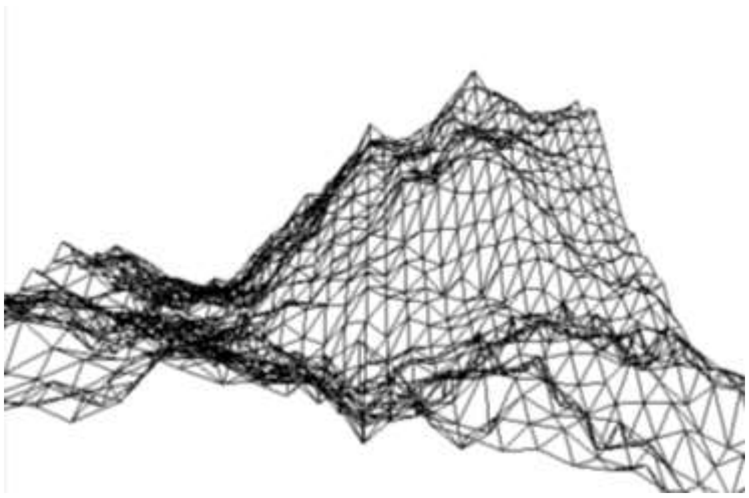
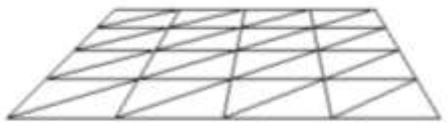
# Advanced Graphics Programming

## Workshop Six (Part B) – Terrain Generation Shaderized

This workshop involves implementing the diamond-square algorithm to procedurally generate terrain as illustrated below:

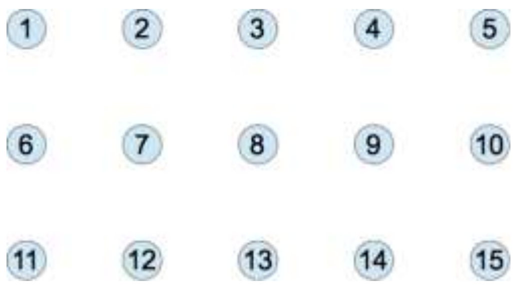


**Step 1:** Download TerrainShaderized.zip from Moodle which contains the TerrainShaderized Visual Studio Solution to get you started with the modern version of OpenGL (4.2). The project uses the libraries: glm-0.9.6.3, glew-1.10.0 and freeglut-MSVC-2.8.1-1 which are available on the lab PCs on the C:\OpenGL drive. The project contains initialisation code to create a window and to draw a flat height map. Compile and run the code and check that you get the grid shown below.

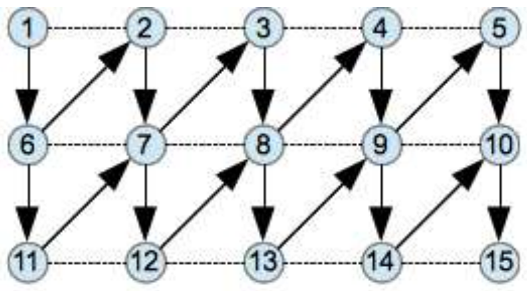


**Step 2:** Review the initialisation routine named `setup` in TerrainGeneration.cpp. The function begins by initialising a height map. Initially this is a small map (5x5) with the height values initialised to zero. Each element (x, z) of the array represents the height (y) of the terrain at that given point. The `terrain` data is stored in a multidimensional array so that it is easier to access and update it when implementing the diamond square algorithm.

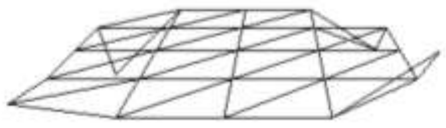
To draw the terrain OpenGL expects a single array of vertex data so the next part initialises the vertex buffer object `terrainVertices`. The vertices of the height map are added to a single dimensional array in the order shown below:



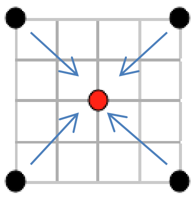
OpenGL also needs to know how to connect the vertices, in this case we want to connect them as a triangle strip so we build index buffer data for each row in our data. This data is stored in a multidimensional array `terrainIndexData` and the connections illustrated below:



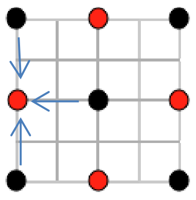
**Step 3:** Initialise the diamond-square algorithm in the `setup` function. Start by setting the corner values of the map to a random number, between -1 and 1. If you use a seed to generate your random numbers, you will be able to change the seed to generate different terrains (e.g `srand(4)`).



**Step 4**: **Diamond step**: Taking a square of four points, generate a random value at the square midpoint, where the two diagonals meet. The midpoint value is calculated by averaging the four corner values, plus a random amount (in the range -1 to 1). Start with the outer four corners of your grid (black points) and generate the value at the centre of the grid (red point) (as shown below):

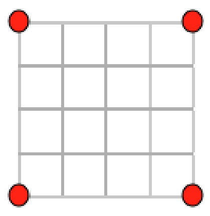


**Step 5**: **Square step**: Calculate the midpoints of the edges between the corner of the diamonds. Calculate the midpoint value by averaging the corner values, plus a random amount generated in the same range as used for the diamond step. To calculate each red point you should take the 4 black points that surround it, in the following example there are only 3 surrounding black points as the 4th in each case is outside of the map. Make sure you check if your points are in the map before averaging them.

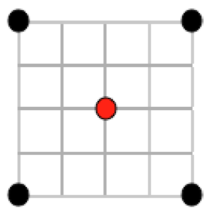


**Step 6:** Repeat the square and diamond steps, iteratively.

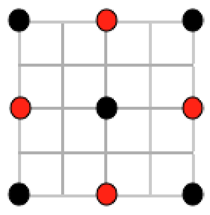
a) initialisation



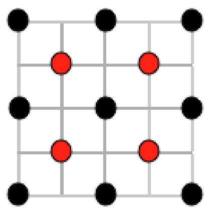
b) diamond step (iteration 1)



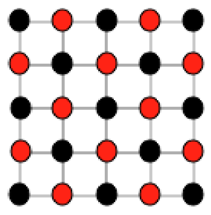
c) square step (iteration 1)



d) diamond step (iteration 2)

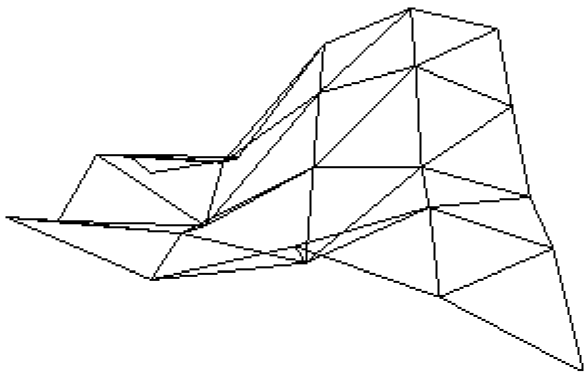


e) square step (iteration 2)

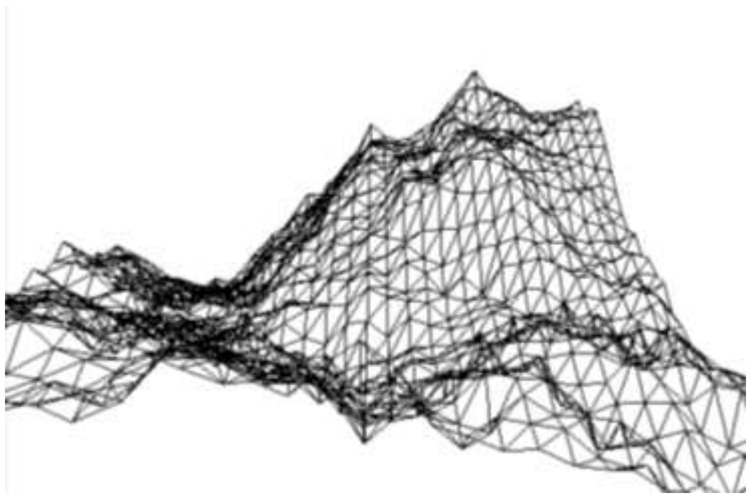


```
map_size = 5
terrain[map_size][map_size]
step_size = map_size – 1
rand_max = 1
H = 1

while step_size > 1

    for (x = 0; x < map_size - 1; x += step_size)
        for (y = 0; y < map_size - 1; y += step_size)
            diamond_step(x, y)

    for (x = 0; x < map_size - 1; x += step_size)
        for (y = 0; y < map_size - 1; y += step_size)
            square_step(x, y)

    rand_max = rand_max * pow(2, -H)
    step_size = step_size / 2
```

You should now get the random generated terrain, shown below.



**Step 7:** Experiment with the parameters to create different terrains. Start by increasing the size of your map. Then experiment with the initial random number range, the random number reduction and the seed value.



In the example above, the size of the map was 33 x 33, the seed value was 4, the initial random number range was 10 to -10 and H was 1. You will also need to move the terrain into view by updating the translate function on the model view.