// ProceduralTerrain

```
[Range( 5,  250)] public int CellSize      =  10;

[Range(1,  20 )] public int   Octaves      = 5;
[Range(1f, 30f)] public float Scale        = 3f;
[Range(0f,  1f)] public float Persistance = 0.5f;
[Range(0f,  4f)] public float Lacunarity  = 2f;

private static int TerrainsGenerated = 0;
```

There is more to Lacunarity - which we will cover when stepping through the height equations

// ProceduralTerrain

```
float height11 = 0f;

float amplitude = 1f;
float frequency = 1f;

for (int i = Octaves; i > 0; i--) {
  float octave_x0 =  x       / Scale * frequency;
  float octave_z0 =  z       / Scale * frequency;
  float octave_x1 = (x + 1f) / Scale * frequency;
  float octave_z1 = (z + 1f) / Scale * frequency;

  height00 += Mathf.PerlinNoise(octave_x0, octave_z0) * amplitude;
  height01 += Mathf.PerlinNoise(octave_x0, octave_z1) * amplitude;
  height10 += Mathf.PerlinNoise(octave_x1, octave_z0) * amplitude;
  height11 += Mathf.PerlinNoise(octave_x1, octave_z1) * amplitude;

  amplitude *= Persistance;
  frequency *= Lacunarity;
}

int x0 =  x      * CellSize;
```