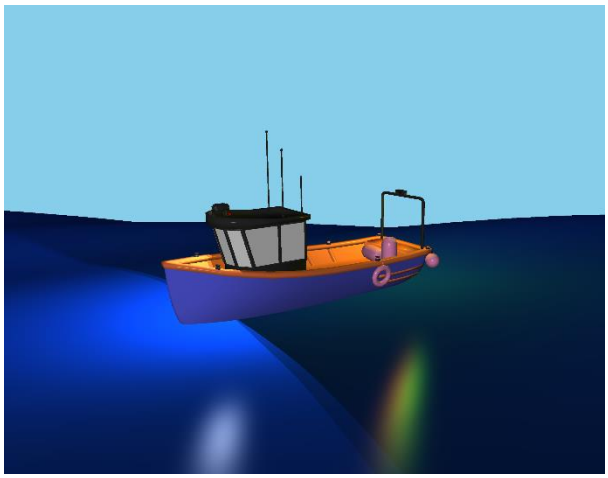
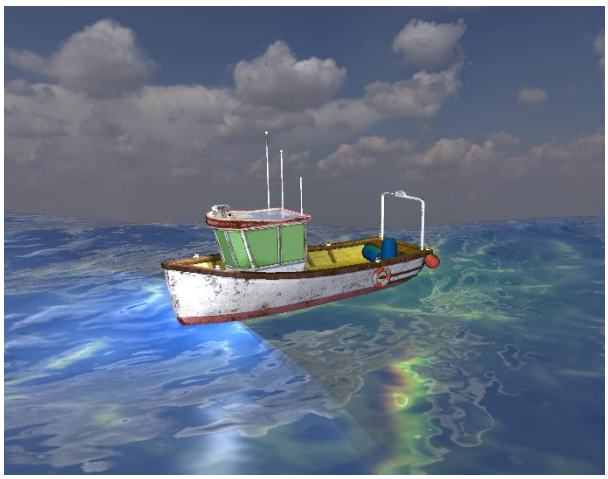
Proseminar Visual Computing
Winter Semester 2023

CG Assignment 3
Hand-out: December 12, 2023
Hand-in: January 08, 2024

Topics

- Texturing
- Shader Programming
- Skybox and Environment Mapping

Outline

The goal of the Computer Graphics assignments of the Visual Computing PS is to build a controllable boat moving on an animated water surface. This work is divided into 3 steps. Each step corresponds to a programming assignment. The objective of this assignment is to enhance the visual appearance of the scene by using textures. Material properties for *Blinn-Phong* illumination should be read from textures (with animated texture coordinates for the water surface) instead of using material colors. Furthermore, a *Skybox* should be added and used to apply *Environment Mapping*.

Template code

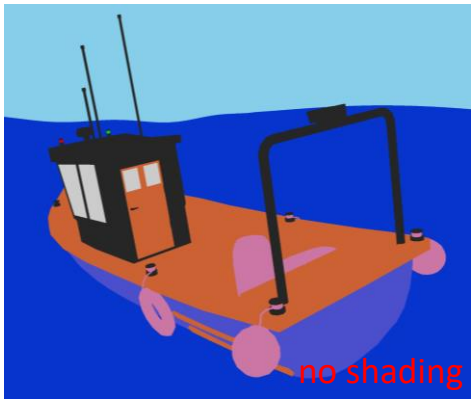
A template code is provided with this assignment. It loads the boat mesh, material properties, and textures given by an .obj file. The boat controls, the wave animation and *Blinn-Phong*

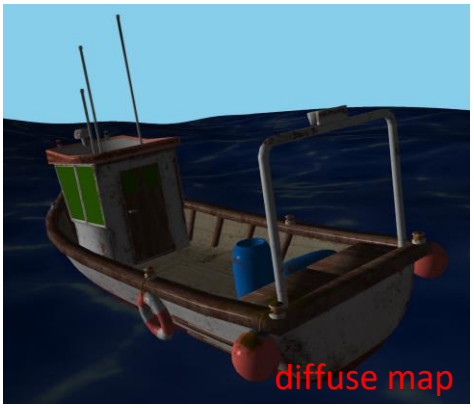
illumination from the previous assignment are already implemented (no textures).

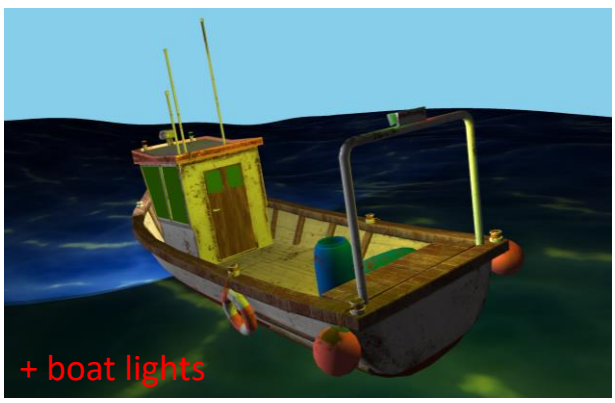
Further, code for loading a cube map (skybox) is available in the files *src/mygl/cube_map.h* and *src/mygl/cube_map.cpp*. We provide an example cube map that you can use in *assets/kloofendal_48d_partly_cloudy*.

Tasks

1. Adapt *Blinn-Phong* illumination to use the loaded texture maps (*map_diffuse, map_specular, map_normal, map_ambient*) to retrieve material properties for each fragment instead of the constant color values per surface material.







The value from the ambient occlusion map represents the strength of the ambient light reaching a certain surface position. Use this value to multiply the diffuse material to get the ambient material color.

The normal maps contain normals retrieved from a high resolution model including small details. Using per fragment normals from this map for shading allows to represent these details without increasing the complexity of the model. Usually, they are given in tangent space and need to be converted into object space and world space. For simplicity, the provided normal maps (boat and waves) are given in **object space**. Texels of the normal maps, have three channels (*x*, *y*, *z*) ranging from 0 to 1. Since the normals are given in object space, you need to convert them ro the range of -1 to 1 (encode directions over the whole sphere).

Given a texel *t*, the object space normal n_o is calculated as follows:

$$n_o = t \cdot 2.0 - 1.0.$$

Further, you need to transform the normal into world space. As already discussed in the lecture, given model matrix *M*, normals can be transformed as follows:

$$n_w = (M^{-1})^T n_o.$$

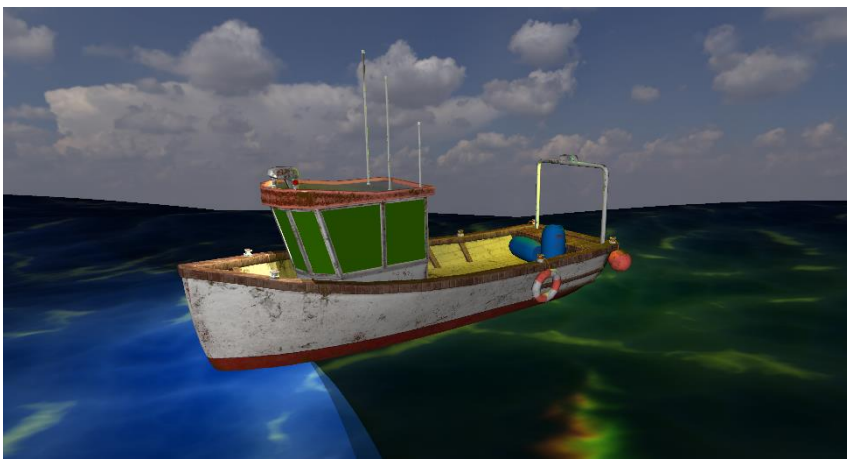
These resulting normals can directly be used for the illumination of the boat. For the water, an additional step is required since the normals from the normal map only add small wave details, but the overall orientation of waves is given by the vertex normals n_v (from the vertex shader (assignment 2)).

The normals for the water surface n_s can be computed as:

$$n_s = \mathrm{norm}(s \cdot n_w + n_v),$$

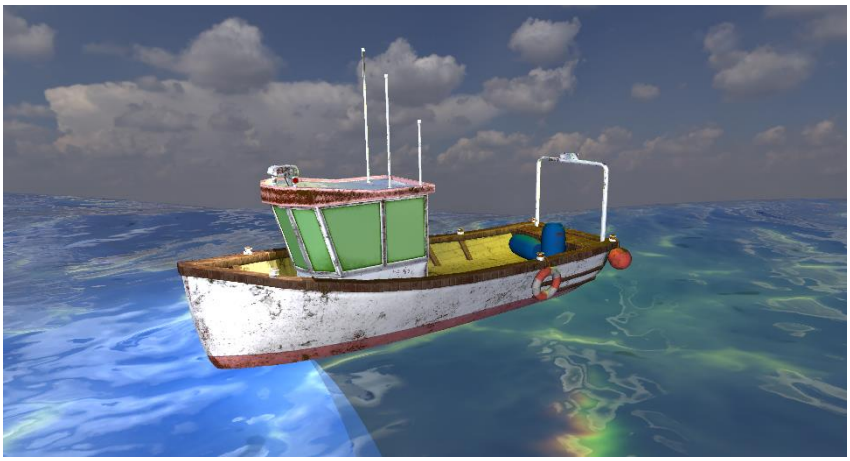
where *s* is a scaling parameter for the wave texture (e.g. $s = 0.25$).

2. In the next step, the visual appearance of the water surface should be further improved by animating its textures. For this task, it is enough if the water texture slowly moves over time along the water surface.
3. Write shaders to render a *Skybox* using *Cube-Mapping*. Code that can be used to load the textures and create the geometry buffers of a *Cube Map* is available in the template code files *src/mygl/cube_map.h* and *src/mygl/cube_map.cpp*. To switch between day and night, multiply the sampled texture from the cube map with the color of the directional light (*e.g.,* sun color).



Next, extend the *Blinn-Phong* rendering to include reflections of the environment. This is done using *Environment Mapping*: for each fragment, the reflection direction is calculated to sample from the cube map. To adjust the reflection strength of different materials, multiply the specular color (from specular map) with the directional light color (*e.g.,* sun color). Add this resulting fragment color to the color calculated with the Blinn-Phong illumination model.

Note: see the resources below for a tutorial on cube maps and environment mapping.



4. At last, write/draw the names of the group members or something similar somewhere on the boat, by modifying the diffuse texture of the boat body. It should be visible in the rendering of the scene.

Implementation Remarks

Make sure that your code is clear and readable. Write comments if necessary. Your solution should contain a readme file with names of the team members, list of keyboard controls, and any explanation that you think is necessary for the comprehension of the code.

Submission and Grading

Submission of your solution is due on January 08th, 2024 (23:59). **Submit the sources** (i.e., only the content of the *src* folder) + the modified diffuse texture and a screenshot of the rendered boat with it in a ZIP archive via OLAT. Do not submit the executable and the content of the *build* folder. Do not submit the external dependencies either. Both folder and archive should be named according to the following convention:

Folder: **CGA3_<lastname1>_<lastname2>_<lastname3>**

Archive: **CGA3_<lastname1>_<lastname2>_<lastname3>.zip**,

where <lastname1>, etc. are the family names of the team members. Development in teams of two or three students is requested. Please respect the academic honor code.

In total there are 15 marks achievable in this assignment distributed as follows:

- Use textures to map material properties to fragment (**6 marks**)
- Water texture animation (**1 marks**)
- Skybox and environment mapping (**5 marks**)
- Draw names of group members to texture **(1 mark)**
- Code readability, comments, and proper submission: (**2 marks**)

Resources

- Lecture and Proseminar slides as well as code and information are available via OLAT.

- OpenGL homepage
  http://www.opengl.org

- OpenGL 3.3 reference pages
  https://www.khronos.org/registry/OpenGL/specs/gl/glspec33.core.pdf

- OpenGL Tutorial for Blinn-Phong Illumination with textures
  https://learnopengl.com/Lighting/Basic-Lighting
  https://learnopengl.com/Lighting/Lighting-maps

- OpenGL Tutorial for Cube Maps, Skyboxes and Environment Mapping
  https://learnopengl.com/Advanced-OpenGL/Cubemaps

- GL Framework GLFW
  https://www.glfw.org/documentation.html

Note: Be mindful of employed OpenGL and GLSL versions!