

```

using UnityEditor;
using UnityEngine;

[CustomEditor(typeof(ProceduralTerrain))]
[CanEditMultipleObjects]
public class ProceduralTerrainEditor : Editor {

    public override void OnInspectorGUI() {
        serializedObject.Update();

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

        if (GUILayout.Button("Generate"))
            (serializedObject.targetObject as ProceduralTerrain).GenerateTerrain();

        serializedObject.ApplyModifiedProperties();
    }
}

```

When that button  
is clicked  
generate terrain

```
using UnityEditor;
using UnityEngine;

[CustomEditor(typeof(ProceduralTerrain))]
[CanEditMultipleObjects]
public class ProceduralTerrainEditor : Editor {

    public override void OnInspectorGUI() {
        serializedObject.Update();

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

        if (GUILayout.Button("Generate"))
            (serializedObject.targetObject as ProceduralTerrain).GenerateTerrain();

        serializedObject.ApplyModifiedProperties();
    }
}
```

Save updated property values