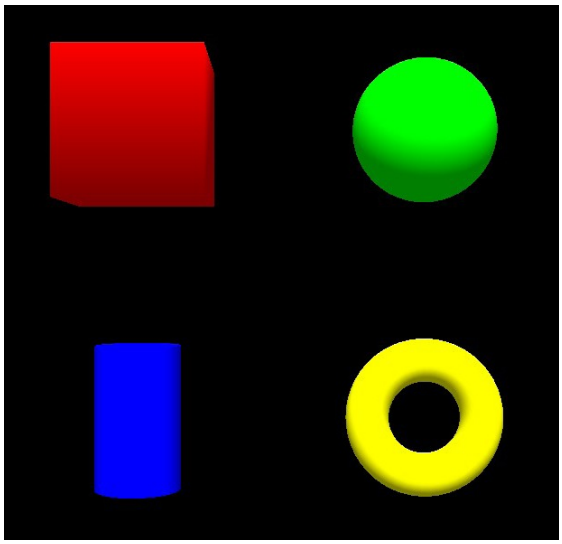
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Advanced Rendering Assignment sheet 1 – ST 2013
Prof. Dr. Gitta Domik, Stephan Arens
In Lab on April 15 to 19 – Homework due to April 24, 9:15
(Send homework to stephan.arens@upb.de)

Modeling, Lighting and Texture Mapping

Assignment 1

Download the `JoglTemplate.java` from koaLA. Create a **derivate** class and **overwrite** the method `display`. Draw four different OpenGL objects in one canvas (red cube, green sphere, blue cylinder, and yellow torus). Use GLU and GLUT functionality for the rendering and object generation.



Assignment 2

Insert different light sources into your OpenGL scene (directional light, positional light and spotlight) and compare their behavior. To see the colors of the objects `GL_COLOR_MATERIAL` must be enabled. Compare shademodel `GL_FLAT` with `GL_SMOOTH`.

Assignment 3

Now disable `GL_COLOR_MATERIAL` and attach different materials to your objects (e.g. try the materials given below).

```
float[] brassMaterials = { 0.33f, 0.22f, 0.03f, 1.0f, 0.78f,
0.57f,0.11f, 1.0f, 0.99f, 0.91f, 0.81f, 1.0f, 5.0f };

float[] redPlasticMaterials = { 0.3f, 0.0f, 0.0f, 1.0f, 0.6f,
0.0f,0.0f, 1.0f, 0.8f, 0.4f, 0.4f, 1.0f, 10.0f };

float[] whiteShineyMaterials = { 1.0f, 1.0f, 1.0f, 1.0f, 1.0f,
1.0f,1.0f, 1.0f, 1.0f, 1.0f, 1.0f, 1.0f, 100.0f };
```

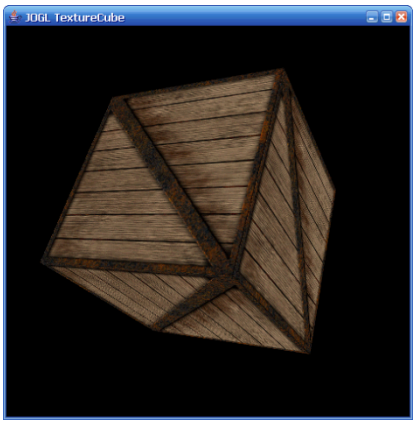
- **Homework:** Send 1 screenshot containing lighting, some materials and flat as well as smooth shading.

Assignment 4

Download the "crate.png" file and draw a textured crate. Use `TextureIO.newTexture(File file, boolean mipmap)` to load the texture in your init method and `Texture.bind()` to bind the texture before drawing the crate. For every vertex of the crate, specify texture coordinates with `glTexCoord2f(float s, float t)`.

Hint: Don't forget to enable texturing (`glEnable(GL.GL_TEXTURE_2D)`)

- **Homework:** Send 1 screenshot similar to the image underneath.



Assignment 5 (team work)

Use a 3D modeling tool of your choice to model supplementary objects for your project (e.g. a banquette, plants or whatever you need). Additionally you can use free models found on the internet.

1. Export each object to an .obj file and use the `MeshLoader` or `OBJLoader` class from koaLA to load them into OpenGL display lists. Create a scenery based on these objects. For now, all objects can be opaque (we will discuss blending and alpha sorting later).
2. Later on, your project will consist of a lot of different objects with different material properties, textures, animation paths and shaders. Create an object hierarchy with sub- and superclasses that can easily be extended for this purpose to handle all your meshes, properties, lighting and a camera in your scene. A very rough starting point can be found in SimpleSceneGraphExample.pdf in koaLA.