

## // ProceduralTerrain

```
[Range(10, 1000)] public int TerrainSize;  
[Range( 2, 100)] public int TerrainHeight;  
[Range( 5, 250)] public int CellSize;
```

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

```
...
```

```
for (int x = 0; x < x_segments; x++) {  
    for (int z = 0; z < z_segments; z++) {  
        ...  
    }  
}
```

```
Mesh mesh = new Mesh { name = $"Procedural Terrain {++TerrainsGenerated}" };
```

```
mesh.SetVertices(vertices);  
mesh.SetTriangles(triangles, 0);
```

```
GetComponent<MeshFilter>().mesh = mesh;
```

A handy property  
to keep track of  
how many terrains  
we've created

```
// ProceduralTerrain
```

```
[Range(10, 1000)] public int TerrainSize;  
[Range( 2, 100)] public int TerrainHeight;  
[Range( 5, 250)] public int CellSize;
```

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

```
...
```

```
for (int x = 0; x < x_segments; x++) {  
    for (int z = 0; z < z_segments; z++) {  
        ...  
    }  
}
```

The mesh for our Terrain

```
Mesh mesh = new Mesh { name = $"Procedural Terrain {++TerrainsGenerated}" };
```

```
mesh.SetVertices(vertices);  
mesh.SetTriangles(triangles, 0);
```

```
GetComponent<MeshFilter>().mesh = mesh;
```