02561 COMPUTER GRAPHICS DTU COMPUTE

Worksheet 6: Texture mapping

Reading	RTR: Section 6.1-6.2.
Purpose	The purpose of this set of exercises is to load images and map them onto 3D objects as color textures. This should lead to an understanding of the principles of 2D texture mapping and how it can be used for polygon meshes.
Part 1	Create a rectangle with vertices $(-4, -1, -1)$, $(4, -1, -1)$, $(4, -1, -21)$, $(-4, -1, -21)$. Set up a perspective camera with a 90° field of view. Use the default view matrix and draw the rectangle in white on a blue background. Map a procedurally generated checkerboard texture to the rectangle using the following steps. • Create a buffer with texture coordinates $(-1.5, 0.0)$, $(2.5, 0.0)$, $(2.5, 10.0)$, $(-1.5, 10.0)$ for your rectangle, such that the texture repeats four times along the width and ten times along the length of the rectangle. Set up the texture coordinates as another input to the vertex shader (another layer in the vertex buffer). • Generate a 64×64 resolution texture image that forms an 8×8 black-and-white checkerboard and write it to the GPU as a texture together with a sampler that sets the address and filter modes (use repeat addressing and nearest filtering). • Pass the texture coordinates from the vertex shader to the fragment shader and use them to sample the texture map in the fragment shader. Return the color received from the texture map.
Part 2	Generate mipmaps using the `genmipmap.js` library file available on DTU Learn. Use the `numMipLevels` function to add a `mipLevelCount` when creating the texture. After writing data to the texture, generate the mipmaps using the `generateMipmap` function. Create selection menus that enable you to switch between different texture wrapping modes (repeat or clamp-to-edge) and different texture filtering modes (`minFilter`: nearest or linear, `magFilter`: nearest or linear, `mipmapFilter`: nearest or linear). Make it possible for the user to enable or mdisable mipmapping. Explain the effect of the different filtering modes and their influence on texture magnification and minification issues.

Worksheet 6: Texture mapping

<table>
<tr><td>Part 3</td><td>Start from a diffuse sphere illuminated by a directional light (Part 3 of Worksheet 4). We will now map a texture depicting Earth onto the sphere. To do this, we load the texture from an image file and compute the texture coordinates in the fragment shader. Some steps to follow:
When initializing the texture, load the texture image from the file <code>earth.jpg</code> (available on DTU Learn).
The next step is to pass the normal of the sphere to the fragment shader and use it (after re-normalization) to compute the texture coordinates. The normals define points on the unit sphere. The unit sphere is then an intermediate surface to which we can map texture coordinates. Use spherical coordinates to define the relation between a surface normal (a point on the unit sphere) and the texture coordinates (u and v).
Invert the relation you found using inverse trigonometric functions. Use the resulting formula in the fragment shader to calculate texture coordinates from the surface normal. An atan2 function is needed to get the signs right.
Use the color found by texture look-up as k_d of the sphere and illuminate the sphere by a directional source and an ambient source.
Set the background color to black and enable camera orbiting of the globe. The earth texture has high resolution leading to minification issues, especially in the mountain ranges during a spin. Choose a filtering option that betters these minification issues without too much blurring of the texture. Explain your choice.</td></tr>
<tr><td>Part 4 (optional)

Think of this as a project proposal.</td><td>Create texture coordinates for the triangle mesh that you rendered in Worksheet 5 and include these texture coordinates in the OBJ file when the mesh is exported. Create or find texture images and use them for texture mapping of your triangle mesh. Modify your rendering of the triangle mesh such that it includes texturing of the object surfaces.</td></tr>
</table>