Mipmapping in OpenGL

Steps to Add Mipmapping to Any Rendering System:

1. Create and load the smaller averaged images.
2. Tell the renderer (in this case, OpenGL) how to use them.

Creating/Loading Mipmap Levels in OpenGL:

- For **Step 1** above, there's two choices: Create & load mipmaps manually or have OpenGL do it automatically.
- To create and load manually, you must:
  1. Manually average 4×4 texel regions of the texture, store the reduced-size images as new image (e.g., PPM) files.
  2. Load each level into OpenGL. If these are stored in an array of images *MIPLevels[]*, with MIPLevels[0] being the original image of 64×64 unsigned-byte RGBA pixels, then you would load them as follows:

     glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 64, 64, 0, GL_RGBA, GL_UNSIGNED_BYTE, MIPLevels[0]);
     glTexImage2D(GL_TEXTURE_2D, 1, GL_RGBA, 32, 32, 0, GL_RGBA, GL_UNSIGNED_BYTE, MIPLevels[1]);
     glTexImage2D(GL_TEXTURE_2D, 2, GL_RGBA, 16, 16, 0, GL_RGBA, GL_UNSIGNED_BYTE, MIPLevels[2]);
     glTexImage2D(GL_TEXTURE_2D, 3, GL_RGBA, 8, 8, 0, GL_RGBA, GL_UNSIGNED_BYTE, MIPLevels[3]);
     glTexImage2D(GL_TEXTURE_2D, 4, GL_RGBA, 4, 4, 0, GL_RGBA, GL_UNSIGNED_BYTE, MIPLevels[4]);
     glTexImage2D(GL_TEXTURE_2D, 5, GL_RGBA, 2, 2, 0, GL_RGBA, GL_UNSIGNED_BYTE, MIPLevels[5]);
     glTexImage2D(GL_TEXTURE_2D, 6, GL_RGBA, 1, 1, 0, GL_RGBA, GL_UNSIGNED_BYTE, MIPLevels[6]);

- To create and load automatically, you would:
  1. Load the original texture into an array (let's call it *origImage*), just as if you were texturing without mipmapping.
  2. Tell OpenGL to automatically averate the 4×4 regions and store them in the texture using:

     gluBuild2DMipmaps(GL_TEXTURE_2D, GL_RGBA, 64, 64, GL_RGBA, GL_UNSIGNED_BYTE, origImage);

Using Mipmaps in OpenGL:

- To use MIP-maps in OpenGL, all you need to do is set the minification filter usign glTexParameteri() function.
- glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, <*value*>);
  - Without mipmapping, <*value*> was either GL_NEAREST or GL_LINEAR.
  - Using mipmaps, there are 4 new choices:
    * GL_NEAREST_MIPMAP_NEAREST (use the nearest neighbor in the nearest mipmap level)
    * GL_NEAREST_MIPMAP_LINEAR (linearly interpolate in the nearest mipmap level)
    * GL_LINEAR_MIPMAP_NEAREST (use the nearest neighbor after linearly interpolating between mipmap levels)
    * GL_LINEAR_MIPMAP_LINEAR (linearly interpolate both the mipmap levels and at between texels)