TERRAIN GENERATION

INTRODUCTION

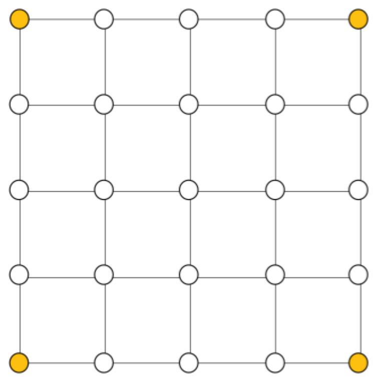
Generating a realistic terrain is a very important part of computer graphics in general. Its is used in many applications like gaming, geology, cinema, simulations etc.., there are many different algorithms used for terrain generation or modelling. But it is easier to classify them into families, like, the fractal based method, the fractal and physical based method, the physical based method and many other miscellaneous methods.

ALGORITHM DESCRIPTION

The fractal models are faster to produce and usually used to produce them from scratch.

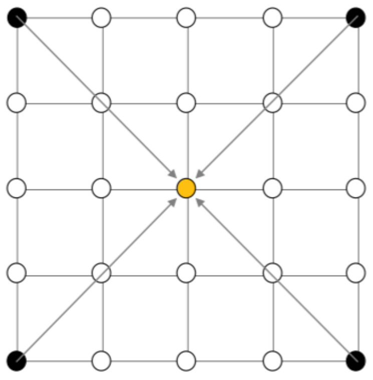
This paper basically describes two approaches to terrain generation, the first is the Midpoint Displacement algorithm or approach which is used to generate terrains from scratch, and the second approach that was described was the Midpoint Displacement Inverse process, which actually uses a already existing set of data elevation maps(DEMs) that are used to generate a smooth terrain.

In this project we are implementing the Midpoint Displacement method of terrain generation using the Diamond-Square algorithm. This algorithm starts of with a basic 2D square array of width and height, first the four corner points are initialized, then there are a alternate sequence of diamond and square methods that are performed.

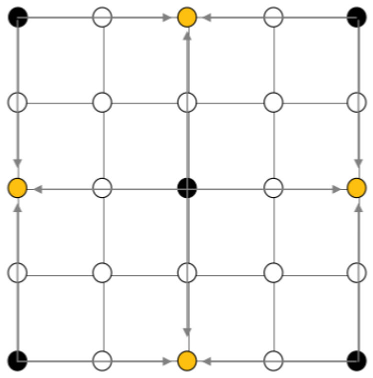


Initialize corner values

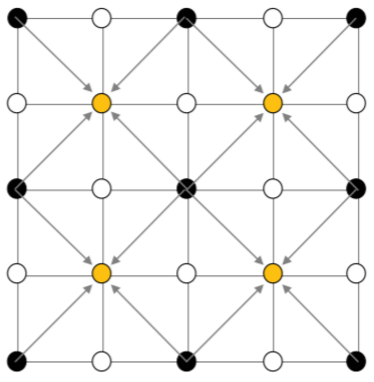
The first step is the diamond step, in which for each and every square, the midpoint of the square is given by the sum of the average of the four corner points and sum random value.

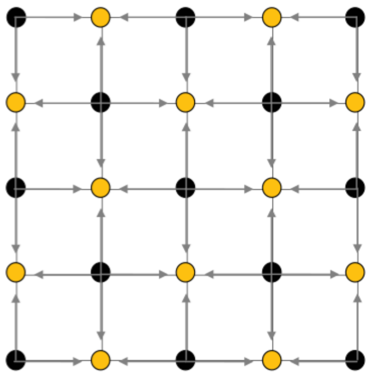

Performdiamondstep

The next step is the Square step after the diamond step, in which the mid point of the diamond formed with the midpoint and the corner points is given by the sum of the average of the four corner points of the diamond and some random value

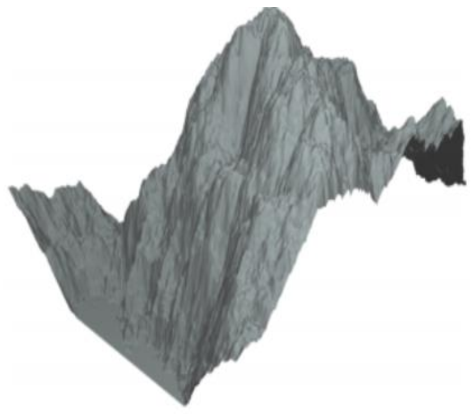
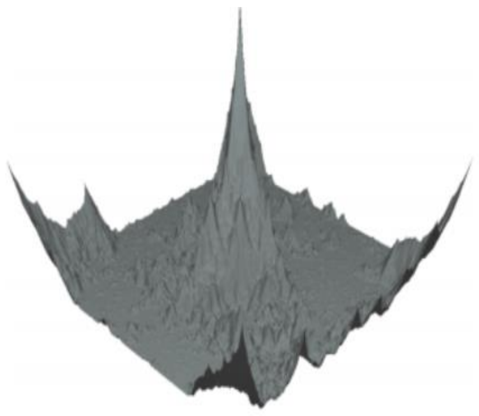

Performsquare step

This steps are carried out alternatively until all the vertices in the array are gone through once.

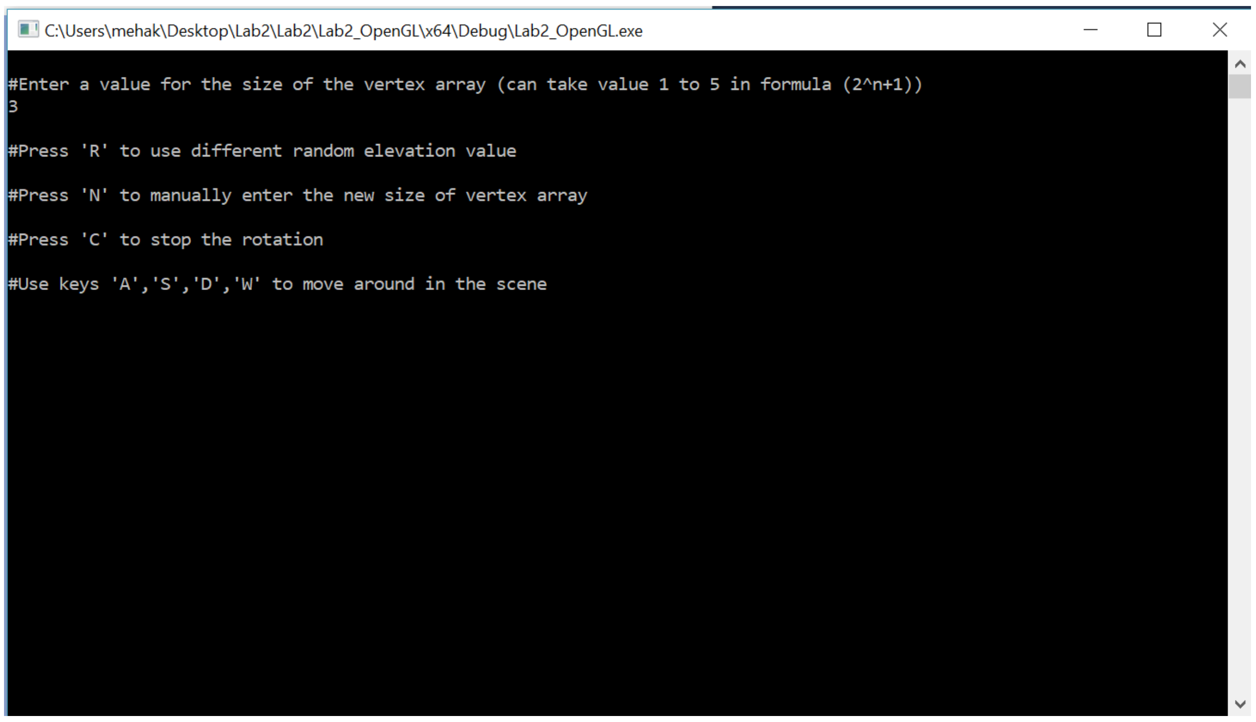

Performdiamond step


Performsquare step

These are some of the results obtained using the above described Midpoint Displacement algorithm, for different random values as the height maps.

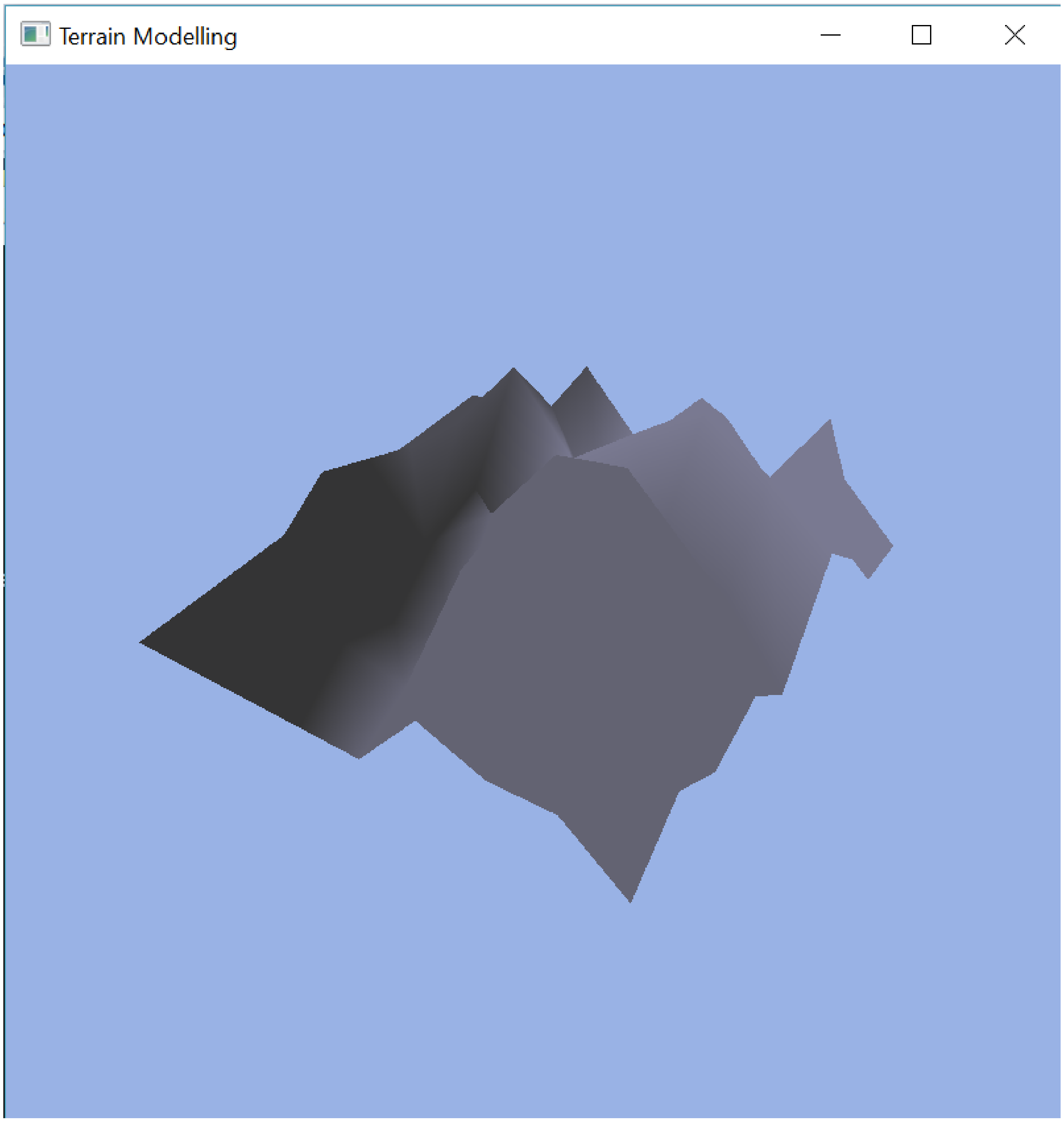



USER INTERFACE:

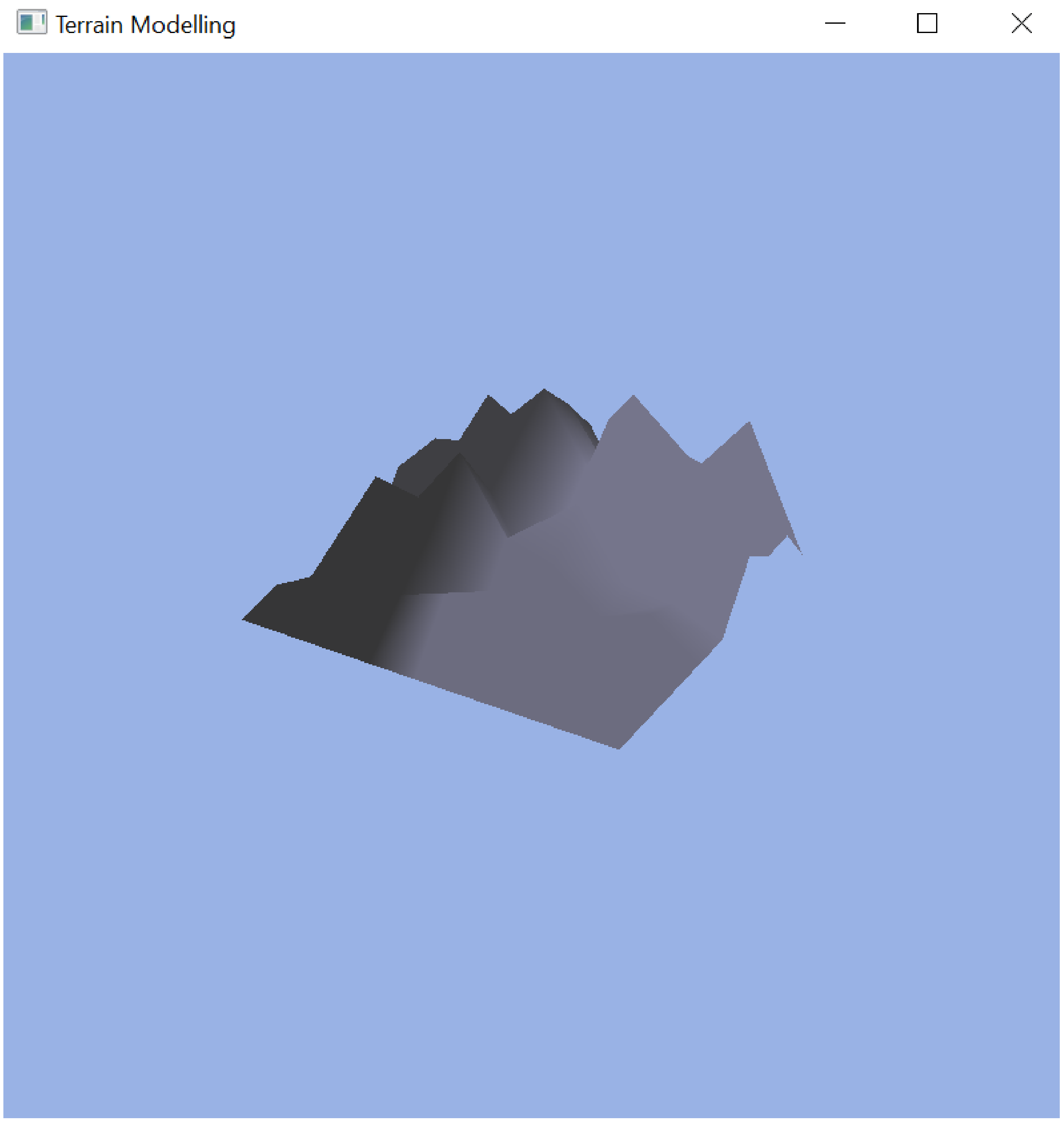


```
C:\Users\mehak\Desktop\Lab2\Lab2\Lab2_OpenGL\x64\Debug\Lab2_OpenGL.exe

#Enter a value for the size of the vertex array (can take value 1 to 5 in formula (2^n+1))
3

#Press 'R' to use different random elevation value

#Press 'N' to manually enter the new size of vertex array

#Press 'C' to stop the rotation

#Use keys 'A','S','D','W' to move around in the scene
```

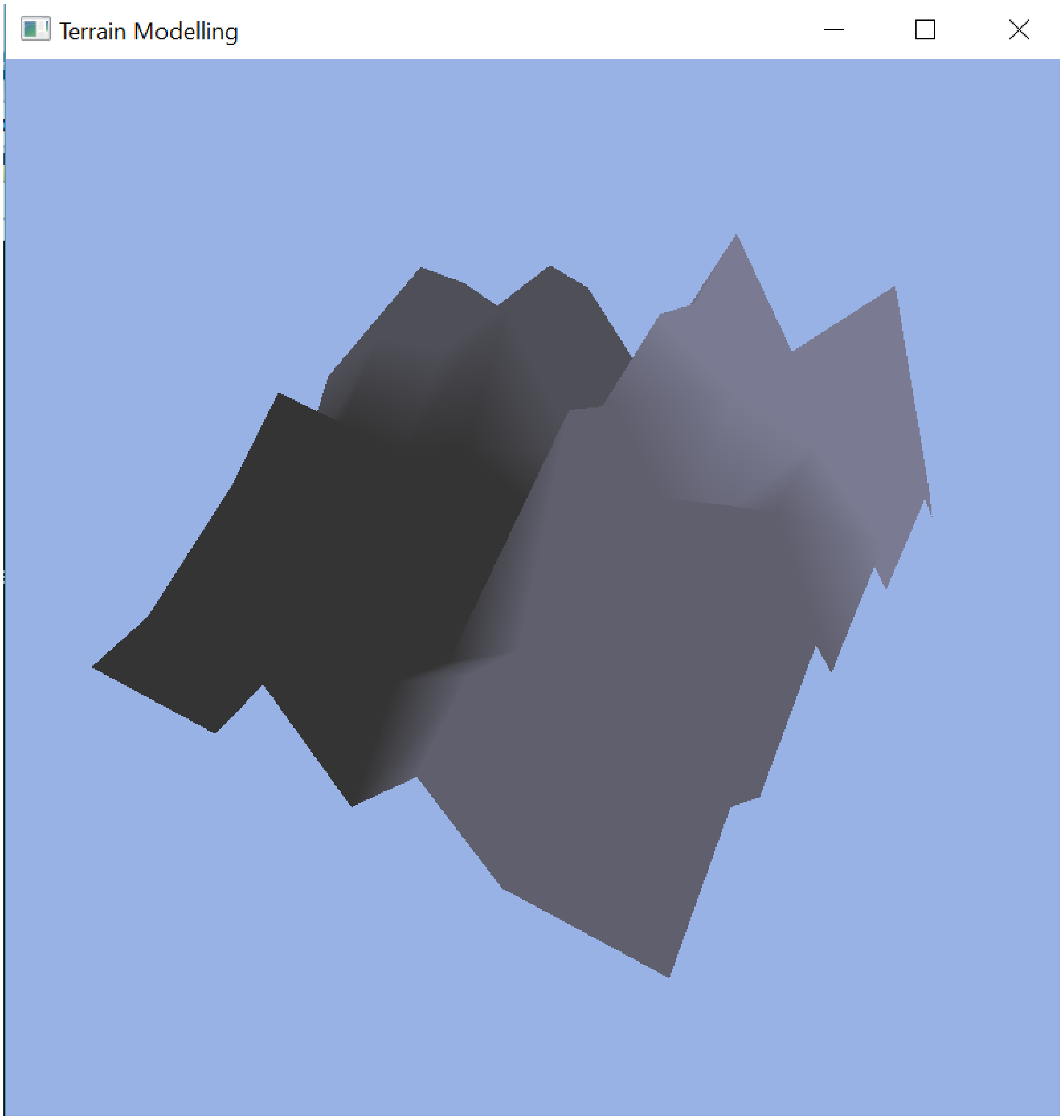
SCREENSHOT 1



SCREENSHOT 2



SCREENSHOT 3



DRAWBACK

The main drawback of this algorithm is the interpolation function: we can not get any control over the interpolation. Thus, specifying curvature constraints on the surface is impossible. This drawback does not affect the terrain generation when ridge lines and rivers networks are precomputed because the interpolation is naturally done between the ridges (high elevations) and the rivers (low elevations). But, for example, when we have either high or low elevation constraints the resulting surface remains flat.