Lab 06: Shadow Mapping

Task 1: Create a scene.

Task 1.1: Study the Light class, the material struct and how we update the parameters to the GPU. Use the keys {I, J, K, L, U, O} = {-z, -x, z, x, y, -y} to move the light in the scene.

Task 1.2: Load a sphere object using the **Drawable** class.

Why is there a texture on the sphere? How can we fix this?

Task 1.3: Create a plane underneath "Suzanne" to project the shadows on. Use the **Drawable** class to create the plane.

Task 1.4: Use different materials and different translations on each object.

Task 1.5: With the press of button "F1", render the scene using the light projection and view matrix.

Task 2: Create a quad at the top right corner of the view port that can render textures. This will be used to visualize the depth map later.

Task 2.1: Load a new shader program to render the quad.

Task 2.2: Create a quad, using the **Drawable** class.

Why does the position of the quad remain constant, no matter the position of the camera?

Task 2.3: Render a sample texture on the quad.

Task 3: Create the depth texture.

Task 3.1: Load the depth shader program.

Task 3.2: Create a depth buffer and a depth texture.

Task 3.3: Render the scene from light's perspective using the depth shader program to create the depth map.

Don't forget to:

1. Use the new framebuffer.
2. Use the depth shader program.
3. Change view-port shape to the shape of the shadow texture.
4. Clear depth_buffer_bit.
5. Bind the default framebuffer after rendering.

Task 3.4: Visualize the depth map on the quad.

Task 3.5: Is the depth map dynamically updated? If not, how can we fix it?

Do we always need dynamic shadows?

Task 4: Use the depth texture to create the shadows.

Task 4.1: Send the depth texture to the shader program. Also, you have to make the light's view-projection matrix uniform, to move the fragments to the light-space.

Task 4.2: Calculate the vertex_position_lightspace in the vertex shader.

Task 4.3: Calculate the shadow component.

Task 4.4: Fix the quantization problem.

Task 4.5: Make the area outside the light's viewport to be bright i.e., no shadow is casted to that area (See depth texture parameters).

Task 4.6: Apply Percentage – Closer Filtering (PCF) to make the edges of the shadow more realistic.

Homework

Homework 1: Introduce a second light source. Modify the program so that the user may press the keys "1" or "2" to select which source to move when the keys {I, J, K, L, U, O} are pressed.

Homework 2: Calculate the shadows casted by both light sources. Some parts of the scene should be lit by one light source while shadowed by the other, some should be lit by both sources, and some should be shadowed by both of them.

Homework 3: Draw two spheres at the positions of the two light sources, in order to visualize the sources' positions in the 3D space. Those spheres should cast no shadow, since their role is only to assist with our understanding of the scene.

Homework 4: Get the aforementioned spheres to be lit by the light source. To do that, since the light source is inside each sphere, you will have to invert the spheres' normals.

(Bonus) Homework 5: There is an inconsistency! While the shading simulates a point light, the shadows are that of a directional light. Fix this by changing the point light to a directional light pointing at the (world space) origin.
Hint: You'll have to modify the "L" vector. You may need to propagate more information from the vertex shader to the fragment shader.