

```
// ProceduralTerrainEditor
```

...                    Add a field to edit TerrainHeight  
                      (we'll add that to our ProceduralTerrain next)

```
EditorGUILayout.PropertyField(serializedObject.FindProperty("TerrainSize"));
EditorGUILayout.PropertyField(serializedObject.FindProperty("TerrainHeight"));
EditorGUILayout.PropertyField(serializedObject.FindProperty("CellSize"));
```

```
...
```

```
// ProceduralTerrain
```

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

The property  
we just added  
to the editor

```
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;
```