CG Programming - S0006E - 2016
Week 3 - Texturing

3D and Texturing

1. Extend to 3D

   - Add another (static) function to the *MeshResource* that will create a cube instead of a quad (add 4 more vertices and the corresponding faces). You should add texture coordinates as well (see task 3).
   - Implement cleanup of all the buffer objects in the destructor of the *MeshResource*
   - Add rotation around the other axes to your demo program.

2. Create a new *TextureResource* class for loading images from files and registering them with OpenGL

   - Add a new *TextureResource*. It should contain a function for loading from a file (e. g. LoadFromFile(const char * filename)) that will:
     - Load the file using *stb_image.h* (`https://github.com/nothings/stb/blob/master/stb_image.h`)
     - after the file was parsed successfully create an OpenGL texture object and configure it to use mipmapping.
   - Add a function for activating the previously generated texture (bind it)
   - The destructor has to cleanup the texture object and other resources properly.

   Hint: The only necessary component required to use the texture later on is the textureID (similar to vertex buffers).

3. Use the *TextureResource* designed earlier

   - Add texture coordinates to the quad vertex buffer used in the earlier task.
   - Modify the vertex shader to use the texture coordinate buffer as well. (You can remove the colour buffer as it has no real use anymore).
   - Add a texture sampler to the fragment shader and use the colour from the texture.
   - Activate the texture resource and use it to draw a texture on the quad using the shader.

Delivery

Commit your complete project to a dedicated folder inside your SVN repository (e.g. S0006E/assignment) and upload the number of the revision to Canvas.