// ProceduralTerrain

```
int z_segments = TerrainSize / CellSize;

int vertex_count = 6 * x_segments * z_segments;
List<Vector3> vertices  = new List<Vector3>(new Vector3[vertex_count]);
List<int>     triangles = new List<int>    (new int    [vertex_count]);

for (int x = 0; x < x_segments; x++) {
```

Each quad
is composed
of two triangles

Each triangle
is composed
of three vertices

```
// ProceduralTerrain
  for (int x = 0; x < x_segments; x++) {
    for (int z = 0; z < z_segments; z++) {

      …

      var vertex11 = new Vector3(
        (float) x1, height11 * (float) TerrainHeight, (float) z1
      );

      int index0 = 6 * (x + z * x_segments);

      int index1 = index0 + 1;
      int index2 = index0 + 2;
      int index3 = index0 + 3;
      int index4 = index0 + 4;
      int index5 = index0 + 5;

      …
```