Perlin Simple Waves

Public variables:

-*heightScale*, determines how terrain will be generated in terms of height.
-*detailScale*, determines how smooth the surface on the top will look like when it generates.
-*waveSpeed*, determines how fast the slopes will be exchanging to form a wave look-like [smaller value – faster the waves, bigger value – slower the waves]
-*waves*, is a boolean value, if enabled it will generate waves and move the mesh and collider correspondingly, if disabled it will be on place and generate once

Private variables:

-*mesh*, stores a Component of type MeshFilter which is used to apply deformed mesh later
-*vertices*, is an array of Vector3 to store the vertices of the mesh/collider
-*counter*, is used to increase by 1 everytime a vertecie is passed to make it look like a double array
-*zLevel*, increases everytime for loop passes thorugh to lower the vertices below top so they all move accordingly

MyMethod, is called form anywhere if wanted, can be called form *Start* to have it as non-moving object, or from *Update* so you can live-time change the properties.
In this method we grab the mesh form the object, and since plane has 11x11 vertices we will drag it thought two *for-loops* 11 times which on the end is 121, exact same as array size of, vertices variable.
After the *for-loop,* we simply apply to our mesh the new vertices which were re-calculated in the CalculationMethod.

CalculationMethod, takes the i = counter which determins the number in the array of vertices, and zLevel = minusZ goes for rows from the array.
PerlinNoise method takes a paramater and returns 0-1 float value based on that we generate our mesh, if you don't know what Perlin is you can look for it here on some of these links....

https://docs.unity3d.com/ScriptReference/Mathf.PerlinNoise.html
https://en.wikipedia.org/wiki/Perlin_noise