// ProceduralTerrain

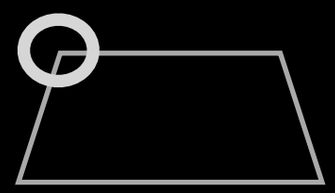
```
for (int x = 0; x < x_segments; x++) {
  for (int z = 0; z < z_segments; z++) {
    float height00 = GetHeight(x + 0f, z + 0f, x_segments, z_segments);
    float height01 = GetHeight(x + 0f, z + 1f, x_segments, z_segments);
    float height10 = GetHeight(x + 1f, z + 0f, x_segments, z_segments);
    float height11 = GetHeight(x + 1f, z + 1f, x_segments, z_segments);

    int x0 =  x      * CellSize;
    int z0 =  z      * CellSize;
    int x1 = (x + 1) * CellSize;
    int z1 = (z + 1) * CellSize;

    var vertex00 = new Vector3(
      (float) x0, height00 * (float) TerrainHeight, (float) z0
    );
    var vertex01 = new Vector3(
      (float) x0, height01 * (float) TerrainHeight, (float) z1
    );
    var vertex10 = new Vector3(
      (float) x1, height10 * (float) TerrainHeight, (float) z0
    );
    var vertex11 = new Vector3(
      (float) x1, height11 * (float) TerrainHeight, (float) z1
    );
  }
}
```

This is how we define the four corners of a terrain segment

Left(x = 0)
Far(z = 1)



// ProceduralTerrain

```
for (int x = 0; x < x_segments; x++) {
  for (int z = 0; z < z_segments; z++) {
    float height00 = GetHeight(x + 0f, z + 0f, x_segments, z_segments);
    float height01 = GetHeight(x + 0f, z + 1f, x_segments, z_segments);
    float height10 = GetHeight(x + 1f, z + 0f, x_segments, z_segments);
    float height11 = GetHeight(x + 1f, z + 1f, x_segments, z_segments);

    int x0 =  x      * CellSize;
    int z0 =  z      * CellSize;
    int x1 = (x + 1) * CellSize;
    int z1 = (z + 1) * CellSize;

    var vertex00 = new Vector3(
      (float) x0, height00 * (float) TerrainHeight, (float) z0
    );
    var vertex01 = new Vector3(
      (float) x0, height01 * (float) TerrainHeight, (float) z1
    );
    var vertex10 = new Vector3(
      (float) x1, height10 * (float) TerrainHeight, (float) z0
    );
    var vertex11 = new Vector3(
      (float) x1, height11 * (float) TerrainHeight, (float) z1
    );
  }
}
```

This is how we define the four corners of a terrain segment

Right(x = 1)
Near(z = 0)