vertex

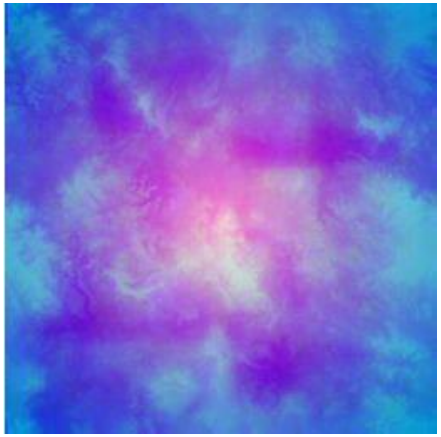
Monochrome Height Map

- Typically uses an 8-bit or 16-bit grey scale bitmap
  - Each pixel is represented by 8 bits (0 –255) or 16 bits (0 –65535)
- In a 16-bit map,
  - White = 65535 = High
  - Black = 0 = Low
  - Grey scale produces gradient



Colour Height Map



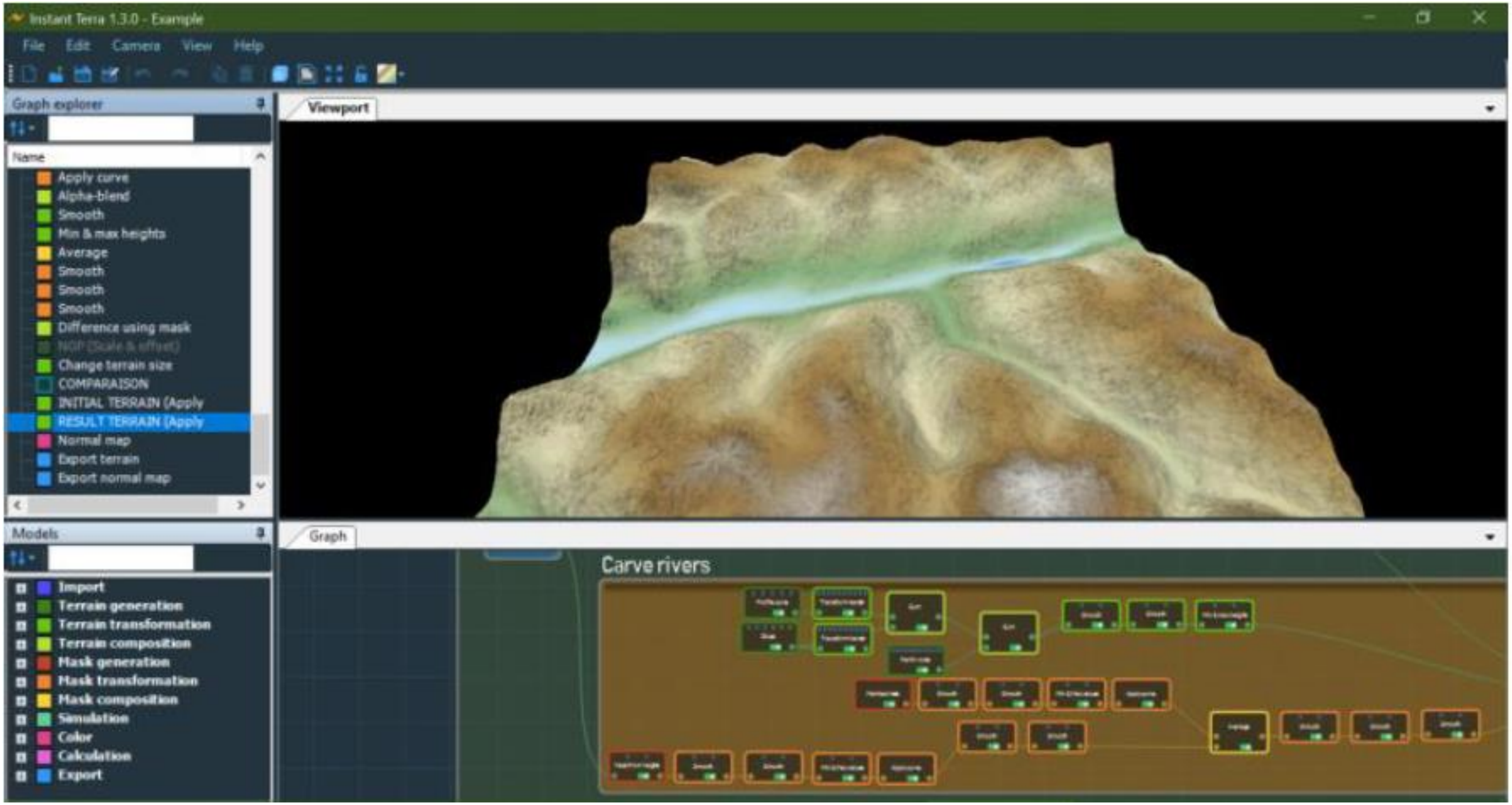


- The red channel deals with large changes in the terrain (say hundreds of metres).
  The green channel deals with medium values (tens of metres).
  The blue channel deals with low values (metre/sub-metre level)

Producing Height Maps

- Image Editor
  - The manual route
- Height Map Editors
- Terrain generators. For example:
  - InstantTerra (www.wysilab.com/)
  - Terragen (www.planetside.co.uk)
  - WorldMachine (www.world-machine.com)

Example: InstantTerra



Generating Terrain in Real Time

- Uses different algorithms to dynamically generate terrain
  - Deformable
  - Dynamic level of detail

Procedural Terrain Generation

- Many techniques available:
  - Fractal Terrain Generation
    http://web.mit.edu/cesium/Public/terrain.pdf
  - Using coherent noise
    e.g. Perlin Noise (https://flafla2.github.io/2014/08/09/perlinnoise.html)

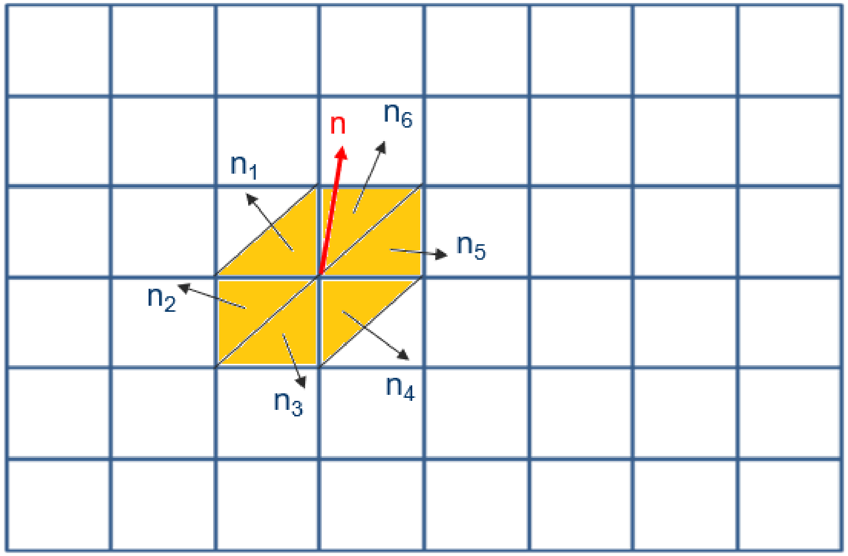
Optimising Polygon Count

- A terrain will have a lot of polygons, e.g. a grid of 1024 x 1024 vertces will have

  1023 x 1023 x 2 = 2093058 triangles (polygons)
- This is a small terrain although having a high number of vertices.
- There are algorithms available that can reduce the polygon count. These work on reducing the number of polygons where the terrain is not changing much (flat areas or steep slopes).
- This is beyond the scope of this module. However, one such technique can be found here: http://www.gamasutra.com/view/feature/3434/continuous_lod_terrain_meshing_.php

Terrain Lighting

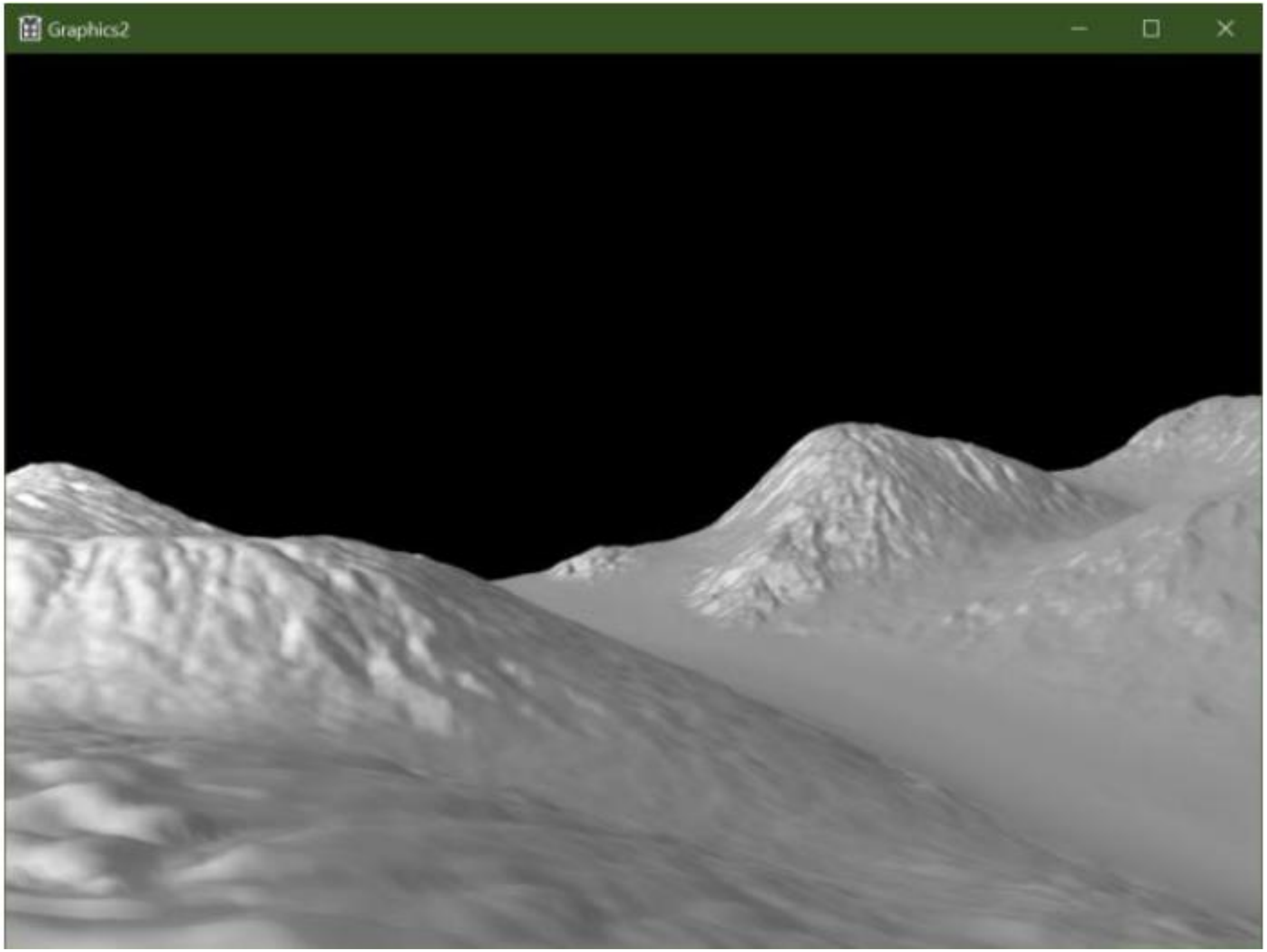
- If using lighting calculations in the shader (e.g. directional light), we need to generate normals for the grid once the height map has been applied
- To do this, we need to calculate the face normal for each triangle in the grid and then add this to every vertex affected by this face.
- Finally, we normalise each vertex normal



$$\mathbf{n} = \frac{\sum_{i=1}^{k} \mathbf{n}_i}{k}$$

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{|\mathbf{n}|}$$

After Normal Generation



Texturing Terrain: The Simple Approach

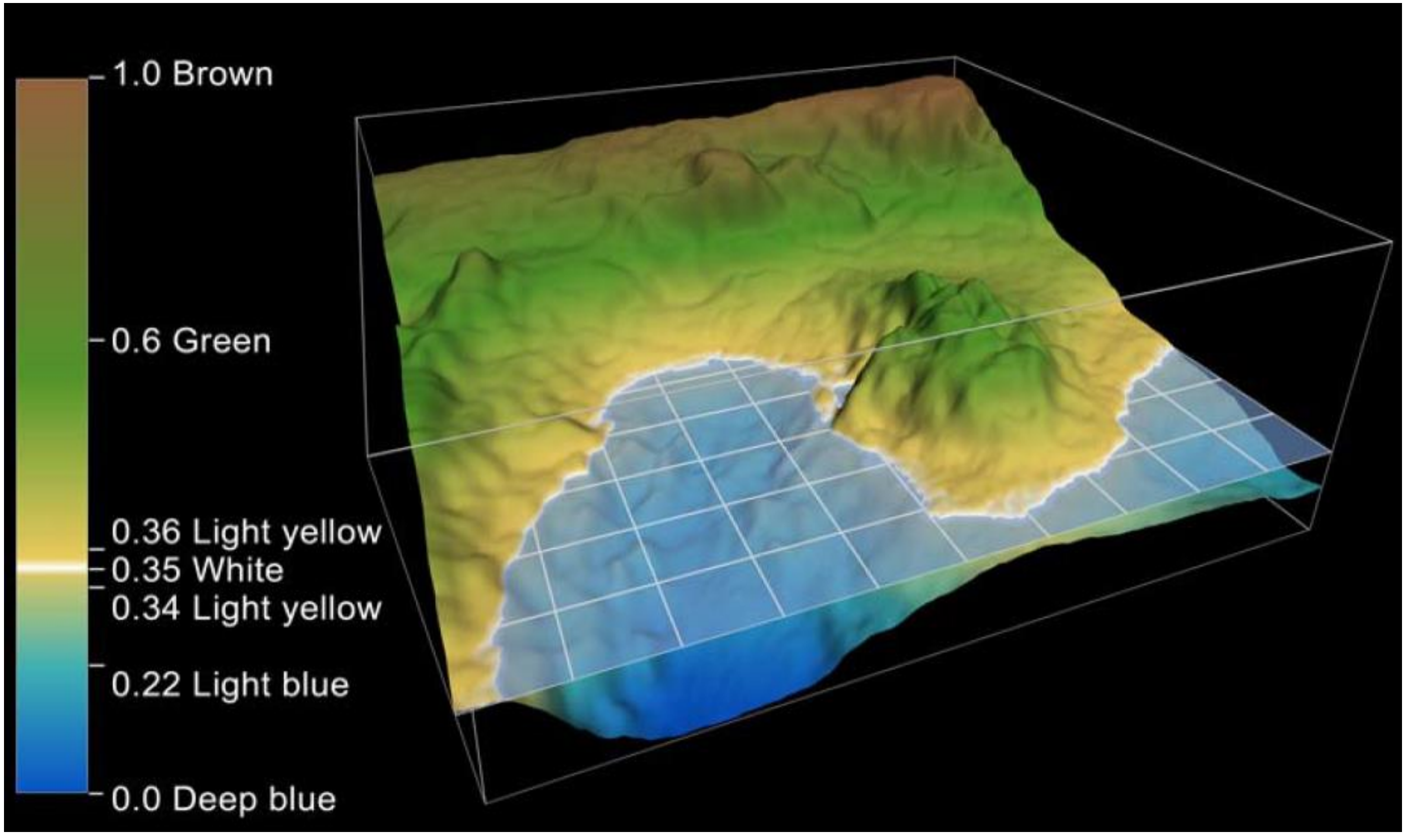
- The simplest way of texturing a terrain is to take a large texture map that has been generated by a terrain generator
- Stretched over entire grid
- Each vertex has UV values in the range 0.0 to 1.0
- Take the width/height of texture image and divide by number of squares in the grid
- Not ideal, especially closeup.

Texturing Terrain using Colormaps

Colormaps:

- Is a look-up table of colors corresponding to specific sorted intensity values (scalar field)
- An input intensity that matches one of the table records is directly mapped to the associated color
- Other values are interpolated from the closest table entries
- Used to go from intensity/gray scale to an arbitrary color gradient

Texturing Terrain using Colormaps



Color coding of land height and sea depth, using a color map that maps relative (normalized) **height information** onto interpolated color values.

[Theoharis et al., 2008]

Texturing Terrain using Blend-maps

- Another solution is to generate a blend-map and apply multiple textures to the squares in the grid. The blend-map determines how much of each texture contributes to the final result
- Each vertex has two sets of texture coordinates – one into the blend map and one into the textures that are applied to each square.
- This is the route we will take.
- But, before we do this, we should take a look at creating a first-person camera that we can move around the terrain.
- We will do this in next step.

This Week

- The tutorial goes through the steps involved in creating a terrain node
- This is a LOT of work that requires careful thought
- It is important though. Your final submission for this module will be expected to demonstrate some rendering of terrain, at least going as far as the material covered this week.