LEI / LCC
Departamento de Informática
Universidade do Minho

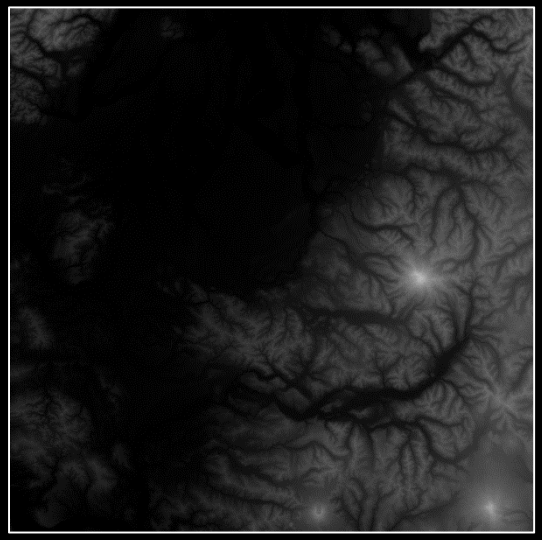
Generating Geometry

Terrains

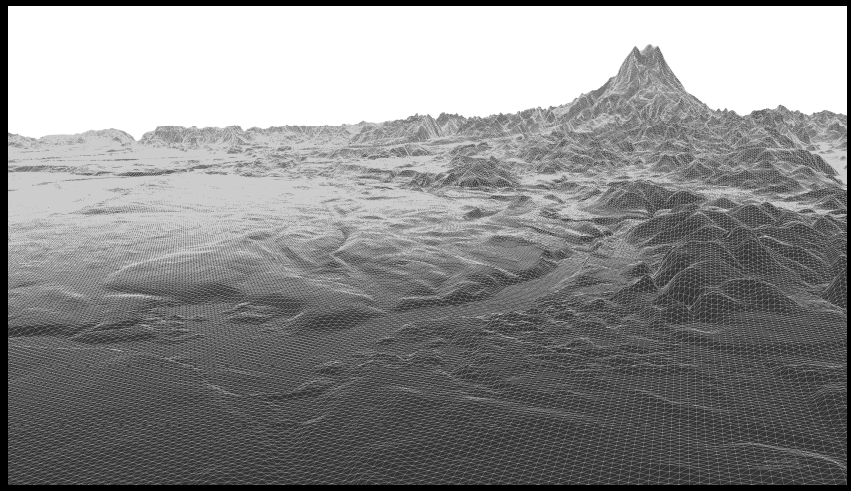
Height Maps

Pixel intensity represents height in a grid

Height map

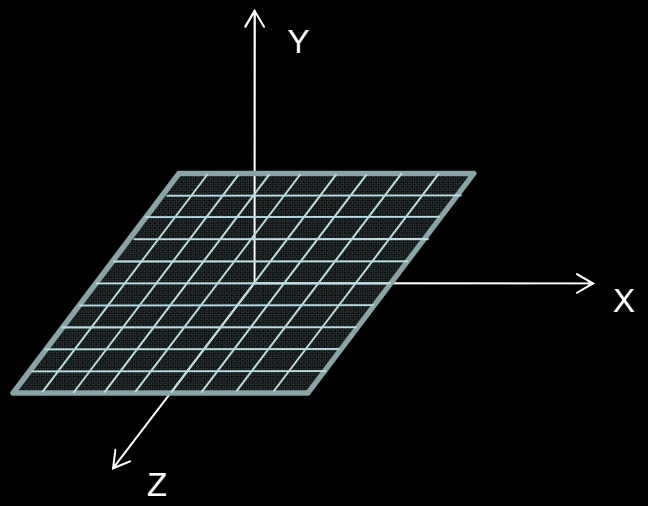


Rendering



Terrains from Images

- Goal:
  - Given an image, create a regular grid such that the height of each grid point matches the corresponding pixel's intensity.
- Tasks:
  - Load the image
  - Create the grid based on the matrix of pixels read from the image.



DevIL

- Cross-platform image loading library
  (http://openil.sourceforge.net/)

In the cpp file:

```
#include <IL/il.h>
// Note: in Linux and Macs it may be required to specify the full path to il.h
```

After GLUT callback registration do:

```
ilInit();
```

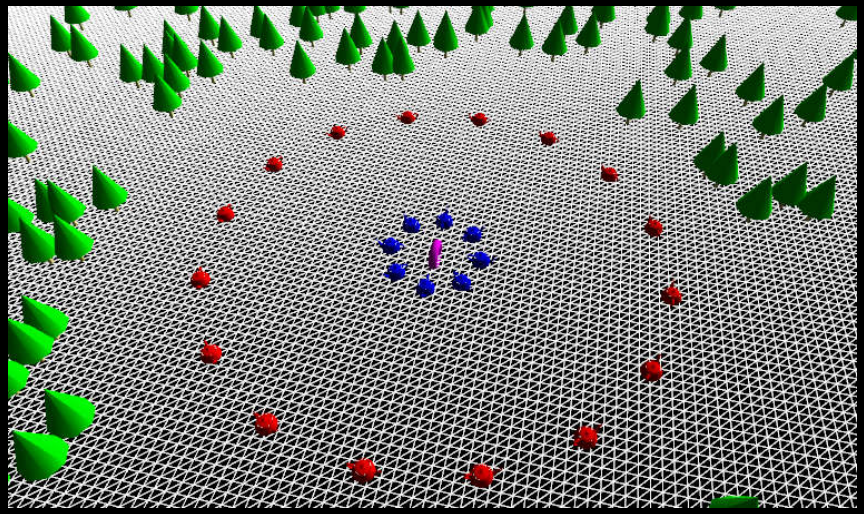
- Other Image Libraries can be found here:
  https://www.opengl.org/wiki/Image_Libraries

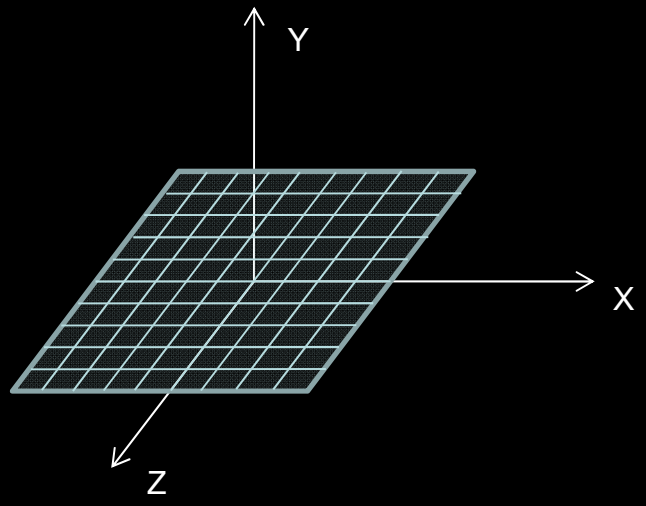
DevIL – Loading an Image

```
unsigned int t, tw, th;
unsigned char *imageData;

ilGenImages(1,&t);
ilBindImage(t);
// terreno.jpg is the image containing our height map
ilLoadImage((ILstring)"terreno.jpg");
// convert the image to single channel per pixel
// with values ranging between 0 and 255
ilConvertImage(IL_LUMINANCE, IL_UNSIGNED_BYTE);

// important: check tw and th values
// both should be equal to 256
// if not there was an error loading the image
// most likely the image could not be found
tw = ilGetInteger(IL_IMAGE_WIDTH);
th = ilGetInteger(IL_IMAGE_HEIGHT);
// imageData is a LINEAR array with the pixel values
imageData = ilGetData();
```

Terrains from Images

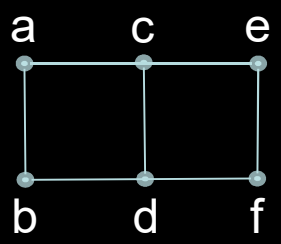
- Step 1:
  - Build a regular grid with height 0.0 for every point. Use the with and height of the image as the input dimensions for the grid:
    - there are as many vertices in the grid as pixels in the image





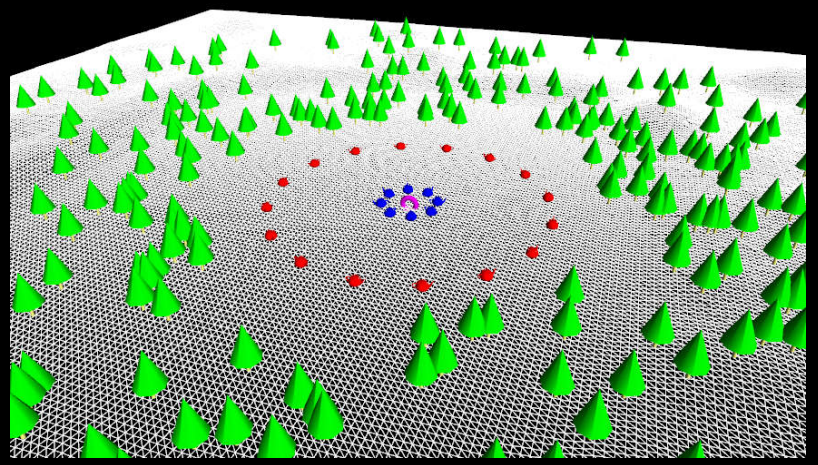
Triangle Strips

- Array with triangle vertices: {a,b,c,d,e,f}
- `glDrawArrays(GL_TRIANGLE_STRIP, first, count)`
- Number of strips = image height - 1



Terrains from Images

- Step 2:
  - Set the height for the grid's vertices according to the pixels intensity
    - `float h(int i, int j)` function to return the value of pixel in column `i`, row `j` (required to build the terrain geometry).
    - Note: scale the heights from 0-255 (pixel intensity) to 0 - 30 meters for a more appropriate rendering
    - Note: this lesson is only focused on the terrain, not on the trees and teapots.



VBO - Init

- Step 1 – Allocate and fill arrays with vertices

```
// array for vertices
float *vertexB;
// fill arrays with vertex values
```

- Step 2 - Enable Buffers

```
glEnableClientState(GL_VERTEX_ARRAY);
```

- Step 3: Generate Vertex Buffer Objects

```
GLuint buffers[1];
...
glGenBuffers(1, buffers);
glBindBuffer(GL_ARRAY_BUFFER,buffers[0]);
glBufferData(GL_ARRAY_BUFFER,arraySize, vertexB, GL_STATIC_DRAW);
```

VBO - Render

- Step 1: Define the semantic for each buffer

```
glBindBuffer(GL_ARRAY_BUFFER,buffers[0]);
glVertexPointer(3,GL_FLOAT,0,0);
```

- Step 2 : Draw VBOs

```
glDrawArrays(GL_TRIANGLE_STRIP, first, count);
```

  - first – the starting index
  - count – the number of vertices (not triangles) to draw

Assignment

- Given an image, interpret it as a height map, and generate the corresponding terrain.
  - Add the initialization part to the `init()` function
  - Add the terrain rendering part to `drawTerrain()` function

DevIL (Linux & Mac)

- Install Linux

```
sudo apt-get install libdevil-dev
```

- Install MacOS

```
brew install devil
```