Exercise 02561-08 **Hardware Shadow Mapping - OpenGL**

Readings Help slides on CN

Introduction Where there is light there are shadows, yet in real time graphics, cast shadows are often conspicuously missing. However, there are techniques for rendering shadows without loosing interactivity. In this exercise, we shall try one such method. The help slides are very important in this exercise, and these instructions for the exercise should be seen as an overview only.

Purpose To understand and implement shadow mapping.

Part 1 First compile `hwshade.cpp` in the HardwareShading directory. You will see a moving teapot. Your goal is to create a quad below the teapot with a texture mapped shadow.

Part 2 `set_from_light_perspective(...)`
Use the bounding sphere of the object to create a perspective projection that just fits the object.

`init_proj_shadow_texture(..)`
Initialize texture: Generate a texture name, bind it and set up clamping and minification and magnification filters.

Part 3 `make_proj_shadow_texture(...)`
First render the scene from the light source and copy the framebuffer to an intensity texture. You should only render the object that casts shadow (the teapot) and not anything else. The teapot should be rendered dark on a white background.

Part 4 `draw_proj_shadow(...)`
Render the quad beneath the teapot using the shadow texture. Render the quad black with alpha=1 and use the shadow texture to modulate. Render the quad once more, texture mapped with an image and use the alpha value in the frame buffer to blend between black (the colour in the frame buffer) and the texture mapped quad.

Part 5 Optional extensions
Try to generate the texture coordinates using both texture coordinate generation and by simply copying the vertex positions to texture coordinates using `glTexCoord`. Extend the program to include a depth test when drawing the shadow. This will make the program work also when the teapot is partially submerged.