CS 405 Project 2 Report

Tolga Aydın 27843

Entrance

This report includes details of the "Texture + Lighting" project, which was expanded within the scope of the CS 405 course. The project was implemented via WebGL, with two main uses: texture support for non-2D images and the ease of changing simple lighting.

Task 1: Non-Power-Of-2 Texture Support

In this task, it is possible to change the texture of images of any size by changing the setTexture method. In WebGL, it is generally preferred that the width and distribution values for texture images be powers of 2. However, to overcome this limitation, the following changes were made to the setTexture method:

If the image's dimensions are not a power of 2, we set the texture wrapping mode using gl.CLAMP_TO_EDGE.
We set the texture mode using gl.LINEAR.
These changes ensure accurate rendering of any size image.

Task 2: Simple Lighting Implementation

Overview

Task 2 involves adding basic lighting effects to the scene. In this task, ambient and diffuse lighting are discussed. These effects are important to add realism and depth to the scene.

Methodology

Adjustments in Constructor:

Initial lighting-related values were defined in the constructor of the MeshDrawer class. These values include the position of the light (lightPos) and ambient light intensity (ambientIntensity). Additionally, uniform locations to control the lighting (for example, this.lightPosLoc, this.ambientLoc, this.enableLightingLoc) were taken from the WebGL program.

Normals in setMesh Method:

In the setMesh method, a buffer was created for the normals of the mesh and these normals were transferred to the WebGL buffer. This step is critical for passing the surface normals that will be used in lighting calculations to the shader.

Lighting Update in draw Method:

Each time the scene was drawn (when the draw function was called), the light position and ambient light intensity were transferred to the shader.
This allows the lighting effects in the scene to be dynamically updated. In particular, it is important so that the user can change the light intensity and position.

Light Calculations in Fragment Shader:

meshFS (fragment shader) updated. Shader calculates diffuse and ambient lighting based on the enableLighting uniform variable.

While ambient light is applied at a constant intensity at every point of the scene, diffuse light changes depending on the surface normals and the location of the light source.

UI Interaction:

The "Ambient Light Density" slider and arrow keys in the user interface allow the user to dynamically change the light intensity and position. These changes take effect by redrawing the scene.

Conclusion

These changes provided a more realistic and impressive visual experience on stage. While ambient light adds general illumination to the scene, diffuse light emphasizes the surface details and depth of objects. Thanks to user interaction, the lighting settings of the scene can be controlled in real time.