Project for the final exam

Development of a 3D OpenGL core profile application for lava flows visualization on a real topographic surface

Build an interactive 3D application which displays a topographic surface and a lava flow on it. The application must be the most general possible, allowing the visualization of datasets of any size. It must also be possible to display data from different points of view. This can be achieved by means of various transformations, like:
- Rotations around the x, y and z axes, with center of rotation in the center of the data set;
- Translations in the x, y and z directions.

As for lighting, use the Blinn-Phong model and also the Phong Shading. Use just a single point light located to the same camera position. The point light must have both a low-white Ambient and a high-white Diffuse component, while the surface color must be white. Lava flow color must be in red scale (as described below). A texture must also be applied to the topographic surface.

The complete dataset to be visualized can be found in the " dataset_lava" directory and is composed by the following files (other example data can be found in the "data" directory):

- topographic surface (altitudes.dat)
- lava thickness (lava.dat)
- lava temperature (temperature.dat)
- texture of the topographic surface (texture.tif)

Accordingly, data to be represented are: topographic altitude, lava thickness and lava temperature. Besides the texture, here provided in tif format (please, feel free to convert the texture in other format), data is characterized by the following format

```
ncols         10
nrows         10
xllcorner     2487289.5023187
yllcorner     4519797.0771414
cellsize      10.0
NODATA_value  -9999
9.00 8.00 7.00 6.00 5.00 4.00 3.00 2.00 1.00 0.00
8.00 7.00 6.00 5.00 4.00 3.00 2.00 1.00 0.00 1.00
7.00 6.00 5.00 4.00 3.00 2.00 1.00 0.00 1.00 2.00
6.00 5.00 4.00 3.00 2.00 1.00 0.00 1.00 2.00 3.00
5.00 4.00 3.00 2.00 1.00 0.00 1.00 2.00 3.00 4.00
4.00 3.00 2.00 1.00 0.00 1.00 2.00 3.00 4.00 5.00
3.00 2.00 1.00 0.00 1.00 2.00 3.00 4.00 5.00 6.00
2.00 1.00 0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00
1.00 0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00
0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00
```

The above example is about a simplified topographic surface (DEM_test.dat in the "data" directory – DEM is for Digital Elevation Model). As you can see, data is composed by a header (first 6 rows) and a matrix of values.

Each matrix element represents the topographic altitude, the lava thickness or the lava temperature characterizing a square region of the dataset (called *cell* in the following). In the specific example, the cell is 10m x 10m sized, as specified by the `cellsize` value in the header, and the value represents an altitude (in meters).

The different header's information have the following meaning.

```
ncols         10
nrows         10
```

represent the number of columns and rows of the matrix.

```
xllcorner     2487289.5023187
yllcorner     4519797.0771414
```

specify the geographical coordinates of the lower left corner of the surface (they can be ignored for the purpose of this project).

```
cellsize      10.0
```

specifies that the discretization of the surface is characterized by (square) cells of 10m side

```
NODATA_value  -9999
```

The `NODATA_value`, if present, indicates the lack of data. Nodata values must not be represented.

Fig. 1a shows a graphic representation of the above topographic surface. It is based on a straightforward color mapping technique, which simply assigns the color black to the cell with the lowest altitude, white to the one with the highest elevation and grey tones to cells having intermediate altitudes. Fig. 1b shows a bigger surface represented by using the same technique (DEM_Albano.asc in the “data” directory).

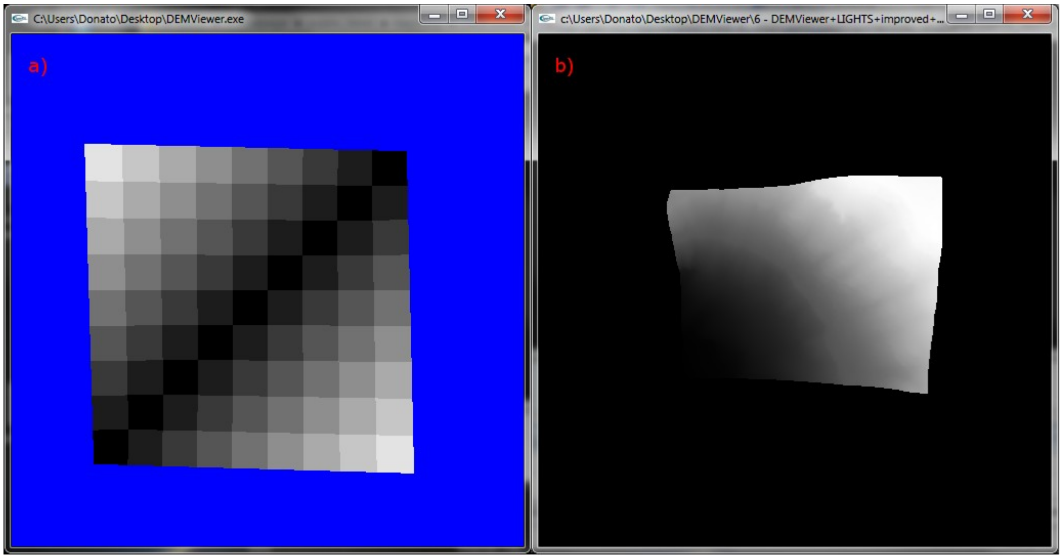
C:\Users\Donato\Desktop\DEMViewer.exe
a)
c:\Users\Donato\Desktop\DEMViewer\6 - DEMViewer+LIGHTS+improved+...
b)

Fig. 1. a) Representation of the example topographic data. b) Representation of a bigger topographic surface.

As a first preliminary step, simply represent the topographic surface by using triangle strips, a triangle strip for each row of the matrix. We will refer such a representation as grid representation. The color of the resulting cells has to be assigned by considering the altitude of the upper left point. In this way, a matrix of n*m points produces (n-1)*(m-1) cells, as illustrated in Fig. 2a. Pay attention to the proper management of NODATA_value, that should not be displayed.

As a second step, draw the surface representing a cell for each point in the matrix. The cell must be built around the known point, as shown in Figure 2b. In this way the color of the cell can be assigned by simply considering the elevation of the cell itself. Note that, using the cell-based representation a matrix of n*m points produces exactly n*m cells. Again, pay attention to the proper management of NODATA_value, that should not be displayed.

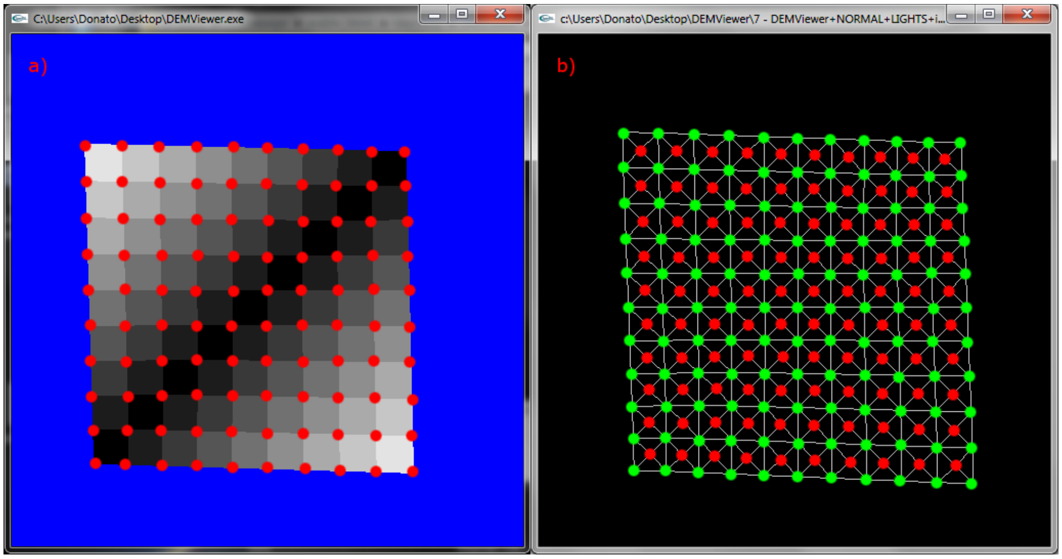
C:\Users\Donato\Desktop\DEMViewer.exe
a)
c:\Users\Donato\Desktop\DEMViewer\7 - DEMViewer+NORMAL+LIGHTS+i...
b)

Fig. 2. a) Grid-based representation of the example surface, where known altitudes are evidenced by red points. b) Cell-based representation of the same surface (known altitudes are still evidenced in red). A cell, made by 4 further vertices (in green), is built around each known point.

Once you have reached the first – preliminary - goal, you can add transformations, light and also apply the texture.

When considering the lighting model, you can avoid to explicitly assign a color to the cell. Instead, you can use the white color for all the cells and let OpenGL evaluate the resulting color on the basis of the cells' orientation with respect to the light direction (please, note that, at this purpose, you need to evaluate at least a normal per cell). You should obtain something like in the result shown in Fig. 3.

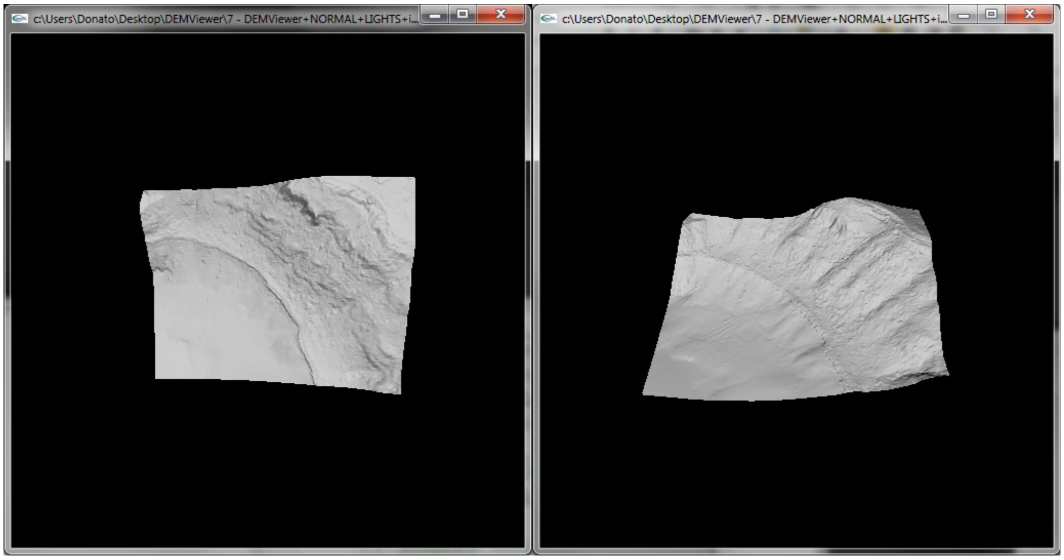
c:\Users\Donato\Desktop\DEMViewer\7 - DEMViewer+NORMAL+LIGHTS+i...
c:\Users\Donato\Desktop\DEMViewer\7 - DEMViewer+NORMAL+LIGHTS+i...

Fig. 3. Representation of the topographic surface with a light point placed in the same camera location.

As regards the lava flow, both the lava thickness and temperature must be properly represented. Such information are provided by two different input files. For the lava thickness, it can be simply visualized by adding the thickness to the surface elevation cell by cell. Moreover, in order to represent temperature, a straightforward red scale color-mapping can be considered, which simply assigns the color (0,0,0) to the cell with the lowest temperature, the color (1, 0, 0) to the to the cell with the highest temperature, and an intermediate color to the other cells. If you prefer, you can also use a color scale ranging from the yellow (for the highest cell) to the red (for the lowest). The final result should be similar to the one shown in Fig. 4.

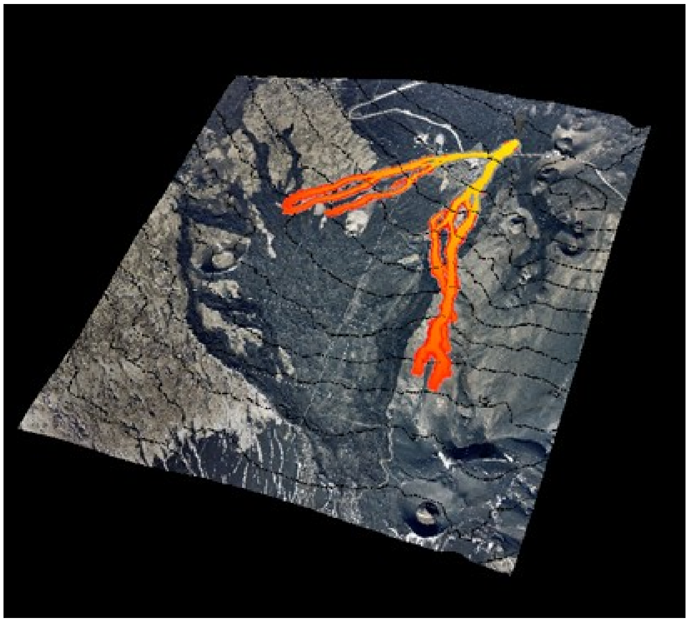

Fig. 4. Final expected result.