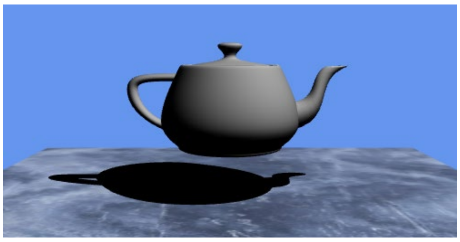
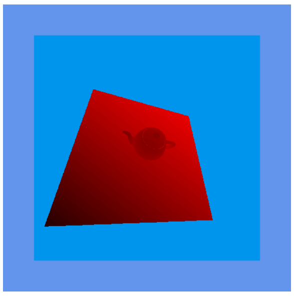
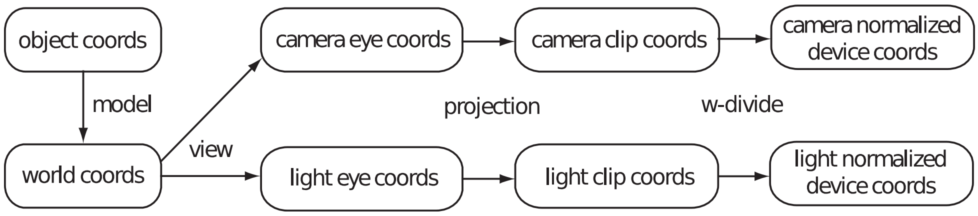
02561 COMPUTER GRAPHICS DTU COMPUTE

Worksheet 9: Shadow mapping

Reading	RTR: 7.4-7.6
Purpose	The purpose of this exercise is to understand and implement shadow mapping. This includes a deeper understanding of the different coordinate spaces in the pipeline as well as mapping between these coordinate spaces.
Part 1 Scene	The scene is a teapot jumping up and down on a textured ground quad with a point light circling the scene. We can set up this scene by combining Part 3 of Worksheet 5 with Part 3 of Worksheet 8.  The main difficulties are that (a) the ground plane and the loaded object use different shaders and (b) we need to position the teapot in the scene. • Use your code from Part 3 of Worksheet 5 to load and render the teapot model, which is available on DTU Learn. Scale the teapot to a quarter of its original size and construct a model matrix for it that translates it by the vector $(0, -1, -3)$. • Insert shaders and the part that initializes and draws the textured ground quad from Part 3 of Worksheet 8. The ground quad and the teapot use different shaders (meaning that they need different pipelines). • Move the teapot up and down over time by modifying the model matrix (translate along the y-axis from -1 to 0.5). Create a button that turns this motion on/off.
Part 2 Projection shadows for reference	 • For reference, insert the black projection shadows from Part 3 of Worksheet 8. In this scene, we use a model matrix to move the shadow-casting object. It is important to realize that the model matrix should be applied first (before the shadow projection matrix) when rendering the shadow polygons. • Set the light direction in the teapot shading according to the position of the point light circling the scene. Create a button that switches point light animation on/off.

Worksheet 9: Shadow mapping

Parts 3-4 Shadow mapping 	Projection shadows have several shortcomings. A significant problem is missing self-shadowing. Shadow mapping solves most of these issues (but introduces other problems). Your task is now to replace the projection shadows from Part 2 with shadow mapping. There are two kinds of coordinate spaces used in this assignment: camera relative and light relative. The following figure illustrates these coordinate spaces and the transformations between them.  The basic steps are: 3. Render the scene from the point of view of the light source. Use a shader that renders fragment depth directly into a texture (we recommend use of an `rga32float` texture). Bind the depth texture when drawing the ground plane to inspect the result and use this inspection to set proper light view and light projection matrices. 4. Use the rendered depth texture in the other shaders to determine whether a fragment is in shadow or fully lit. Make the shadows dark but not pitch black. This is done by adding ambient light regardless of whether a point is in shadow or not.
Part 5	Compare projection shadows to shadow mapping by listing advantages and disadvantages of the two techniques.
Part 6 **Optional**	Implement shadow antialiasing by averaging multiple shadow map lookups close to each other instead of using a single lookup. This technique is called “percentage-closer filtering” (RTR: 7.5).[1]

[1] See Bunnell, M., and Pellacini, F. Shadow Map Antialiasing. In *GPU Gems*, Chapter 11, Addison-Wesley, 2004. https://developer.nvidia.com/gpugems/gpugems/part-ii-lighting-and-shadows/chapter-11-shadow-map-antialiasing