

// 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 Persistence = 0.5f;
```

```
[Range(0f, 4f)] public float Lacunarity = 2f;
```

```
private static int TerrainsGenerated = 0;
```

Works similarly, but opposite, to
Scale

// 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 Persistence = 0.5f;
```

```
[Range(0f, 4f)] public float Lacunarity = 2f;
```

Larger values “zoom out”

```
private static int TerrainsGenerated = 0;
```